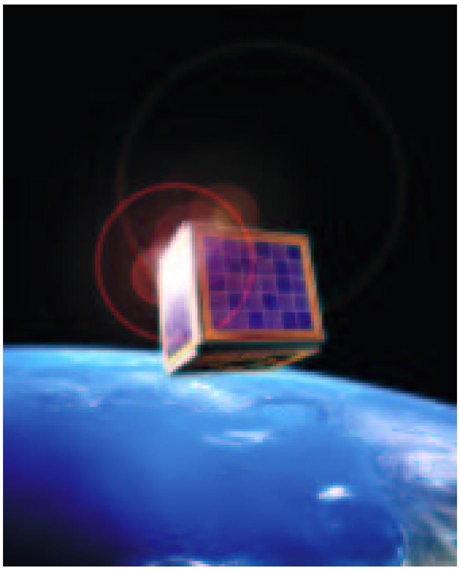
# Power Supply

## for the AAU Cubesat



REPORT

AALBORG UNIVERSITY
P5 - Project
Group 01gr509

# Power Supply Unit for the AAU-Cubesat

01GR509

December 20, 2001

Institute of Electronic Systems
University of Aalborg

**TITLE:** Power Supply for the AAU Cubesat

**PERIOD:** E5, 4th of September - 20th of December, 2001

**PROJECT GROUP:**

Group 01gr509

**GROUPMEMBERS:**

Lars Alminde
Morten Bisgaard
Fjölnir Gudmundsson
Claus Kejser
Toke Koustrup
Christian Lodberg
Tor Viscor

**SUPERVISOR:**

Gert K. Andersen

**COPIES PRINTED:** 10

**PAGES:** 238

**APPENDIXES:** 14

**ENDED:** 20th of December

**SYNOPSIS:**

This report is about the Power Supply onboard the AAU Cubesat, which is a student satellite designed at Aalborg University. First the space environment is analyzed in order to specify the requirements to the product.

Hereafter requirements are specified for the individual parts of the hardware and then it is designed and built. All hardware parts are module tested and evaluated. Software is designed and tested.

When the module tests have been completed with a satisfying result, the different hardware parts are integrated and tested. After that the product is evaluated and a conclusion on the product is made.

Because of time shortages the project has not ended with satisfying results, since work had to be abandoned before the product was brought to an functional level suited for the system acceptance test.

# Preface

This report is written by seven students from the fifth semester of the electronic engineering education of the Aalborg University.

This project was chosen primary on basis of the groupmembers great interest in space-technology and because of the possibility to work on an inter-disciplinary project and finally because of the possibility to work with a project that in a sense is more ”real” than an ordinary university project where the aim normally is to develop a limited prototype-model.

The project group would like to give thanks to group PED9 from the institute of power electronics for both good cooperation and competition. Also thanks to Elmo Schreder for figures, the projekt groups 01gr732, 01gr930 and 01gr931 who wrote some material we use in the report. Finally thanks to Analog Devices and Danionics for giving us samples and batteries for the power supply.

Lars Alminde

Toke Koustrup

Morten Bisgaard

Christian Lodberg

Claus Kejser

Tor Viscor

Fjölnir Gudmundsson

# About this Report

## Disposition of the Report

The report is divided into the following main parts:

- Analysis and Requirements
- Hardware and Control Design
- Software Design and Implementation
- Deployment, Acceptance-Test and Result
- Appendixes

All test-specifications both for module-tests and the system acceptance test are enclosed as appendixes of the main report. Further this approach aims to increase readability by focusing on requirements and design in the main report. The following will briefly state the contents of each part of the report.

### Analysis and Requirements

The Analysis and Requirements part begins with a short description of the environment of the satellite, hereunder examinations of radiation, temperature, pressure and vibrations. After that the requirement specification is situated in order to introduce the specific requirements put on the power supply. The part ends with a system analysis where different possible topologies for the system design are discussed.

### Hardware and Control Design

The Hardware and Control Design begins with a chapter that provides a general overview of the hardware of the power supply. Hereafter the different hardware parts of the power supply are designed and implemented. First the converters and then the protection circuits are designed. Followed by design and simulation of controllers for the converters and finally the digital hardware is designed, implemented and tested.

### Software Design and Implementation

This parts analyses the resources available for the software and investigates what requirements the software must fulfill. Then an overall design approach is chosen and the individual parts of the software is designed and implemented. Finally the software is integrated and emphasis is put on software scheduability.

### Deployment, Acceptance-Test and Results

In this part the assembled implementation of all designs together are described and the system acceptance test is conducted. Finally the results of the project are evaluated and stated in the conclusion.

### Appendixes

Immediately after the conclusion, the bibliography is located as well as a list of commonly used abbreviations. Hereafter the appendixes of the report is located in the following general order.

1. Power input and output analysis appendices
2. Converter design appendices
3. Test appendices
4. Miscellaneous appendices

## Typographic Aids

Bibliographic references are given in square brackets with the name of the author(s) and the year of publication. An example of a bibliographic reference: [Connolly, 2000]. For datasheets of electronic components the capitalized component name is used for reference: [ICL7660, 1999]. If no explicit author can be identified then the company or organization name is used instead. For example: [IEEE, 1998a]. Some references may include a specific page number in the reference, but this principle is not adhered to in general.

References within the report comes in two types. If the reference is within the current chapter then only the section number is used for reference (e.g. section 2.1). Else if the reference is to another chapter of the report then the reference is given with both section number and page number (e.g. section 2.1 on page 21).

Notes in the report are given in footnotes[1]. Figures and tables are identified by the chapter number and an incrementing number within each chapter. The same numbering scheme is used for equations and formulas, but only important and/or referenced formulas are numbered. An example:

$$E = m \cdot c^2 \tag{1.1}$$

Snippets of software-code are written with C-syntax and given between two horizontal bars. For example:

```
for (int a=0; a<5; a++)
{  b[a]=c();
}
```

In text references to entities in the software-code such as variables, objects or functions the reference is indicated by use of the courier font. Examples: `counter`, `car::engine` or `Push()`.

Hexadecimal numbers are prefixed with a $ symbol (e.g. $2A) and binary numbers are prefixed by the % symbol (e.g. %101).

## Contents of the Enclosed CD-ROM

The CD-ROM that is enclosed with this report contains all sourcecode[2], datasheets[3] used in the project, as well as some documents written by other groups working on the Cubesat project referenced from this report. The CD-ROM has a file `README.TXT` in its root directory which list the full contents of the CD-ROM. References to files on the CD-ROM are given by the complete path in a footnote.

[1] This is an example of a footnote

[2] CDROM/Sourcecode

[3] CDROM/Datasheets

# Contents

# List of Figures

# List of Tables

# Chapter 1
# Introduction

The AAU Cubesat is a satellite made by students at Aalborg University. Therefore it is made primarily in an educational aspect, and thus it has to fulfill the requirements to the project work at Aalborg University. Therefore the mission of the AAU Cubesat is stated as follows (quoted from [01gr930, 2001] p. 5):

**Primary Mission**

- To show that the students at AAU are able to design and build a satellite. Hereby also gain experience in designing satellites and pico-satellites in particular.

**Secondary Missions**

- To communicate with the satellite while it is in orbit, in order to receive health data for the on-board subsystems. Hereby getting knowledge on how well each subsystem works.

- To take satellite photos of the Earth and transmit them to a ground station. Hereby testing the attitude control and CMOS camera in LEO.

- To let people via the Internet choose a geographic site to be photographed and then later retrieve the photo from the Internet. Hereby increasing public interest in space science, technology and natural science in general.

- To see if the Cubesat with its configuration of hardware and especially the camera will be able to monitor the stars in order to find unknown planets. A future Cubesat project considering such a mission could benefit from this knowledge.

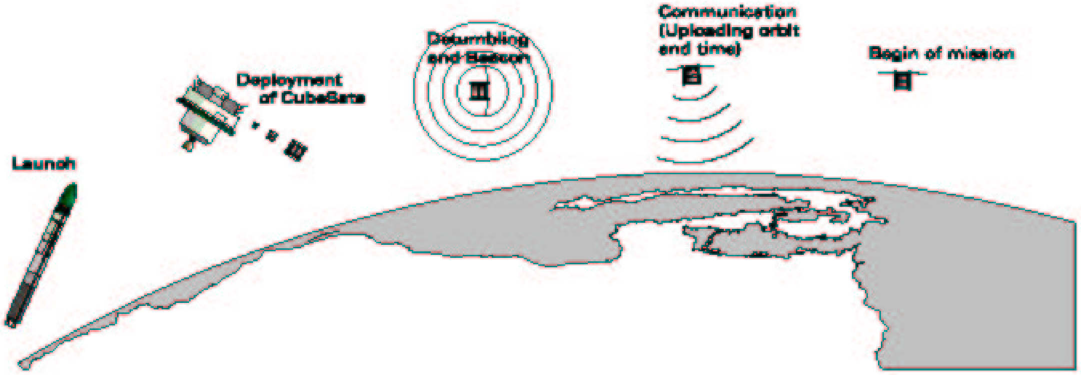


**Figure 1.1:** The Cubesat mission profile

The idea behind Cubesat is to give students an opportunity to develop items conditioned for the environment that space offers. The name ”Cube” origins from the fact that the satellite is shaped as a cube measuring $10\ cm \cdot 10\ cm \cdot 10\ cm$. Compared to other satellites, Cubesat is a very small satellite and is therefore called a ”pico satellite”.

There are no limitations to the satellite itself other than the physical dimensions are limited to $10\ cm \cdot 10\ cm \cdot 10\ cm$ and a maximum weight of one kilogram. These restrictions have to be maintained in order to restrict the launch costs in order to make the satellite fit in the P-Pod launch system. The launch is planned to take place November 2002.

The interdisciplinary project runs as a cooperative effort between different institutes at Aalborg University. The following institutes are involved in the project: Institute of electronic systems, institute of mechanical engineering, institute of computer science, and institute of power systems.

The AAU-Cubesat project has been split up in to smaller projects which the concerning institutes have responsibilities for. The project must be made by students within the institutes.

The aim of this specific project is to design the power supply for the satellite. The power supply is designed in cooperation with two other groups and the AAU-Cubesat project. In the beginning of the project there were no final demands or specifications for the final satellite, therefore the group had to make their own specifications and adopt the real ones as they arrived.

# Part I

# Analysis and Requirements

# Chapter 2
# The Space Environment

In this chapter the environment, in which the satellite is going to operate, and the effect it has on the power supply, is analyzed. The analysis is limited in that calculations on the environment will not be performed, instead the discussion will be more general. The environment that will be analyzed are the surroundings during launch and after deployment in orbit. The satellite will be released in ”Low Earth Orbit (LEO)”, which is at an altitude of about 600 km.

## 2.1 Radiation

In space there are basically two kinds of radiation that effects the satellite. The electro magnetic radiation and particle radiation. Most of this radiation comes from the sun, but cosmic radiation and particle radiation from Earth’s magnetic field are also significant. The particle radiation from the magnetic field, is basically particle radiation from the sun that has been caught in the earths magnetic field, and moves around the earth in broad belts. Because this radiation is contained (mostly) in these belts, it is of little significance outside these belts. In LEO, the satellite is below these belts, which start at about 1000 km of altitude.

Both kinds of radiation, particle and electro magnetic, will wear down electronics over time. For that reason electronics used on board have to be tested to make sure they are durable enough to withstand the radiation. To get an idea of the amount of radiation in the Cubesat’s expected orbit, the ESA program SPENVIS has been used to calculate the radiation. [1] The result of this can be seen in figure 2.1

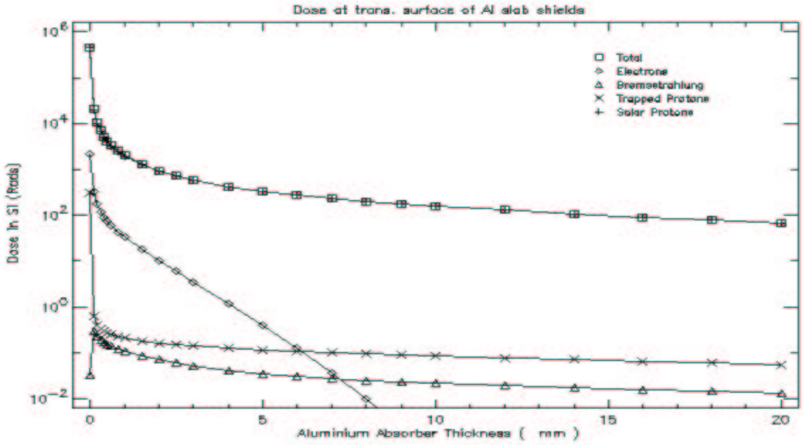


**Figure 2.1:** Yearly radiation dose vs shielding thickness

As can be seen from figure 2.1, the radiation on a unshielded surface during a one year period is about $1 \cdot 10^6 Rad(Si)$. Radiation can also cause errors in logic devices. This happens when a high energy particle hits the effected device and one of two things happen: First scenario is when the particle shoots through the device material and ionizes a path of particles on its way (see figure 2.2 left). This path will then act as a short circuit between the individual parts/layers of the devise. The second scenario is when a the high energy particle hits an atom with enough energy to split it, and the individual parts will then each trace a path of ionized particles throughout the devise material(see figure 2.2right).

This phenomenon is called a Single Event Upset (SEU), and can change digitally stored data or cause a gate to open or close at the wrong time. Is the impact on the device of a more serious nature, the high energy particle can directly cause damage to the devise. This is called a Single Event Latch-up (SEL). If unlucky, these events can also effect other devices, e.g. by causing a bus contention, where separate devices attempt to force different voltages at the same point, and thereby burn out. Software can be programmed to detect and correct digital errors, but bus contentions and SEL’s must be detected and corrected in hardware.

[1] SPENVIS can be found at http://www.spenvis.oma.be/spenvis/

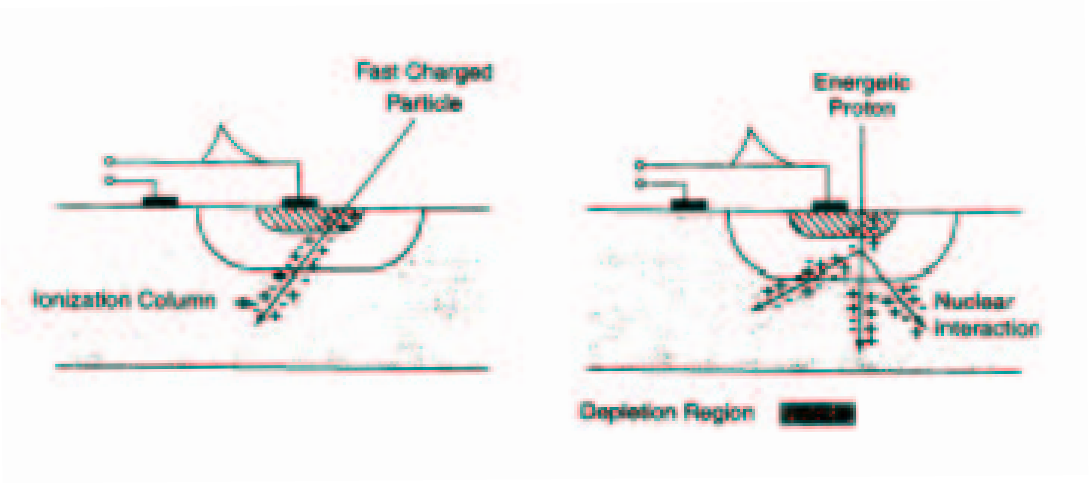


**Figure 2.2:** SEU through Ionisation column (left) or Nuclear interaction(right)

## 2.2 Temperature and Pressure

At an altitude of 600 kilometers there is very little atmosphere left and it can be considered a vacuum. This implies that components consisting of volatile materials will suffer from evaporation (out gassing) and therefore all components used in the power supply must be tested thoroughly for their ability to withstand the vacuum.

Since the power supply will be in a vacuum, all heat transportation will be by either heat conduction internally in the satellite or heat radiation and there will be no convection. Because of this, it must be considered that problems may arise concerning the discharge of heat from the satellite; therefore the power supply must be constructed so that the heat production is as low as possible. The temperature range that the satellite must be able to endure during launch is -40 °C to 80 °C. [Connolly, 2000] and these are the temperatures that the satellite will be exposed to when it is in the P-POD [2]. The temperatures in orbit will depend mainly on the thermal design of the satellite and since this has not been done yet we will here depend on calculations from other universities also building Cubesat's. The TIT of Tokyo calculates that the temperatures in orbit should varies from -40 °C to 80 °C [Yamaguchi, 2001] and the SSEL of Montana estimates that the temperature range is from -120 °C to 100 °C [Roesch, 2001].

All though there are differences in the calculations of the two universities then it can be concluded that the satellite must be able to survive in very hostile thermal environment

## 2.3 Acceleration and Vibration

As long as the satellite is in orbit, there will be no externally-induced vibrations, and the acceleration will be smaller than on Earth. During the launch, however, there will be both strong vibrations and a powerful acceleration. The satellite, and therefore also the on board power supply, has to be able to withstand an acceleration of 15 g [Connolly, 2000] This in itself should have no direct effect on most electrical components, but should be kept in mind in the overall design, e.g. avoiding tall, heavy components that could break during the acceleration.

The overall design also has to reflect the vibration caused during launch, and the vibration test the satellite will have to pass before being allowed to launch [Connolly, 2000]. Again, most components by themselves should be able to withstand the vibrations, but the use of mechanical components, such as turn potentiometers, should be avoided whenever possible.

## 2.4 The Effect of the Environment on the Power Supply

As it has been described above there are a few criterias that must be considered when constructing a space born power supply. The two major considerations are: The radiation and the temperature. As for the temperature it implies that the components used in the power supply should be industrial grade to be able to withstand the extreme temperatures in space. Considering the radiation effect the SEU is the most serious threat to the power supply as the components will be able to withstand the wear from radiation within the life time of the satellite. This implies that a redundence method to withstand SEU should be developed.

[2] the container, Cubesat will be in during launch

# Chapter 3
# Power Supply Unit Requirement Specification

## 3.1 Introduction

In this chapter the requirements specifications for the AAU-Cubesat Power Supply Unit (PSU) will be presented. This specification has been organized in accordance with the IEEE ”Recommended Practice for Software Requirements Specifications” [IEEE, 1998a]. In order to fulfill this specification all stated requirements must be met by the final product, and this should be verified by means of the product acceptance-test that is enclosed as appendix vI.

The process of acquiring the requirements for this specification has been on-going work from the start of the project till the end. The requirements have been negotiated both between the three groups from the AAU-Cubesat project that have been working to provide a Power Supply Unit for the Cubesat and the other groups in the Cubesat project that represent the users of the system.

Through this project a document called ”Power Supply Unit Interfaces” (see appendix N on page 233) has been maintained by this group on behalf of all three power supply groups. This document is the main source for requirements specified in this chapter. It should however be noted that some requirements have been updated, since the last revision of that document.

The whole process of creating the requirement specification has been very realistic compared to other projects the members of this group have been working on earlier. This is because of the need to negotiate requirements instead of ”guessing” what the users need, as done in traditional student projects.

### 3.1.1 Purpose

In this project the aim is to develop a PSU suited for the AAU-Cubesat project. This means that the final project should end with a product designed to survive in the space environment and which provides the functionality, required by the users on board the satellite.

### 3.1.2 Abbreviations

The following abbreviations are used throughout the requirement specification and the rest of the report:

PSU: Power Supply Unit

OBC: On Board Computer

DHS: Data Handling Software

ACS: Attitude Control System

TRD: Transmitter/Radio Device

### 3.1.3 References

The requirement specification has references to the following appendixes of this report:

Appendix N on page 233: Power Supply Unit Interfaces

Chapter 2 on page 21: The Space Environment

Appendix I on page 207: Acceptance Test Specification

Appendix G on page 201: Protocol Specification for External Communication

The following documents which have been worked out by other groups from the Cubesat project are referenced in this specification:

Document [01gr732, 2001]: On Board Computer Documentation

Document [MK9-P22A, 2001]: Mechanical Design of AAU-Cubesat

These documents are available from the enclosed CD-ROM[1]

### 3.1.4 Disposition of the Requirements Specification

The requirement specification consists of the following main sections:

- Introduction
- General description
- Functional requirements ordered by interfaces
- Quality of service requirements
- Environmental requirements

The general description presents the power supply interfaces and functionality in general terms. Hereafter requirements on functionality are specified for each of the PSU interfaces. Then requirements on efficiency, reliability and fault tolerance are specified. Finally environmental requirements such as structural requirements and temperature range are covered.

## 3.2 General Description

This section provides a general description of all power supply interfaces and functionality. The purpose of this description is to provide an overview of the power supply before the specific requirements are given in the following sections.

### 3.2.1 Perspective of the Product

Since two other groups, one other from the 5th semester and one from the 9th semester, are also developing a PSU for the AAU-Cubesat it is not given that the PSU developed in this project will come to be on the satellite. But if it is deemed the best option of the three when the projects are evaluated it will fulfill the perspective of the product which is to fly as the PSU of the satellite when it is launched in November 2002.

### 3.2.2 Product Functionality

The power supply is a fully autonomous system that receives power from a number of solar arrays with photovoltaic cells and conditions and distributes this power to a number of users which are the other electronic subsystems on board the satellite. Further, if more power is obtained from the photovoltaic cells than consumed by the connected subsystems then this power must be stored in a number of batteries, and when power consumption exceeds power input then the batteries should discharge in order to provide the needed power output. If a subsystem draws a current from the PSU that is large enough to suggest a malfunction of that user then the PSU must cut the current to the users and inform the DHS on the OBC. Further it must be possible for the DHS to specify which users are allowed to receive power at a given time.

In addition to the above described main functionality the PSU has a set of secondary functions that also must be fulfilled:

1. Collect housekeeping information including satellite temperatures
2. Act as boot-master and external watchdog for the OBC

In figure 3.1 a conceptual diagram of the PSU system is depicted with focus on the main functionality. The following paragraphs will discuss some of the PSU functionality in more detail than already given above.

The power delivered to the users is distributed from a 5 V controlled powerbus.

In order to communicate both housekeeping data and information about e.g. an user shut down due to excessive power consumption the PSU must be able to exchange information with the DHS. This is performed by

[1] CDROM/Datasheets/

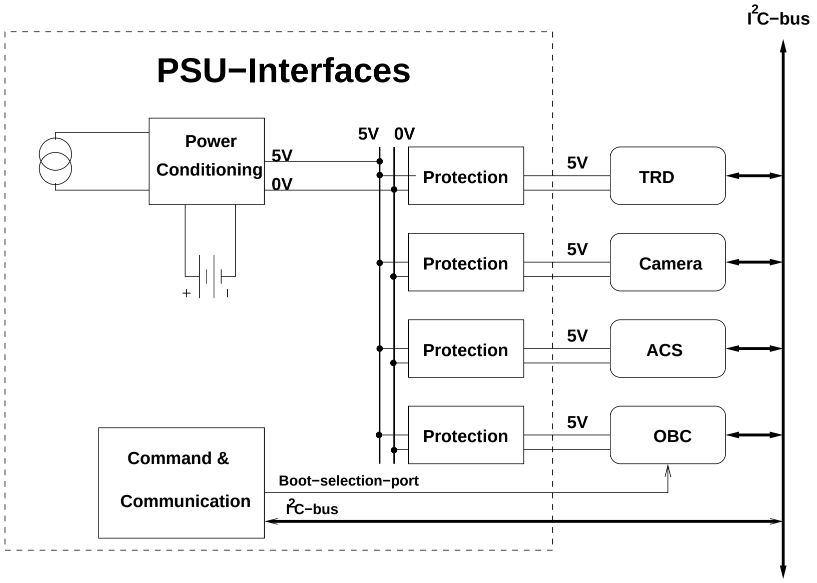


**Figure 3.1:** Overview of the PSU main functionality

means of an I$^2$C-bus, which has been selected as the communication interface between the different systems on board the satellite.

The secondary functions of the PSU regarding boot-master and watchdog functionality is to provide the OBC with a safe method for choosing whether it should boot up using its failsafe software placed in PROM circuitry or boot up using software placed in EEPROM. Furthermore, if the OBC fails to send signals to the PSU through then the I$^2$C-bus the PSU must reboot it because of the assumption that it is not working properly.

When the PSU receives power from the photovoltaic cells it is to control the current-flow in such a manner that maximum power is obtained. This control is necessary because there is a complex relationship between power-output, cell voltage, cell current and and received solar radiation.

The batteries chosen for the PSU is to be selected such that they are able to hold such an amount of charge that the satellite is allowed to fully operate throughout its active periods. This of course is under the assumption that the activity level of the satellite is constrained in such a way that the power input/output balance is either positive or zero for a time period corresponding to one period of both satellite activity and inactivity.

### 3.2.3 User Characteristics

The users of the PSU are the other subsystems of the AAU-Cubesat that requires electrical power in order to operate and the DHS which needs to acquire housekeeping information from the PSU and which specifies which users are to be turned on. All identified users are:

**OBC** is the hardware which runs the DHS

**DHS** handles information from subsystems and controls the satellite

**ACS** is responsible for controlling the orientation of the satellite such that camera and antennas are pointed in a specified direction

**Camera Payload** is responsible for the part of the satellite mission which is to take pictures of Denmark.

**TRD** is responsible for transmitting and receiving communication between the satellite and the ground-station. The TRD consists both of the specific radio-device as well as interface circuitry

The following table states for each user of the powerbus the requirements on minimum power (idle mode), maximum (active) power and the current-drain at which the user is to be shut down. These numbers have been specified in cooperation with the users.

| User | Minimum [mA] | Maximum [mA] | Shutdown current [mA] |
|---|---|---|---|
| OBC | 40 | 90 | 99 |
| ACS | 15 | 90 | 99 |
| CAMERA | 0 | 10 | 11 |
| TRD | 30 | 2200 | 2420 |
| Total | 95 | 2390 | n/a |

**Table 3.1:** Table of user requirements

### 3.2.4 Constraints

This project period on 5th semester runs from 4/9-01 to the 20/12-01. Since the requirements for a ”space-ready” product is more extensive than what is normal for a student project then time is the major constraint for this project.

Therefore, the primary goal for this project is to have a report that in itself is ready for evaluation on the 20/12-01 and then it may be that final preparations and testing of the product may have to be conducted in January and February prior to shipment of the satellite in April. Especially this means that environmental testing such as e.g. radiation tests or vacuum test have low priority.

### 3.2.5 Assumptions

The values for ”shutdown current” in table 3.1 have not been finally specified and will first be available when each of the users have been tested finally. The numbers given are the maximum currents plus 10 %.

## 3.3 Requirements on Interfaces and their Functionality

The following subsections each describe an interface that is associated with the PSU. The purpose of the interface is first described where after physical implementation and functional behavior are explained.

### 3.3.1 Power Distribution Interface

The purpose of this interface is to distribute power to each user and provide user protection for each user. This interface consists of one power wire and a common wire for each user. The voltage difference between the two lines are 5.0 V for all users.

#### 3.3.1.1 Output Voltage Control

The voltage drop between the two output terminals must be controlled so that it under normal operation remains within 1 % of accuracy, i.e. between 4.95 V and 5.05 V. When users are shifted on or off the powerbus then deviations of ±5 % are accepted for a duration of less than 20 ms, i.e. between 4.9 V and 5.1 V

#### 3.3.1.2 User Protection

Each connected user has a specified maximum current that it is allowed to consume, if this limit is exceeded then the user is disconnected from the power-bus by the PSU. This event is informed to the DHS and the user is turned on again when requested by the DHS. This communication uses the communication interface that is described later.

#### 3.3.1.3 User Protection of the OBC

Since the DHS is situated on the OBC the above protection mechanism cannot be used, because obviously the DHS cannot tell the PSU to turn on the OBC again. Therefore, another mechanism is used for the OBC user protection.

If the OBC-user current exceeds the threshold it is shutdown by the PSU and then again turned on when 5 minutes have elapsed. After turn-on the event is communicated to the DHS. The waiting is meant to allow the temperature of the OBC-circuitry to drop in case the error condition was due to thermal effects.

### 3.3.2 DHS Communication Interface

The purpose of this interface is to provide DHS with information from the PSU. This information includes:

- Information on power-status
- Information on status of users

- General housekeeping information from PSU

It is also possible for the DHS to shut users on or off using this interface. The physical interface is the I$^2$C-bus that is chosen as the main internal communication bus of the satellite. The software protocol specification for the communication is given in appendix G on page 201.

#### 3.3.2.1 Communication of House-keeping Information

The following data is available for the DHS (numbers in parenthesis shows the number of measurements):

- Voltage across parallel connection of solar-panels (1)
- Currents through solar-panels on each side of the satellite (5)
- Voltage across parallel connection of batteries (1)
- Voltage on 5 V power bus (1)
- Currents to each user (OBC, ACS, CAM, TRD & PSU) (5)
- Temperature measurements (6)

Above values are all described with 12 bit words, except for temperatures which are 8 bit, and information is updated on the PSU at least once every 10 seconds. The temperature measurements will be used to measure hot-spots in the satellite and this gives about one or two temperature measurements for each user.

#### 3.3.2.2 Communication of User On/Off status

If the PSU user-protection mechanism shuts down a user because it exceeds its maximum current this is communicated to the DHS in order to notify it of the event and identify the user that was shutdown. If the PSU has shut down the OBC and turned it on again after the specified period of time then this event is communicated to the DHS after reactivation in order to inform about the event.

The DHS can by sending a signal order the PSU either to shut down or turn on a connected user. This does however not apply for the OBC-user, which can only be shut down by the PSU in case of a protection fault. DHS can at all times request information on which users are turned on or off.

### 3.3.3 OBC Boot Selection Interface

The purpose of this interface is that the OBC needs an external subsystem to help it select the boot-mode for the OBC. Further the OBC needs an external watchdog. Since it is the PSU that must cycle the OBC power in case of a malfunction of the OBC, then the PSU has been chosen to implement this functionality.

The physical interface consists of a boot-selection port which is simply a connection from an output port on the PSU digital hardware to the OBC. Further, the I$^2$C-bus is used for both the watchdog functionality and to set the value of the boot-selection port. The OBC-groups documentation on this interface can be found in [01gr732, 2001].

#### 3.3.3.1 Boot Sequence Algorithm

When the satellite is deployed in space or if the OBC has malfunctioned and as a result been shut down by the PSU (as described in 3.3.1) a special boot-sequence is initiated by both PSU and OBC in order for the OBC to select whether it is to boot from PROM or EEPROM, and in order to control if the boot succeeded. The algorithm is specified in figure 3.2.

The **Signal()** command is sent through the I$^2$C-bus and is specified in the software protocol in appendix G on page 201. The time-out period is specified to 10 seconds.

#### 3.3.3.2 External Watchdog Timer

When the OBC is successfully booted the timer on the PSU continues to run and is reset every time the **Signal()** command is sent from the DHS on the OBC to the PSU through the I$^2$C-bus. If the DHS fails to call the command before the timer has expired then the OBC is shut down and the above boot sequence is initiated.

#### 3.3.3.3 Selection of Boot Mode

The DHS can at anytime read and change the value of the boot-selection-port by means of commands sent through the I$^2$C-bus. These commands are specified in appendix G on page 201

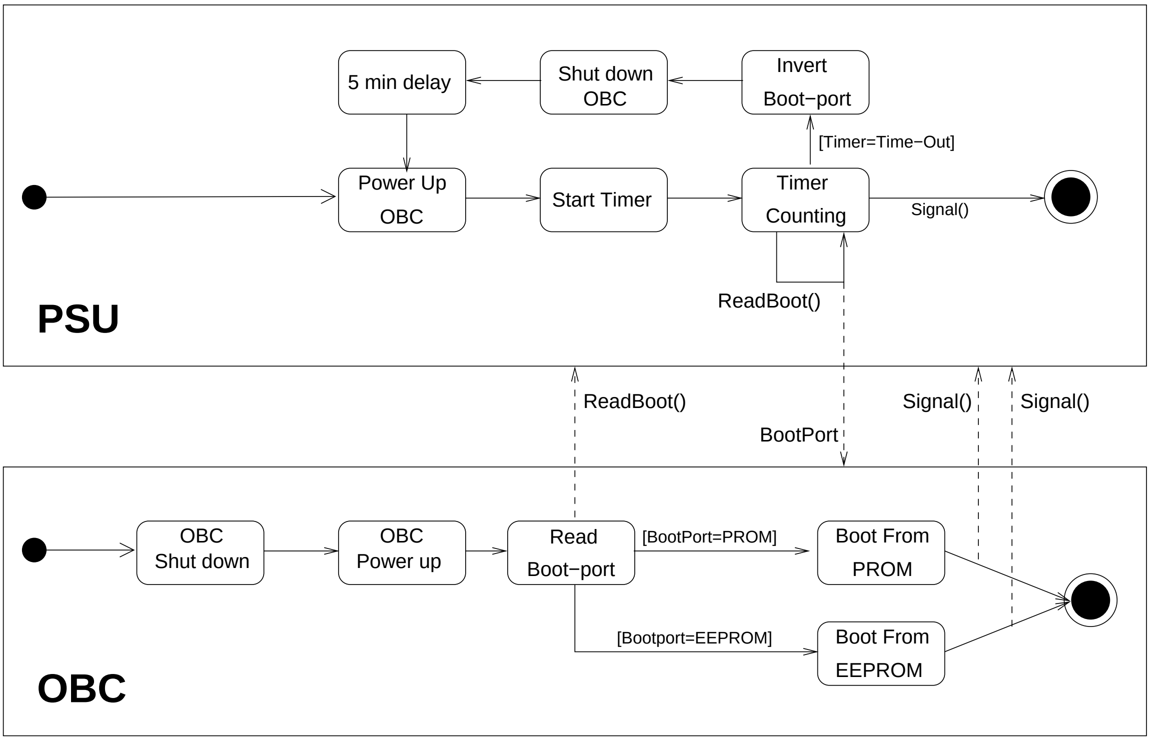


**Figure 3.2:** The Boot Sequence interaction between the PSU and the OBC specified in UML

## 3.4 Quality of Service Requirements

The following will specify requirements for the quality of the services provided by the PSU in the cases where it is not already covered by the functional requirements.

### 3.4.1 Requirements on Efficiency

No exact requirement on efficiency of the PSU has been formulated, but it is clear that the better efficiency of the PSU the more power is available for the satellite and a higher degree of activity can therefore be maintained. In order to set forth a goal the group has decided to design for an overall efficiency from input from solar cells to output to users of minimum 75%.

### 3.4.2 Accuracy of Housekeeping Data

The purpose of collecting housekeeping data is to have data from which one can evaluate the performance of the satellite and detect cause of possible errors. The accuracy of these data are not critical as long as these data are reasonable values with an accuracy that does not vary in large degree with environmental conditions such as temperature. No exact requirement on the accuracy will be given, but is considered to be a best effort requirement.

### 3.4.3 Requirements on Reliability and Fault Tolerance

The following states requirements regarding reliability and fault tolerance of the power supply.

#### 3.4.3.1 Operation without Batteries

In the case where the batteries are either fully discharged or if they are not working at all, the PSU must be able to start up and operate normally from the solar-cells alone. If the batteries are fully discharged this will allow the PSU to start up every time the satellite is in the sun and charge the batteries until a minimum working voltage is reached.

#### 3.4.3.2 Single Event Upsets and Latch-Up

Due to the possibility of system failure as a result of either a single event upset or a latch-up (as described in section 2.1 on page 21) it is required that the power supply implements an internal watchdog-timer that will restart the system if the system fails to respond. It is also required that the current consumption of the PSU is monitored and the system is restarted on excessive consumption.

Other measures against failure due to SEU's should be utilized if feasible methods exist that can be implemented with regard to time-schedule and other requirements.

#### 3.4.3.3 Required Lifetime

The Satellite is specified to survive in the space environment for at least a year [01gr930, 2001]. Therefore this requirement applies to the PSU as well.

## 3.5 Environmental Requirements

The following will specify requirements originating from the environment in which the PSU is to be deployed, i.e. within the satellite.

### 3.5.1 Physical Dimensions

The maximal physical size of the PSU is constrained by requirements put on the PSU by the group working with mechanical design of the satellite and the requirements stated here are taken from their documentation [MK9-P22A, 2001].

Both batteries and solar-panels have been accepted by the mechanical group and they are responsible for placing these on the satellite. The Printed Circuited Board (PCB) for the PSU is specified to a weight below 30 g and physical dimensions of maximally 95.2 mm x 82.2 mm x 16.5 mm. A complete technical drawing of the maximum allowable dimensions of the PCB board is given in appendix M on page 231.

### 3.5.2 Thermal Requirements

As of the date of the delivery of this report, no complete thermal analysis has been conducted for the AAU-Cubesat and there is therefore no final thermal requirements for the satellite, but experience from other Cubesat projects suggests that a temperature range of -40°to 80°can be expected internally in the satellite.

Therefore, this temperature range will be the requirement of this project until other results are provided for the AAU-Cubesat. This temperature range corresponds to what components graded for industrial use can withstand.

### 3.5.3 Radiation Requirements

All parts of the PSU have to be able to withstand the radiation in space. The amount of this radiation is, in chapter 2 determined to be about $1 \cdot 10^6 Rad(Si)$.

### 3.5.4 Vacuum

The PSU must be able to withstand absolute vacuum since it is to be deployed in space. Apart from being able to function in vacuum there may be no detectable out-gassing of matter from any of the PSU components.

### 3.5.5 Acceleration Requirements

The conditions during launch require that all parts of the satellite can withstand accelerations of 15 g [Connolly, 2000].

# Chapter 4
# System Analysis

In this chapter a more general description of the Power Supply Unit (PSU) will be given and an analysis of different topologies will be given. The chapter begins by describing the purpose of the system, whereafter the blocks of the system will be defined and some design solutions will be presented. One of these solutions will be chosen for this project. At the end of the chapter the hardware and software control solutions will be described.

## 4.1 The Functionality of the PSU

The overall purpose of the PSU is to transfer power from the solar array to the users. This could be done with one converter. Sometimes the satellite will be in shade and therefore, in order to fulfill its mission, the PSU has to be connected to a battery pack. This battery can be controlled by another converter, but this it is possible to make an overall solution with only one converter.

The overall function of the PSU can be seen in figure 4.1. This figure shows the solar array on the left producing the power which will be transfered to the PSU. In situations with insufficient power coming from the solar array, the battery pack has to be connected and then supply the users. If a situation occurs where a user is using to much current because of a malfunction, the PSU must cut off the connection to the user. The PSU has to communicate the malfunction to the on board computer. This is done through a communication interface. This interface also provides the possibility to send other information e.g. battery status. The PSU has to carry out Maximum Power Point Tracking (MPPT), which ensures that the solar array operates at its best all time and thereby produces maximum power. The PSU should also be able shut down all the users. The PSU must be able to start up by its own. In order to be able to create a system which fulfill the demands mentioned above, it must to be split into three parts:

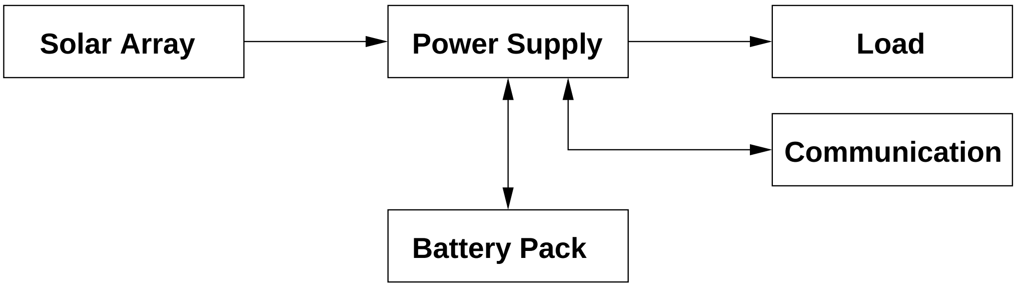


**Figure 4.1:** The overall purpose of the PSU

- The Converter Hardware
- The Control Hardware
- The Control Software

## 4.2 Converter Hardware Solutions

In this section hardware system solutions for converter topologies in the PSU will be presented. The purpose of this part is to connect the solar array to the battery pack and the power bus with the appropriate converter circuits between them. This gives three major building blocks:

1. Solar array
2. Battery pack
3. Converter

The solar array, discussed in appendix B, is the only longterm power income of the satellite. The battery pack is needed to store energy, so it can be used when the energy from the solar cells is insufficient. The battery for the AAU-Cubesat is described in appendix D on page 185.

A converter is an electronic circuit capable of changing a voltage level into another, higher or lower (depending on converter type). This is used e.g. to provide the correct voltage to the OBC. To control each of these converters a $\mu$controller unit (MCU) is used.

To find out different solutions to arrange the blocks the group members have made a brainstorm. The results of this brainstorm is illustrated in 4.2.

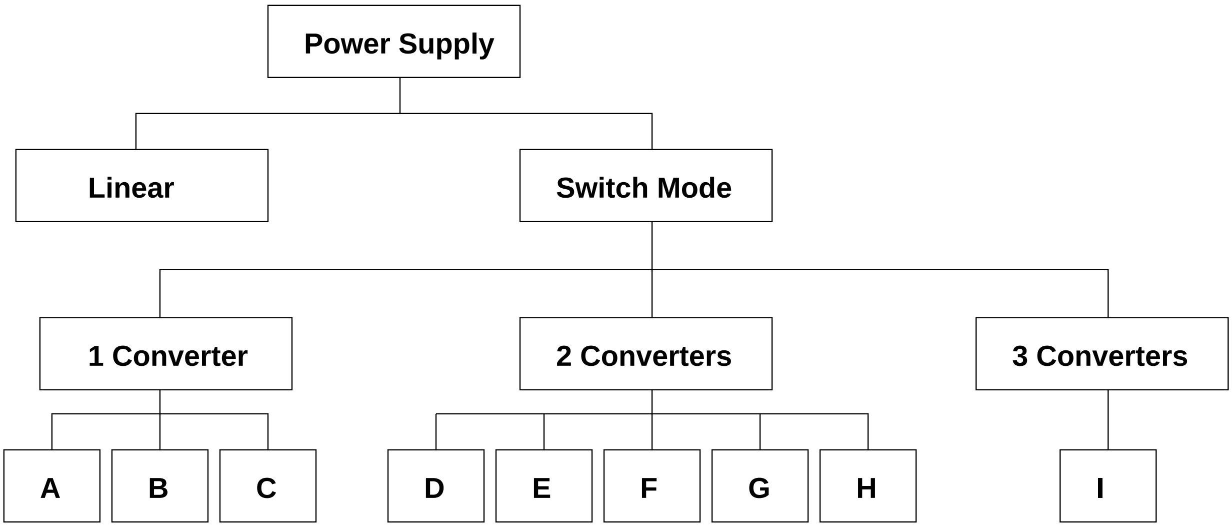


**Figure 4.2:** Solution tree

In the first block there is the PSU and under it there are two solution options, linear and switch mode. It has been decided to work further with the switch mode topology, because the linear PSU have great loss of power. Under the switch mode power supply there were found solutions with one, two or three converters; each of them will be described below.

**Solutions with one Converter**

The solutions with one converter are described below, where there are three blocks: solar array, battery pack and a converter. These are shown in different topologies in figure 4.3.

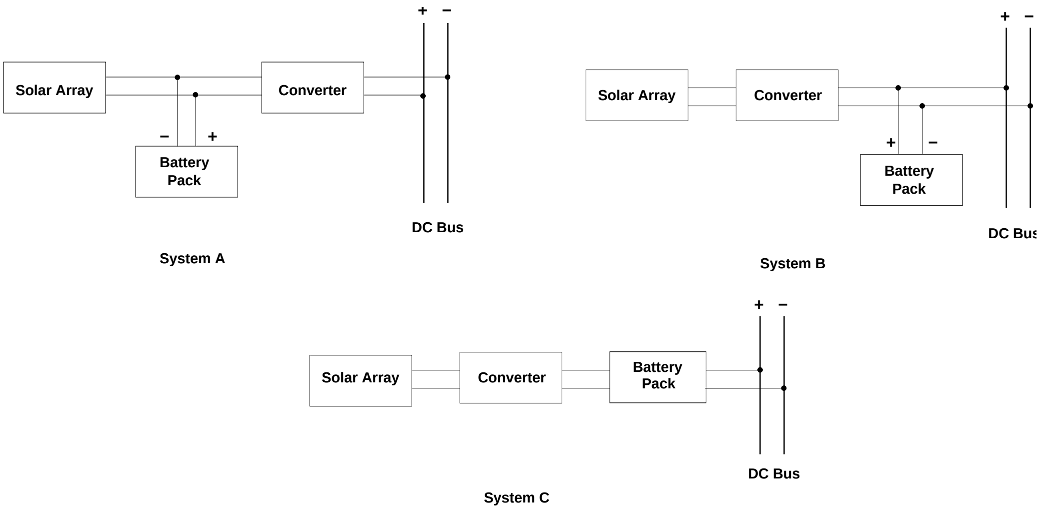


**Figure 4.3:** Possible solutions with one converter

- Solution A: The disadvantage of this solution is that it is not possible to make maximum power point tracking and constant voltage on the bus at the same time with just one converter. Therefore this solution will not be discussed.

- Solution B: The solar array are connected directly to the converter and the battery pack is connected after the converter together with the bus. The disadvantage of this system is that the voltage level on the bus must be the same as the battery voltage level. Furthermore the converter has to control the charging of the battery pack.

- Solution C: Before the battery pack there is one converter. This solution does not fulfill the possibility of maximum power point tracking because of the same reason as in solution A. Therefore this solution is not to be considered.

Due to the fact that the energy only has to travel through one converter, the power loss is restricted to only that. Another advantage of the systems with only one converter is the limited weight that it adds to the satellite.

**Solutions with two Converters**
In the system solutions described below, see figure 4.4, the blocks will be lined up in different ways with two converters, solar array and a battery pack.

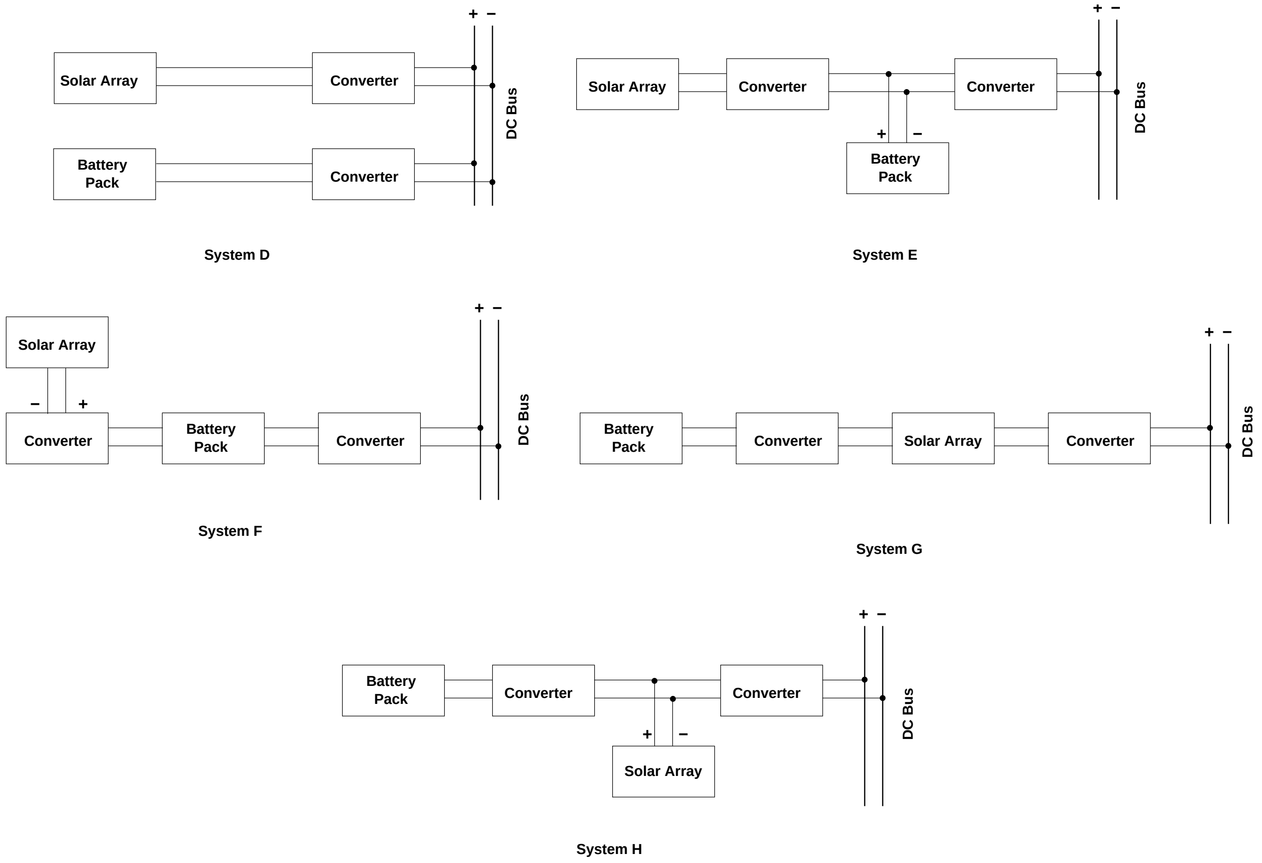


**Figure 4.4:** Solution with two converters

- System D: This system is very reliable. If one converter fails the other one will continue supplying power (thus this will be limited by the operation of the power source). A disadvantage is that if the power has to be stored before use and therefore it has to travel through three converters. Through two converters, when it is in charge mode and through one converter when the battery pack is supplying the DC-bus.

- System E: In this system there is a short way from the solar array to the battery pack. This ensures less power loss in the charging phase. The disadvantage is that there is a long way from solar array to DC-bus. This system has the possibility to keep providing power from the battery pack, even if the solar array fails.

- System F: In this system all the blocks are conneted together in series. The output from the first converter and battery pack is joined in the second converter. This solution have the disadvantage that if the batteries have to be charged then the polarization of the batteries have to be turned.

- System G: The battery is connected to one converter and the solar array are connected between the two converters. A disadvantage is that when the solar cells are not producing any power, their internal PN junction will result in a power loss.
- System H: In this system the battery is connected to a converter and the solar cells are connected between to converters. The advantage of this system is that there is a short way from the solar cells to the DC-bus. The disadvantage in this system is that there are no possibility for maximum power point tracking.

The systems with two converters can use the energy from the solar cells better then the systems with only one converter. This is mainly due to the fact that one converter can control the MPPT, while the other one control the output voltage to the DC-bus. From the above solutions D, E and G will be considered.

**Solutions with three Converters**

System I: This system (see figure 4.5) is similar to system D except from the converter between the solar array and the battery pack. The main advantage of this system is that it is possible to supply power even if one of the converters fail. Another advantage is that no matter where the power is sourced from there will only be a power loss through one converter under normal condition. A disadvantage is that the overall system requires three converters, which all must be able to function in both directions. This type of solution with three converters also requires a lot of control. Weight should also be considered, when building the system. This system is also to be considered.

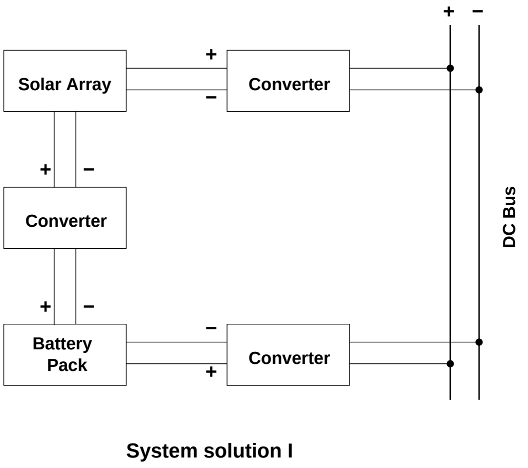


**Figure 4.5:** Solution with three converters

### 4.2.1 Choice of Solution

To find out which of the four solution is the best option (solution D,E,G and I), they were put in a table as shown in figure 4.1. In this method of chosing a design solution, the system solutions were listed horizontally and criterias vertically. Then the weight of the criterias were defined in the scale from 1 to 5 (were 1 is lowest and 5 highest). The criterias are representative of the groups oppinion of what is important in a space based power supply. After each criteria had been given weight the relative scores were filled in for each solution. Again these weights are an expression of the groups oppinion on how important these criteria are. The scores were from -2 to +2. In the end the total score of each solution was calculated.

- Reliability is how the system works if some of the circuits fails
- Mass is how much space and weight the PSU takes up.
- Efficiency, how much of the incoming power will result in outgoing power
- Complexity, how many converters are there and how many of them should work in two directions.
- MPPT, is the PSU’s ability to keep the MPP
- Battery charging, is how easy it is to control charge/discharge

| | Weight | D | E | G | I |
|---|---|---|---|---|---|
| Reliability | 3 | +1 | -1 | -1 | +2 |
| Mass | 4 | 0 | 0 | 0 | -2 |
| Efficiency | 3 | -1 | 0 | -1 | 0 |
| Control | 3 | 0 | +2 | +1 | +1 |
| Complexity | 2 | 0 | +1 | -1 | -2 |
| MPPT | 4 | +2 | +2 | -1 | +2 |
| Battery Charging | 1 | +2 | -1 | +2 | +2 |
| Total | | 10 | 11 | -6 | 9 |

**Table 4.1:** The vote of the different converter solutions

- Total, the weights multiplied with the sum of each column.

As seen in the table 4.1 solution E got the highest score, so this solution will be used in the power supply.

## 4.3 Control Hardware Solution

The purpose of this part is to provide hardware capable of controlling the converters and gather information (voltages and currents) about the status of the PSU and communicate this information to the DHS. [1] Thus this part consist of two tasks:

1. Control
2. Housekeeping data

In order to do so, two kinds of building blocks are needed: A MCU, and sensors. These sensors are to measure voltages, currents, and temperatures. The MCU is the brain of the PSU doing the calculations necessary to control the converters. The voltage and current sensors are needed both to gather information to be used in the control of the converter hardware and also for housekeeping, while the temperature sensors are purely for housekeeping. The interface between the sensors and the MCU is an ADC either externally or internal in the MCU.

### 4.3.1 Control

This is the primary task of the control hardware and it can be divided into three parts: Measuring, calculation and control. The measuring part must measure the necessary information. The calculation part must use the measured information to calculate the control. The control part interfaces with the converter hardware, using the calculated numbers. This division can be seen i figure 4.6.

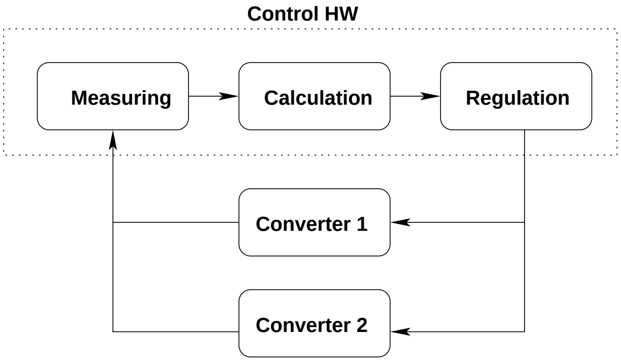


**Figure 4.6:** The control part of the control hardware

### 4.3.2 Housekeeping

The housekeeping task can be divided into two parts: Measuring and communication. The task of the measuring part is to gather information from the appropriate point in the converver hardware while the task of the communication part is to relay this information to the OBC. This division can be seen in figure 4.7.

[1] Data Handling Software

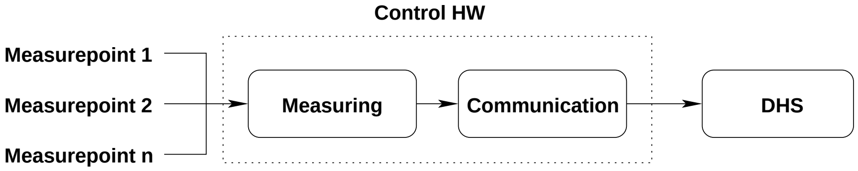


**Figure 4.7:** The housekeeping part of the control hardware

## 4.4 Control Software Solution

The purpose of the control software is to use the tools provided by the control hardware in order to: control the output of the PSU, gather housekeeping data and control which of the connected loads are supplied with power. Further, in order to know which loads to turn on or off and to communicate the housekeeping data a communication task is needed that communicates with the DHS.

This implies that there are four kinds of independent tasks to be performed asynchronously and in order to do so an operating system (OS) is needed to schedule task execution. Since there are two converters in the chosen system (see section 4.2.1) there is a need for two control tasks, one for each converter. This leaves us with the following basic tasks to run on the OS:

- 2 x Control
- Subsystem on/off control
- Gathering of housekeeping data
- Communication
- Watch dog timer

The OS that will be used cannot be a normal OS since both the control and the housekeeping must be executed in real time. This implies that the OS must be a real-time operating system (RTOS) or that the software must be implemented directly without the use of an underlying OS.

# Part II

# Design of Hardware and Control Loops

# Chapter 5
# Hardware Overview

## 5.1 Overview

In this part of the report the hardware for the Power Supply Unit (PSU) will be designed. This chapter will provide an overview of the hardware configuration by extending the information given in the ”System Analysis” with more details regarding the configuration of solar arrays and battery pack. The PSU is depicted in figure 5.1 with all hardware modules. The following sections will state the functionality of the different modules that make up the PSU.

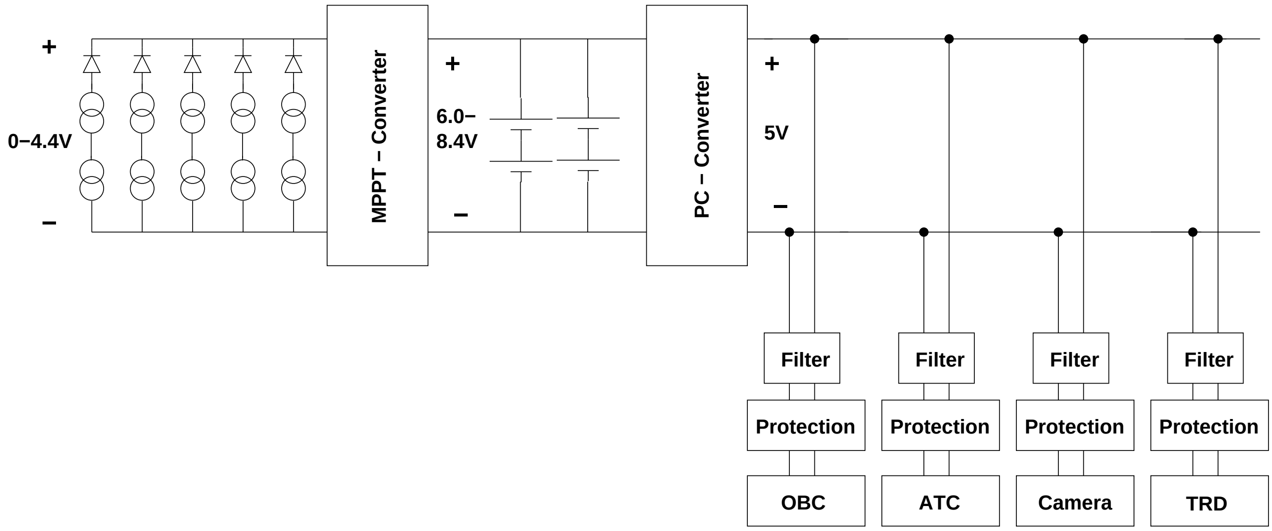


**Figure 5.1:** An overview of the PSU hardware configuration

## 5.2 Functionality

In this section a ”walk-through” of figure 5.1 from left to right and a brief discussion on the functionality and configuration of the different modules of the PSU hardware will be done.

### Solar Arrays

The solar arrays are configured such that the cells of each side on the satellite are connected in series and the five sides are connected in parallel. This configuration has been chosen, because it was found that it is the best tradeoff between ease of Maximum Power Point Tracking (MPPT) and to improve the converter performance. The analysis of the solar array configuration and MPPT can be found in appendix B on page 173.

The voltage across the parallel connection will in this configuration be approximately 4.0 V depending on illumination input (See appendix B on page 173). The diodes protect the cells from conducting a reverse current.

### MPPT Converter

The MPPT-converter (MPPTC) performs MPPT on the solar array i.e. it transfers the maximum obtainable power from the solar array to the outputside of the converter. Further, it steps up the voltage to fit the level of the battery pack.

### Battery Pack

The battery pack consist of 2 x 2 batteries connected in two parallel strings of two batteries in series. This gives a voltage of between 6.0 V and 8.4 V depending on the state of the batteries. The choice of battery configuration and capacity is documented in appendix D on page 185.

The battery is also responsible for voltage control of the intermediate powerbus between the two converters. The batteries keep the average voltage in the range 6.0 - 8.4 V.

**PC-Converter**
The PC-converter (PCC) conditions the voltage for the main 5 V powerbus by stepping down the voltage from the intermediate powerbus.

**Output Filters**
Finally each user is connected through a passive low-pass filter which filters the voltage specifically to ensure that noise from one user does not propagate to the main powerbus and thereby to other users. This problem is specifically severe when a user is either switched on or off the powerbus. Secondly the filter serves to limit inrush currents to the user when it is switched on.

**Protection Circuits**
Each load has an associated protection circuit which will shut down the user i.e. disconnect it from the powerbus if the load draws a current that is higher than a specified maximum for that load. This could happen if the load suffered from an internal latch-up due to radiation effects.

The load protection circuit also serves as a controllable on/off switch that lets the PSU MCU switch loads on and off according to the orders that it receives from the Data Handling Software (DHS) which the MCU communicates with through the $I^2C$-bus.

**Digital Hardware and Measuring**
On figure 5.1 the digital hardware is not shown. The digital hardware is responsible for the measuring of physical values in the system depicted in figure 5.1 and measurement of temperatures at various locations in the satellite. These measurements serve two distinct purposes. Firstly the measurements are used to control the two converters and secondly the measurements are sent to the DHS for housekeeping information.

## 5.3 Disposition of the Following Chapters

The following chapters will describe the design of the parts of the PSU in detail. The disposition is:

1. Maximum Power Point Tracking Converter
2. Power Conditioning Converter and Output Filtering
3. Protection Circuits
4. Controller Design
5. Digital Hardware

The two converter chapters will each take a different approach in designing the converter. The MPPT converter will be designed using conventional circuit analysis to derive key design formulas and then use these formulas to choose components. The power conditioning converter chapter will use design formulas from the literature and then try to model losses in non-ideal components in order to optimize converter efficiency.

The ”Controller Design” chapter is not about hardware design, but it develops the control model of the converters on basis of the hardware design that is necessary to specify measurement points that the digital hardware must be designed to handle. Finally in the digital hardware chapter the digital part of the system is designed as well as the analog circuitry that is needed in order to measure data for housekeeping information and converter control.

The next part of the report will use the hardware and control algorithms designed in this part to design and implement the software that controls the converters and protection circuits and communicates with the other subsystems on the satellite.

# Chapter 6
# Maximum Power Point Tracking Converter

## 6.1 Overview

The following chapter describes the selection, design and implementation of the Maximum Power Point Tracking Converter (MPPTC) that is located between the solar arrays and the battery pack as seen on figure 6.1. The main purpose is to draw the maximum current from the solar array and deliver the current the users need, when the batteries are fully charged[1].

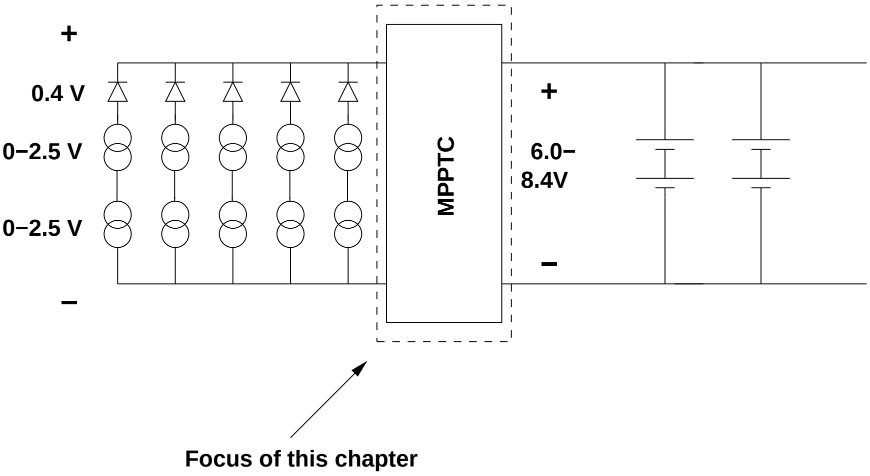


**Figure 6.1:** Block diagram of the PSU

**Input Characteristics**

According to appendix D on page 185 the solar arrays are connected as seen on figure 6.1. Each array consists of two solar cells connected in series and a diode. Each cell produces a maximum voltage of 2.5 V and the voltage over the diode is 0.4 V. Then the maximum input voltage to the MPPTC will be 4.6 V.

**Output Characteristics**

The output from the MPPTC is connected to the intermediate power bus. The battery pack, that consists of four batteries connected two and two in series will be connected to the bus. To prevent the batteries from damage they must have a voltage level varying between 6.0 - 8.4 V. If the voltage on the intermediate power bus crosses the maximum voltage level (8.4 V) the current that flows into the battery will be too high. To avoid this situation the converter should not utilize MPPT when the voltage level rises above this threshold voltage.

### 6.1.1 Specific Requirements

Following are the specific requirements for the MPPTC stated.

1. The output must be between 6.0 V - 8.4 V (see appendix D on page 185)

2. The output ripple must be less than $2\%_{p-p}$. This requirement has been set in order to improve the life time of the batteries, which wears down faster the more charge-discharge cycles it undergoes (see appendix D on page 185)

3. The converter efficiency should be 90% or better in order to help fulfill the overall efficiency requirement of 75% (see subsection 3.4.1 on page 28)

[1] Intermediate power bus is the bus between the two converters

## 6.2 Design of the MPPTC

Figure 6.1 illustrates where the MPPT converter is placed in the PSU topology. The solar array supply a maximum voltage of 4.6 V. As mentioned in 6.1 the batteries are placed in series giving a maximum voltage of 8.4 V. The task of the converter is to draw the maximum current from the solar array and step up the voltage. Of the general converters there are several that are able to handle that task. The Boost converter and the Buck-Boost converter will be described.

### 6.2.1 Boost Converter

Figure 6.2 shows a diagram of the Boost converter.

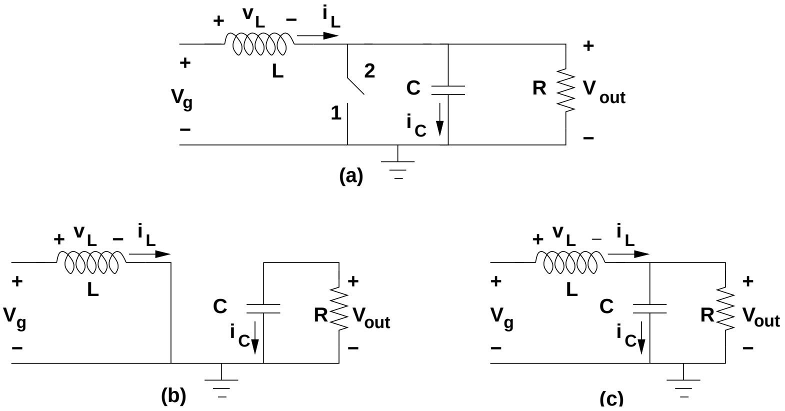


**Figure 6.2:** (a) Diagram of the Boost converter. (b+c) Diagram of the Boost converter with the switch respectively in position 1 & 2. R is a variable load that models the impedance conditions on the intermediate power bus

With the switch in position 1, see figure 6.2 (b), the inductor is connected to common and the current rises in the inductor and the capacitor is discharged through the resistor. When the switch is in position 2, see figure 6.2 (c), the inductor is a current source. The Boost converter draws a continuous current from the solar array, which is efficient because the current is from the solar array is drawn in both switch situations.

### 6.2.2 Buck-Boost Converter

Figure 6.3 shows a diagram of the Buck-Boost converter.

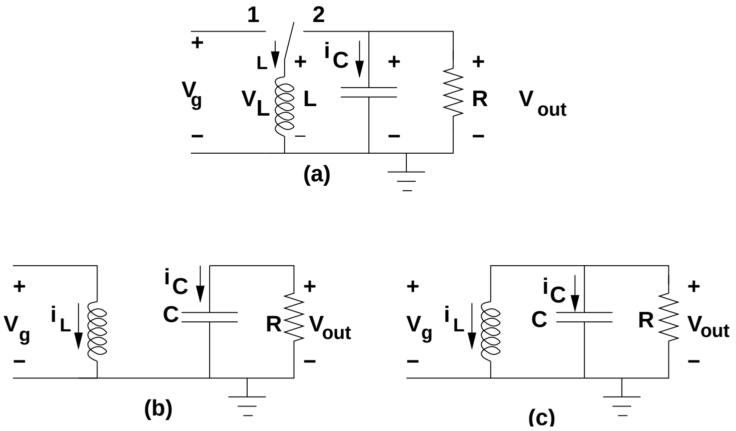


**Figure 6.3:** (a) Diagram of the Buck-Boost converter. (b+c) Diagram of the Boost converter with the switch respectively in position 1 & 2

In position 1 the inductor current is drawn from the solar array, but when the switch is in position 2 the solar array is switched out and the inductor now serves as a current source. This means that the current from the solar array is not used when the switch is in position 2 and the maximum power from the solar cells cannot be drawn in this case. The current is reversed in the Buck-Boost converter, and therefore the voltage is reversed. The Buck-Boost converter is able to step the voltage up and down.

### 6.2.3 Choice of Converter Type

The Boost converter is more suited to the MPPT than the Buck-Boost converter because a continuous current is drawn from the solar array in both switch situations. Besides the need to step down the voltage will never arise. For these reasons it is decided that the MPPTC will be implemented as a Boost converter.

## 6.3 MPPTC in Steady State

Figure 6.2 shows the Boost converter when it is operating. The voltage across the inductor and the current through the capacitor is analyzed. With the switch in position 1 the voltage across the inductor and the capacitor current is:

$$\begin{aligned} v_L &= v_g \\ i_C &= -i_{out} = -\frac{v_{out}}{R} \end{aligned}$$

And with the switch in position 2:

$$\begin{aligned} v_L &= v_g - v_{out} \\ i_C &= i_L - i_{out} = i_L - \frac{v_{out}}{R} \end{aligned}$$

In a well designed converter in steady state there should be no significant ripple on the current and voltage during one switching period. Therefore, it is assumed that the switching ripple is much smaller than the DC components of the current and voltage. This means that the current and voltage are the same at the beginning and at the end of a switching period. This is the small ripple approximation and it leads to the following expressions for the inductor voltage and capacitor current during a switching period $T_s$.

$$\int_0^{T_s} v_L dt = 0$$
$$\int_0^{T_s} i_C dt = 0$$

Therefore, expressions for the inductor voltage and capacitor current in both switch situations are evaluated and considered over one switching period. The switch in position 1:

$$\begin{aligned} v_L &= V_g \\ i_C &= -\frac{V_{out}}{R} \end{aligned}$$

The switch in position 2:

$$\begin{aligned} v_L &= V_g - V_{out} \\ i_C &= I_L - \frac{V_{out}}{R} \end{aligned}$$

The quantities with the capital letters are assumed to be constant and therefore the voltage across the inductor and current through the capacitor are now assumed to be constant. This is used when the inductor voltage and capacitor current are considered over one switching period. First the inductor current is considered. The time-interval $0 - DT_s$ is when the switch is in position 1 and $DT_s - T_s$ corresponds to position 2.

$$\begin{aligned} \int_0^{T_s} i_c dt &= 0 \Leftrightarrow \\ \int_0^{DT_s} -\frac{V_{out}}{R} dt + \int_{DT_s}^{T_s} I_L - \frac{V_{out}}{R} dt &= 0 \Leftrightarrow \\ -\frac{V_{out}}{R} \cdot D + \left( I_L - \frac{V_{out}}{R} \right) \cdot (1 - D) &= 0 \Leftrightarrow \\ I_L - \frac{V_{out}}{R} - I_L \cdot D &= 0 \Leftrightarrow \\ I_L &= \frac{V_{out}}{R} \cdot \frac{1}{1 - D} \end{aligned} \qquad (6.1)$$

With this formula the inductor current can be calculated, and the ripple on the inductor current is a percentage of $I_L$. Similar calculations are made to determine the voltage across the capacitor, which equals the output voltage.

$$\begin{aligned} \int_0^{T_s} v_L dt &= 0 \Leftrightarrow \\ \int_0^{DT_s} V_g dt + \int_{DT_s}^{T_s} V_g - V_{out} dt &= 0 \Leftrightarrow \\ V_g \cdot D + (V_g - V_{out}) \cdot (1-D) &= 0 \Leftrightarrow \\ V_g - V_{out} \cdot (1-D) &= 0 \Leftrightarrow \\ V_{out} &= V_g \cdot \frac{1}{1-D} \end{aligned} \tag{6.2}$$

As for the ripple on the inductor current the ripple on the output voltage can be calculated as a percentage of $V_{out}$. Next the ripple on the inductor current and capacitor voltage in steady state is examined closely. Figure 6.4 sketch the ripple on the inductor current. The ripple current is not linear, but it is a very small AC-signal, which will be assumed linear. This can only be done because of the small magnitude of the AC-signal.

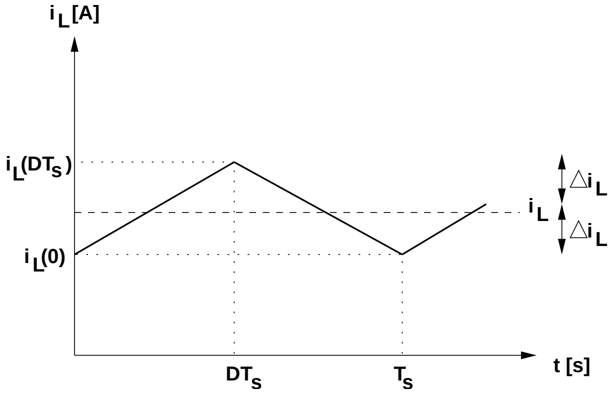


**Figure 6.4:** Ripple on the inductor current

It can be seen on figure 6.4 that the size of the ripple can be calculated as:

$$\Delta i_L = \frac{1}{2}(i_L(DT_s) - i_L(0)) \tag{6.3}$$

From the definition of the inductor voltage the slope in the first subinterval can be calculated:

$$\begin{aligned} v_L(t) = L\frac{di_L(t)}{dt} \Leftrightarrow \\ \frac{V_g}{L} = \frac{di_L(t)}{dt} \end{aligned} \tag{6.4}$$

It is used that $v_L(t) = V_g$ in the first subinterval. With a slope in the first subinterval of $\frac{V_g}{L}$ the current at the end of first period is $i_L(DT_s) = i_L(0) + \frac{V_g}{L} \cdot (DT_s)$. If this is inserted in formula 6.3 the ripple current is

$$\begin{aligned} \Delta i_L &= \frac{1}{2} \cdot \frac{V_g}{L} \cdot DT_s \Leftrightarrow \\ L &= \frac{V_g}{2\Delta i_L} \cdot DT_s \end{aligned} \tag{6.5}$$

This expression is used to determine the inductance of the inductor. With a similar approach an expression for the capacitor voltage can be derived. Figure 6.5 sketches the ripple on the capacitor voltage. The capacitor current is $i_C = C \cdot \frac{dv_C}{dt}$. In the first subinterval of the switching period that is:

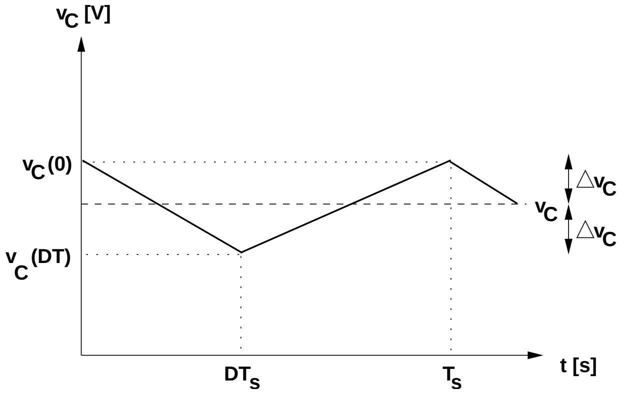


**Figure 6.5:** Ripple on the capacitor voltage

$$\begin{aligned} i_C &= C \cdot \frac{dv_C}{dt} \Leftrightarrow \\ \int_{\Delta v_c}^{-\Delta v_c} dv_C &= \frac{1}{C}\int_0^{DT_s} i_C dt \Leftrightarrow \\ -2\Delta v_C &= -\frac{1}{C}\cdot\frac{V_{out}}{R}DT_s \Leftrightarrow \\ C &= \frac{V_{out}}{2\Delta v_C R}DT_s \end{aligned} \tag{6.6}$$

In the calculations it is used that the capacitor current equals $i_C = -\frac{V_{out}}{R}$ in the first subinterval of the switching period. Finally an expression for the duty cycle is needed. This can be derived by formula 6.2.

$$D = 1 - \frac{V_g}{V_{out}} \tag{6.7}$$

### 6.3.1 Determination of L and C

For the determination of L and C the worst case situation is considered. The switching frequency is set to 50 kHz and the efficiency is estimated as high as possible, that is 100%. The ripple on the inductor current is set to 10%, and the ripple on the output voltage is set to 2% to protect the batteries. To get realistic values of the components it is assumed, that the converter gets no input power and is shut down in the shadow. This is done because a very low current at the input is equivalent to a huge load resistance i.e. an open circuit. A very low inductor current also leads to large inductance of the inductor, if the converter is going to boost up the voltage.

The MPP is in the interval 3.4 V - 4.2 V, see appendix B on page 173. The batteries on the output of the MPPTC is considered as a voltage source between 6.0V - 8.4V, which changes slowly. Therefore, the output voltage will be considered constant when MPPT is utilized. The highest input power is 2.54 W, when the converter is utilizing MPPT in the sun, see appendix C on page 179. But in case of illumination from infrared radiation the power rises. The highest input power is estimated to 3.0 W.

In space the only parameter in the converter that can be changed is the duty cycle. Thus, the components have to be suited to the span of the duty cycle. Therefore, the component values corresponding to the highest and lowest duty cycle is calculated. It should be stressed that the duty cycle can be forced beyond the steady state interval.

#### High Duty cycle

This duty cycle corresponds to the situation where the output voltage from the MPPTC is 8.4 V and MPPT is performed. From equation 6.7 it is clear that the largest conversion is being made when $V_g = 3.4\ V$ and $V_{out} = 8.4\ V$.

$$D_{max} = 1 - \frac{V_{G,min}}{V_{out,max}} = 1 - \frac{3.4\ V}{8.4\ V} = 0.595 \tag{6.8}$$

Then the equivalent load resistance is calculated. The equivalent load resistance can be expressed as:

$$P_{out} = \frac{V_{out}^2}{R} \Leftrightarrow R = \frac{V_{out}^2}{P_{out}} \tag{6.9}$$

In this case the output voltage is $V_{out} = 8.4\ V$ and the output power is $P_{out} = 3.0\ W$. If this is inserted into equation 6.9 it yields:

$$R = \frac{(8.4\ V)^2}{3.0\ W} = 23.5\ \Omega$$

This result and the result from equation 6.8 is inserted into equation 6.1 giving:

$$I_L = \frac{V_{out}}{R} \cdot \frac{1}{1-D} = \frac{8.4\ V}{23.5\ \Omega} \cdot \frac{1}{1-0.595} = 0.8826\ A$$

Since the ripple must be maximum 10 % the maximum ripple current is:

$$\Delta I_L = 0.10 \cdot 0.8826\ A = 0.08826\ A$$

Now the inductor inductance can be determined by equation 6.5:

$$L = \frac{V_g}{2\Delta I_L} \cdot DT_s = \frac{3.4\ V}{2 \cdot 0.08826\ A} \cdot 0.595 \cdot 20\mu s = 229\ \mu H$$

The maximum ripple on the output voltage, which equals the capacitor voltage, is $2\%_{p-p}$ which gives a maximum ripple on the voltage of 1%:

$$\Delta V_C = 0.01 \cdot 8.4\ V = 0.084\ V$$

Now the capacitance of the capacitor is determined using equation 6.6:

$$C = \frac{V_{out}}{2\Delta V_C \cdot R} \cdot DT_s = \frac{8.4\ V}{2 \cdot 0.084\ V \cdot 23.5\ \Omega} \cdot 0.524 \cdot 20\ \mu s = 22.3\ \mu F$$

**Low Duty cycle**

Now the output voltage is 6.0 and MPPT is performed. The lowest conversion is now calculated according to equation 6.7. This is when $V_g = 4.2\ V$ and $V_{out} = 6.0\ V$. Similar calculations as for the high dutycycle is now made for the low dutycycle giving the results listed in table 6.1, where the results from high dutycycle are listed as well.

| Constants | High dutycycle | Low dutycycle |
|---|---|---|
| $D$ | 0.595 | 0.30 |
| $R$ | $23.5\ \Omega$ | $12.0\ \Omega$ |
| $I_L$ | $0.8826\ A$ | $0.7143\ A$ |
| $\Delta I_L$ | $0.08826\ A$ | $0.07143\ A$ |
| $L$ | $229\ \mu H$ | $176\ \mu H$ |
| $\Delta V_C$ | $0.084\ V$ | $0.060\ V$ |
| $C$ | $22.3\ \mu F$ | $19.4\ \mu F$ |

**Table 6.1:** Results for high & low dutycycle for the MPPTC

**Component Values**

From the calculated quantities of the inductor and the capacitor the larger one is chosen, because the components have to able to withstand the whole span of the duty cycle. Therefore $L = 229\ \mu H$ and $C = 22.3\ \mu F$ are chosen. In the implementation a capacitor with $C = 22\ \mu F$ will be used because of availability.

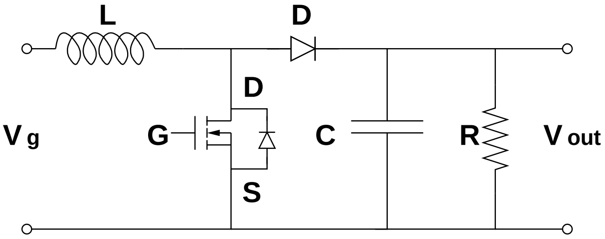


**Figure 6.6:** The implementation of the MPPTC

### 6.3.2 Discontinuous Conduction Mode

Figure 6.6 shows the realization of the MPPTC. The switch has been replaced by a MOSFET and a diode.

This configuration can however cause some problems. If the ripple on the inductor current is large enough it might cause the polarity of the diode or MOSFET to change. This phenomenon is called Discontinuous Conduction Mode (DCM), and it violates the calculations and assumptions made previously. The inductor current in DCM is illustrated in figure 6.7.

In the first subinterval the ripple on the current is positive and therefore the overall current through the diode is positive. In the second subinterval the ripple becomes negative and at a point of time the ripple is bigger than the diode current itself. Then the polarity of the diode changes making the previous calculations invalid. Therefore, an investigation of the probability of going into DCM is needed.

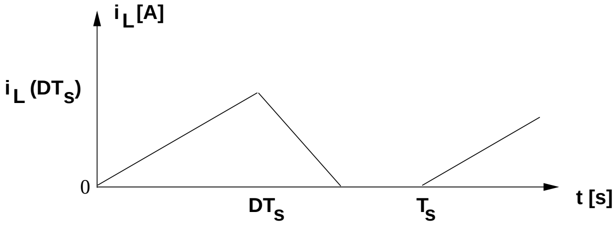


**Figure 6.7:** The inductor current in DCM

The diode current equals the inductor current when the diode conducts. If the ripple on the DC-component of the inductor current is larger than the ripple current itself, it is possible for the diode to become reverse biased ($I - \Delta I_L < 0$ for $\Delta I_L > I$). The DC-component of the inductor current can be expressed as [Erickson, 1999], page 112.

$$I = \frac{V_G}{R} \cdot \frac{1}{(1-D)^2} \tag{6.10}$$

To avoid DCM the following inequality needs to be true

$$I > \Delta I_L \tag{6.11}$$

Equation 6.5 and 6.10 is now inserted into 6.11

$$\begin{aligned} \frac{V_g}{R} \cdot \frac{1}{(1-D)^2} &> \frac{V_g}{2L} \cdot DT_s \Leftrightarrow \\ \frac{2L}{RT_s} &> D \cdot (1-D)^2 \end{aligned} \tag{6.12}$$

This is the condition that needs to be fulfilled to avoid evaporation in DCM.

#### DCM in High Duty Cycle

The quantities determined in section 6.3.1 for high dutycycle are inserted into inequality 6.12:

$$\begin{aligned} \frac{2 \cdot 229\,\mu H}{23.5\,\Omega \cdot 20\,\mu s} &> 0.595 \cdot (1-0.595)^2 \Leftrightarrow \\ 0.974 &> 0.098 \end{aligned} \tag{6.13}$$

It is seen that in high duty cycle the converter will not operate in DCM.

**DCM in Low Duty Cycle**

Now the quantities determined in section 6.3.1 for low dutycycle are inserted into inequality 6.12:

$$\begin{aligned}\frac{2 \cdot 229\ \mu H}{12.0\ \Omega \cdot 20\ \mu s} &> 0.30 \cdot (1 - 0.30)^2 \Leftrightarrow \\ 1.91 &> 0.147 \qquad (6.14)\end{aligned}$$

It shows that the converter will not operate in DCM in low duty cycle either.

## 6.4 Switch Design

To switch the converter between position 1 and 2, as shown on figure 6.2 on page 42 a gate driver will be used to switch the MOSFET. The gate driver will be supplied with a control signal from the digital hardware and voltage from the battery as shown on figure 6.8. The reason for using a gate driver instead of supplying the MOSFET directly from the MCU is to protect the MCU. The gate driver that has been chosen to the converter is MC33151 [MC33151, 1999].

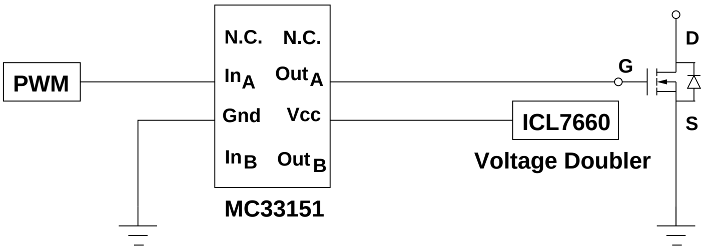


**Figure 6.8:** Gate driver connected to the MOSFET

### 6.4.1 Switching loss

The switching losses are calculated according to appendix E on page 191, and the loss in the diode and MOSFET is calculated.

**Power loss in the diode**

When the diode conducts it can be seen on figure 6.2 that the diode current is given by:

$$I_D = I_L \cdot D' = 0,8826 \cdot (1 - 0.595) = 0.357\ A$$

Therefore, the conducting loss will be

$$P_{conduct} = I_D \cdot V_D = 0.357\ A \cdot 0.3\ V = 107 mW \qquad (6.15)$$

The diode used in the MPPTC is of type MBR1100 and it is a Schottky diode. Therefore, there is no loss from reverse recovery charge. The power loss due to the equivalent capacitor in the diode is given by equation E.3 on page 192

$$P_{eq-cap} = \frac{1}{2} \cdot C_{eq} \cdot V^2 \cdot f_s\ [W]$$

The diode has an equivalent capacitor of $C_{eq} = 1200\, pF$ at a reverse voltage of $V = 7.2\ V$ [1N5823, 2001]. With a switching frequency of $f_s = 50\ kHz$ this gives a power loss of

$$P_{eq-cap} = \frac{1}{2} \cdot 1000\, pF \cdot (7.2V)^2 \cdot 50\ kHz = 1.3\ mW \qquad (6.16)$$

**Power loss in the MOSFET**

The MOSFET used in the MPPT is of type HUF76145P3. The power loss in a MOSFET is calculated by E.6 on page 193.

$$P_{gate} = Q_g \cdot V \cdot f_s\ [W]$$

The MOSFET is supplied with $V = 7.2\ V$ and has a total gate charge $Q_g = 130\ nC$. With a switching frequency of $f_s = 50\ kHz$ that is a power loss of

$$P_{gate} = 130\, nC \cdot 7.2\ V \cdot 50\ kHz = 47\, mW \qquad (6.17)$$

**Inductive loading of MOSFET power loss**

The power lost due to inductive loading of the MOSFET is given in equation E.8 on page 193, where the energy losses $W_{on}$ and $W_{off}$ both are calculated by equation E.7 on page 193. Here the time interval is the rise time $t_r$ and the fall time $t_f$ for $W_{on}$ and $W_{off}$ respectively. The transistor has $t_r = 57\ ns$ and $t_f = 38\ ns$ and the gate driver has $t_r = 31\ ns$ and $t_f = 32\ ns$. The gate driver can not boost the transistors $t_r$ and $t_f$ and therefore the times for the transistor is used and this gives a loss of

$$\begin{aligned} P_{ind} &= (W_{on} + W_{off}) \cdot f_s = \frac{1}{2} \cdot V \cdot I_L \cdot (t_r + t_f) \cdot f_s \\ P_{ind} &= \frac{1}{2} \cdot 7.2\ V \cdot 0.8826\ A \cdot (57\ ns + 38\ ns) \cdot 50\ kHz = 15.1\ mW \end{aligned} \tag{6.18}$$

## 6.5 Inductor Design

For the inductor design a ferrite core of type RM5/I made of ferrite 3F3 will be used. This material was selected because it satisfies the equations F.18 and F.19 on page 199. The following calculations are all made from the equations in appendix F on page 195.

The core is saturated at a magnetic flux of 450 mT [PHILIPS, 2000]. Therefore, the maximum magnetic flux in the inductor is set to $B_{max} = 300mT$. This gives an air gap in the core of

$$l_g = \frac{\mu_0 L I_{max}^2}{B_{max}^2 A_c} = \frac{4\pi \cdot 10^{-7} \frac{H}{m} \cdot 229\mu H \cdot (0.8826\ A)^2}{(300\ mT)^2 \cdot 24.8 \cdot 10^{-6} m^2} = 100\mu m$$

The number of windings can be determined with F.21 on page 200:

$$n = \frac{L I_{max}}{B_{max} A_c} = \frac{229\ \mu H \cdot 0.8826\ A}{300\ mT \cdot 24.8 \cdot 10^{-6}\ m^2} \approx 27 \tag{6.19}$$

This gives 27 windings in the inductor. The wire size $A_W$ is calculated with equation F.23 on page 200

$$K_U = \frac{n \cdot A_W}{W_A} \tag{6.20}$$

Where $K_U$ is the fill factor of the core, which in this is estimated to 0.5, see appendix F.4 on page 199. $W_A$ is the winding area on the core, which according to the data sheet for the core is $9.8\ mm^2 = 9.8 \cdot 10^{-6}\ m^2$, and $n$ is the number of windings. If these quantities is inserted into equation 6.20 it yields

$$A_W = \frac{K_U \cdot W_A}{n} = \frac{0.5 \cdot 9.8 \cdot 10^{-6}\ m^2}{27} = 1.81 \cdot 10^{-7}\ m^2 \tag{6.21}$$

With the wire size we are able to calculate the radius of the wire:

$$\begin{aligned} A_W &= \pi r^2 \Leftrightarrow \\ r &= \sqrt{\frac{A_W}{\pi}} = \sqrt{\frac{1.81 \cdot 10^{-7}\ m^2}{\pi}} = 240\ \mu m \end{aligned}$$

### 6.5.1 Inductor loss

Following the loss in the inductor will be calculated. This is respectively the loss in the wire and the loss in the core.

**Wire loss**

The resistance in the inductor is calculated so the power lost in the inductor can be determined. The penetration depth of the current in the wire is calculated:

$$\delta = \sqrt{\frac{\rho}{\pi \mu f}} = \sqrt{\frac{1.7 \cdot 10^{-8}\ \Omega \cdot m}{\pi \cdot 4\pi \cdot 10^{-7} \frac{H}{m} \cdot 50\ kHz}} = 293.5\ \mu m \tag{6.22}$$

Here it is used that $\rho = 1.7 \cdot 10^{-8}\ \Omega \cdot m$ and $\mu = 4\pi \cdot 10^{-7} \frac{H}{m}$ for copper. For the core used the inductor core has $MLT = 24.9\ mm$. It can be seen from equation 6.22 and 6.22 that the penetration detph is bigger that the radius of the wire. Because the penetration depth is bigger than the radius of the wire, the whole wire is conducting current and therefore the resistance in the wire can be determined as a DC-resistance according to equation F.5 on page 196.

$$\begin{aligned} R &= \rho\frac{l_w}{A_w} = \rho\frac{n \cdot MLT}{A_w} \Leftrightarrow \\ R &= 1.7 \cdot 10^{-8}\Omega \cdot m \cdot \frac{27 \cdot 24.9 \cdot 10^{-3}\, m}{1.63 \cdot 10^{-7}\, m^2} = 0.070\,\Omega \end{aligned}$$

The power loss in the wire is then

$$P_{wire} = R \cdot I^2 = 0.070\,\Omega \cdot (0.8826A)^2 = 54.5\,mW \qquad (6.23)$$

There is no loss from skin effect and proximity effect because the wire conducts a DC-current.

**Core loss**

In the data sheet a graph illustrates the relationship between core loss, frequency and $B_{max}$. From this graph the constant $K_f$ from equation in appendix F.16 on page 198 which describes the core loss, can be approximated to the following ($\beta$ has been chosen to the value 2.7, which is an average value [Erickson, 1999] page 474):

$$P_{core} = (9 \cdot 10^{-4} f^2 + 102.6f - 85798) \cdot B_{max}^{2.7} A_c l_m\ [W] \qquad (6.24)$$

$A_c l_m$ equals the volume of the core. This volume is $V_c = 574mm^3 = 574 \cdot 10^{-9} m^3$ and $B_{max}$ was set to 300 mT. With these values inserted into equation 6.24 the core loss is calculated:

$$P_{core} = (9 \cdot 10^{-4} \cdot (50\,kHz)^2 + 102.6 \cdot 50\,kHz - 85798) \cdot (0.300\,T)^{2.7} \cdot 574 \cdot 10^{-9}\,m^3 = 162\,mW \quad (6.25)$$

## 6.6 Efficiency

The total power lost in the converter is the sum of the losses calculated in 6.15, 6.16, 6.17, 6.18, 6.23 and 6.25:

$$P_{loss} = 107\,mW + 1.3\,mW + 47\,mW + 15.1\,mW + 54.5\,mW + 162\,mW = 387.5\,mW$$

With this loss calculated the efficiency of the converter can be calculated:

$$\eta = \frac{P_{out}}{P_{in}} = \frac{P_{in} - P_{loss}}{P_{in}} = \frac{3.0\,W - 0.3875\,W}{3.0\,W} = 87.1\% \qquad (6.26)$$

We see that the efficiency of the converter does not match the required one specified in section 6.1.

## 6.7 Simulation

It is important to see the results of the design before beginning implementation. In this section a simulation for the MPPTC will be carried out with the Simulink [2] program.

### 6.7.1 Simulation Model

To be able to carry out the simulation, a model of the MPPTC is made in Simulink. Figure 6.9 illustrates the simulation model. The component values that has been found in the previously sections are used. In the model there is one current measurement, and one voltage measurement. The current that is measured is the current that is running in the inductor, and the voltage measured is the output from the converter. This measurement node is connected to a scope that illustrates the results. The simulation is made for low duty cycle and high duty cycle. Figure 6.9 shows a diagram for high duty cycle. When a simulation for the low duty cycle is carried out the same model is used, except the voltage from the solar array is defined to 4.2 V, the resistance ($R$) to 12 $\Omega$ and the duty cycle is set to 0.3.

### 6.7.2 Results

The result from the simulation can be seen in figure 6.10 and 6.11 . In the figures there are two graphs. The upper graph illustrate the current and the lower graph illustrate the output voltage. In the current graph the Y-axis represents the current $(I)$ and the X-axis represent the time in $ms$. In the voltage graph the Y-axis represent the voltage $(V)$ and the X-axis represents the time as in the current graph.

[2] Simulink is a simulation program in Matlab

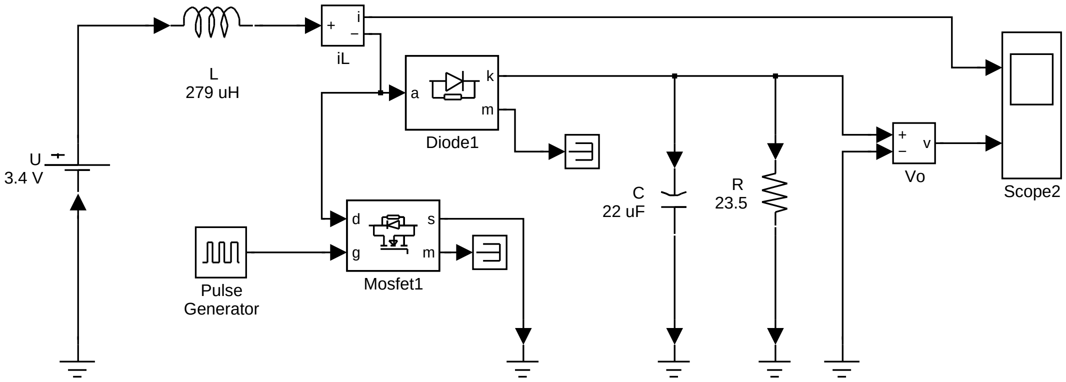


**Figure 6.9:** Simulation for the MPPTC in high duty cycle

### Results for high Duty Cycle

The results from the simulation in high duty cycle is shown in figure 6.10. As seen in the current graph the current is approximately 0.7 A. On the voltage graph the voltage is approximately 8.0 V. It can also be seen that the converter goes in discontinuous conduction mode in the start up because of the over-shoot.

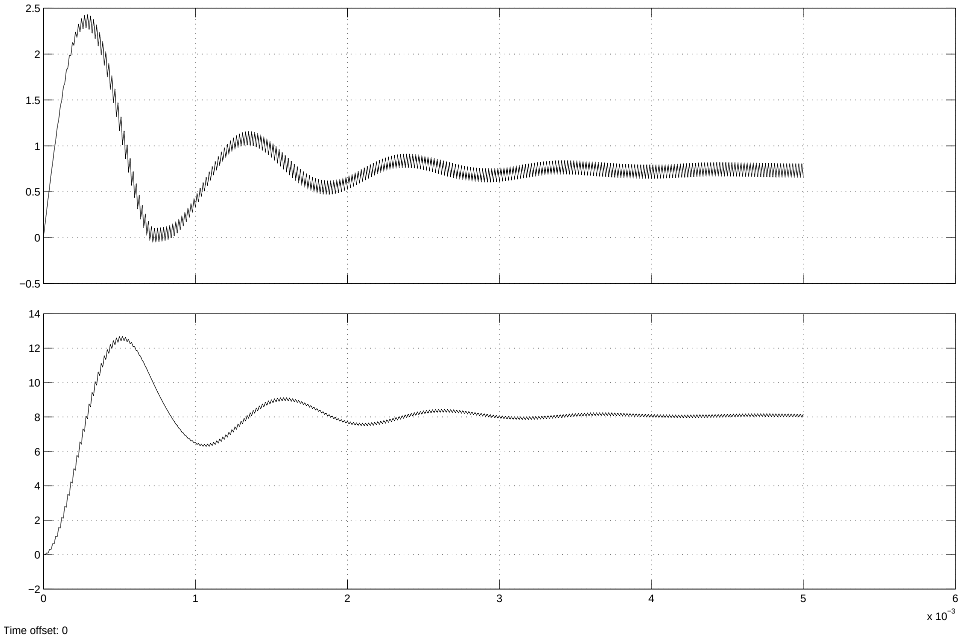


**Figure 6.10:** The results from the simulation for high duty cycle

### Results for low Duty Cycle

As seen in figure 6.11 where simulation for the low duty cycle is carried out the current is about 0.68 A and the voltage level 5.7 V.

### Common Results

Regarding the results above, it can been seen that the requirements about the voltage level and the ripple that was describe in 6.1.1 on page 41 has not been fulfilled in the simulation with low dutycycle, since the voltage on the output of the MPPTC is 5.7 V, and the requirement was 6.0 V. In high dutycycle the output voltage is 8.0 V where the calculated value was 8.4.

## 6.8 Implementation of the MPPTC

This section describes the implementation of the MPPTC. Figure 6.12 shows a complete circuit diagram of the MPPTC.

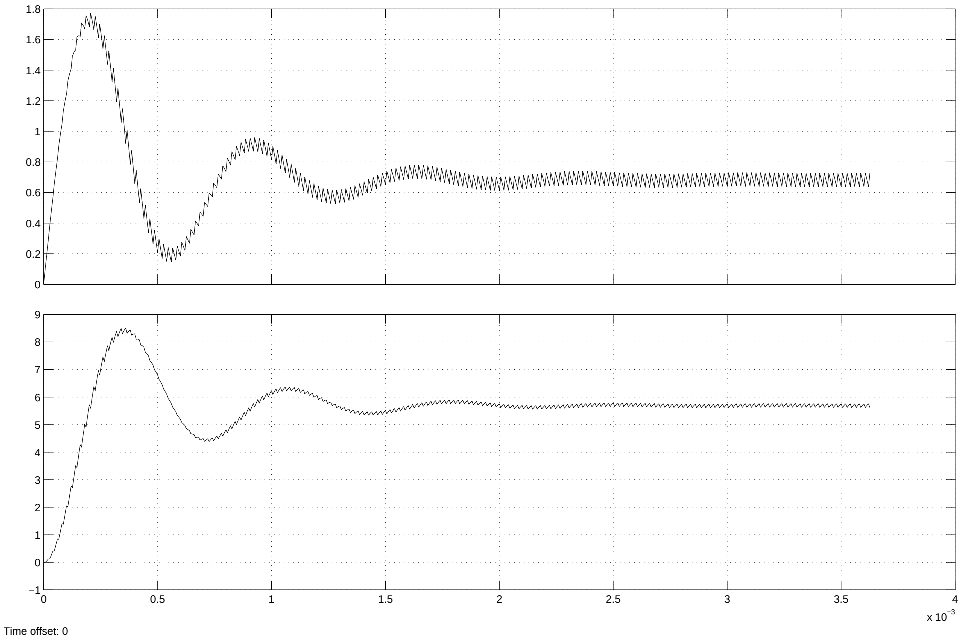


**Figure 6.11:** The results from the simulation for low duty cycle

**Inductor Realization**

The core used is of type RM5 with an air gap of 100 $\mu m$. For the implementation of the inductor a triple parallel wire configuration, with a wire diameter of $250$ $\mu m$ is used. This is done to lessen the inductor resistance at 50 kHz. The number of windings is kept at 27. At a frequency of 50 kHz the inductance of the inductor was measured to be $228$ $\mu H$ and the resistance in the wire was measured to $0.3\Omega$. This resistance deviates a lot from the calculated resistance i equation 6.23. Since the measured resistance is an AC-resistance an the one in equation 6.23 is a DC-resistance it is concluded that there must be skin effect in the inductor. This violates the calculations made on the power loss in the inductor made earlier.

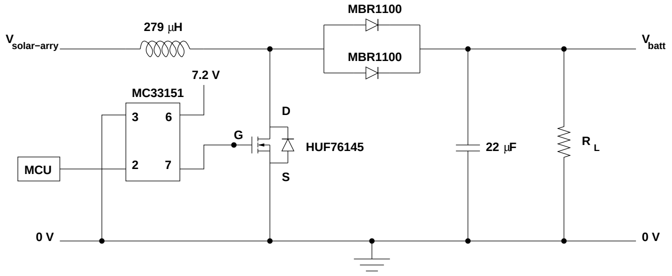


**Figure 6.12:** The complete circuit diagram of the MPPTC

## 6.9 Test Results and Evaluation

In this section the results from the module-test of the MPPTC will be described. The test was made according to a test specification that can been seen in appendix J.1 on page 211. In the test the following three functions were tested:

1. The output voltage from the converter
2. The output ripple
3. The converters efficiency

The test was performed without using a gate driver because the driver was not available at the time where the test were taking place. Either solar arrays and battery pack was used in the test. Instead of solar arrays a power supply was used and variable resistor was used instead load.

### 6.9.1 Output Voltage

To test the output voltage the equipment was set to the values that are listed in table 6.1. For the high duty cycle the input voltage is 3.4 V and the output is supposed to be 8.4 V with duty cycle of 59.5 % and a resistor of 23.5 Ω. The test was made with 2 W input power. During the test the duty cycle had to be set up to 68 % and the resistor to 4.2 Ω to be able to reach the 8.4 V on the output.

The test for the low duty cycle was performed as for the high duty cycle. The input voltage was set to 4.2 V, the duty cycle to 30 % and the resistor to 12 Ω. The duty cycle and the resistor was then adjust until the output voltage was 6.0 V. When the output voltage was 6.0 the duty cycle was 43.4 % and the resistor 21.9 Ω.

The duty cycles and the resistors in the test were higher than they were found in section 6.3.1 because of losses in components.

### 6.9.2 Output Ripple

To find the highest ripple on the output voltage the converter was set to convert from the lowest possible input voltage to highest possible output voltage with maximum input power. That is input voltage of 3.4 V to output voltage of 8.4 V with 3 W. The results from the test can be seen in figure 6.13.

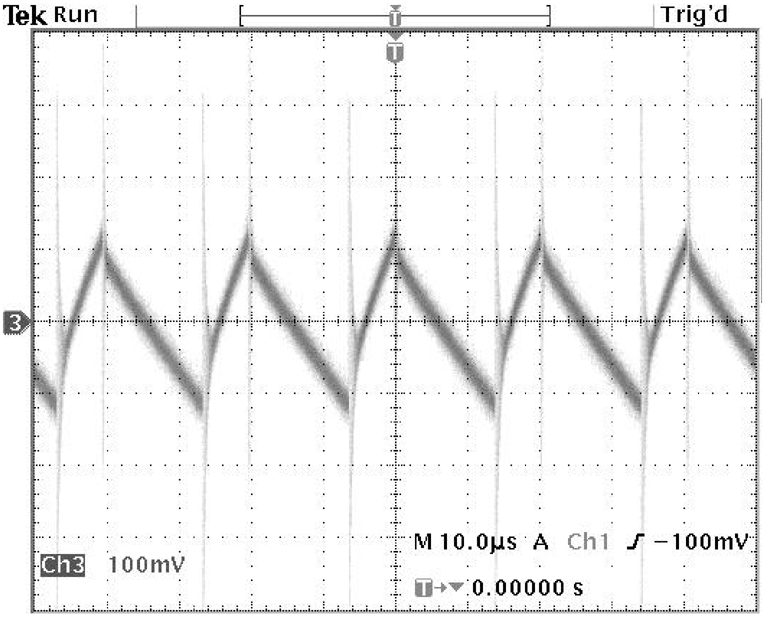


**Figure 6.13:** Ripple on the output voltage of the MPPTC

In the requirements for the converter in section 6.1.1 the ripple was define to be maximum $2\%_{p-p}$ that is 168 mV of 8.4 V . As can been seen in figure 6.1.1 the ripple is approximately 240 mV, what is considerably more than the requirements.

### 6.9.3 Efficiency

The test for efficiency was performed with the converter in three different modes (high-, low- and average duty cycle) with three different input power, which is because the components are not ideal.

#### High Duty Cycle

For the high duty cycle the converter boost the voltage from 3.4 V to 8.4 V. The results can be seen in table 6.2.

#### Average Duty Cycle

For the average duty cycle the converter boost the voltage from 3.8 V to 7.2 V. The result from the test can be seen 6.3.

| Input Power [W] | Efficiency [%] |
|---|---|
| 1.5 | 85.3 |
| 2 | 82.5 |
| 3 | 77.1 |

**Table 6.2:** Results with 3.4 V input voltage and 8.4 V output

| Input Power [W] | Efficiency [%] |
|---|---|
| 1.5 | 86.4 |
| 2 | 85.7 |
| 3 | 79.5 |

**Table 6.3:** Results with 3.8 V input voltage and 7.2 V output

**Low Duty Cycle**

For the low duty cycle the converter boost the voltage from 4.2 V to 6 V. The results from the test can bee seen in 6.4

| Input Power [W] | Efficiency [%] |
|---|---|
| 1.5 | 88 |
| 2 | 85.2 |
| 3 | 82 |

**Table 6.4:** Results with 4.2 V input voltage and 6 V output

The worst efficiency is when the converter is boosting from the lowest input voltage to the highest output voltage. As seen in table 6.2 where the input power is 3 W the efficiency is 77 %. Because of this loss in the circuit a measurement for each component was carried out. It was found out that most power loss was in the inductor. The best efficiency for the converter is 88 % and the worst is 77 % as mentioned above, which is not good enough to fulfill the requirements for efficiency because of loss in the circuit.

### 6.9.4 Test Evaluation

From the results of the tests that is described above it can been seen that the converter has fulfilled the requirements for the output voltage. The requirements for efficiency and ripple on the output has not been fulfilled.

# Chapter 7
# Power Conditioning Converter and Output Filtering

## 7.1 Overview

This section will provide an overview of the Power Conditioning Converter (PCC) and provide the specific design requirements that applies to this converter. Figure 7.1 shows the environment of the PCC. The following will briefly discuss the conditions that exists on each side of the converter.

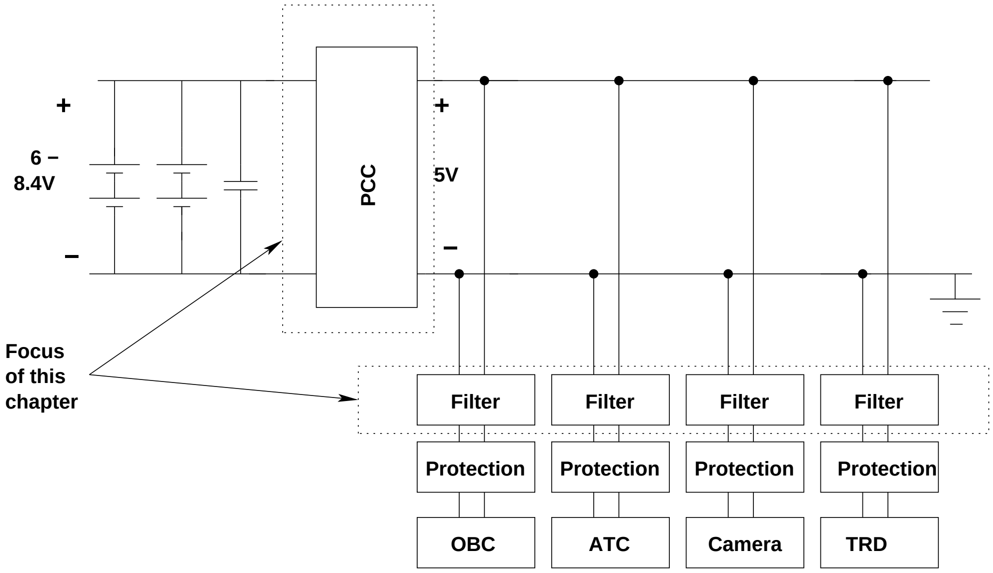


**Figure 7.1:** An overview of the PCC

### Input Characteristics

The converter transfers power from the intermediate power-bus to the main power-bus. On the intermediate power-bus the voltage-level varies slowly between 6.0-8.4 V due to changes in the battery charge condition. Because of the design requirements on the MPPT-converter it is known that the switching ripple on the intermediate bus is less than $2\%_{p-p}$ (see section 6.1 on page 41).

The DC-impedance seen by the PCC is difficult to estimate because it varies with the condition of the battery and the output-impedance of the MPPT converter. The AC-impedance however is very small due to the fact that the capacitor that is in parallel with the batteries decouples the DC-impedance.

### Output Characteristics

The output side of the converter is the main 5 V power-bus that through filters and protection circuits is distributed to the loads on the satellite. The impedance on this bus depends on which loads are turned on or off and on the current that is consumed by each load. It is possible to calculate the minimum resistive impedance of this bus since it is known that the maximum power consumption on the bus is 13 W (see chapter 3 on page 23). This information yields a minimum resistive impedance of:

$$R_{min} = \frac{U^2}{P} = \frac{(5V)^2}{13W} = 1.9\Omega \tag{7.1}$$

Further, it can be seen from formula 7.1 that the resistance goes to infinity as power consumption goes to zero.

When the satellite operates in idle mode the current consumption of the PSU users will be about 200mA (see chapter 3 on page 23), adding an expected 50mA for the consumption for the PSU itself a total of about 250 mA is used when the satellite operates in idle mode as it does most of the time. This yields an equivalent resistance in idle mode of:

$$R_{max} = \frac{5V}{250mA} = 20.0\Omega \tag{7.2}$$

### 7.1.1 Specific Requirements

In the following when referring to ”output” it is the output-wires to each connected load that are meant. On basis of the ”System Requirements Specification” (chapter 3 on page 23) the following requirements can be derived:

1. The average output voltage-level must be 5 V. Derived from subsection 3.3.1 on page 26
2. The output ripple must be less than $2\%_{p-p}$, but may have $4\%_{p-p}$ deviations for durations of 20 ms when loads are turned on or off. Derived from subsection 3.3.1 on page 26

In order to fulfill the overall efficiency requirement of the PSU as stated in the ”System Requirements Specification” (see subsection 3.4.1 on page 28) the following requirement on efficiency is set for the PCC converter:

3. The efficiency of the PCC should be 90% or better

Finally all environmental requirements which are stated in section 3.5 on page 29 of the ”System Requirements Specification” applies to the PCC circuitry as well.

## 7.2 Design of the PCC

In the following the PCC will be designed. Apart from the previous chapter which focused on circuit analysis to choose component values for the MPPT converter this chapter will focus on loss-modeling in order to find a trade-off between switching frequency, inductor size and capacitor size to produce an efficient converter.

### 7.2.1 Design Solution

To fulfill the requirements specified above, especially the step-down criterion, the obvious choice of converter type is the buck converter, but also other types can be used; A buck-boost or a Cuk converter are both capable of being used as step-down or step-up converters. However the buck-converter generally exhibits superior switch-utilization compared to the two fore mentioned topologies [Erickson, 1999] page 177.

### 7.2.2 Operation Overview

This subsection briefly discusses how the buck converter operates. The circuit diagram of the Buck-converter is depicted in figure 7.2. It basically consists of a switch and an output filter.

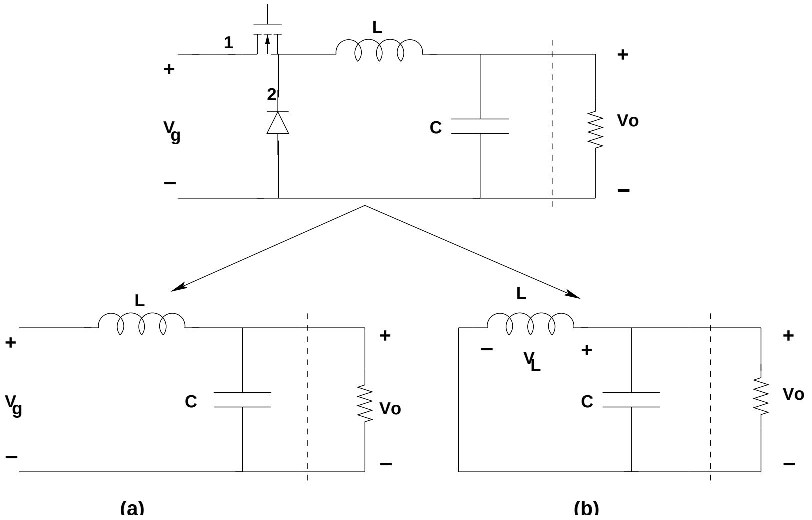


**Figure 7.2:** The operation of the Buck converter (seen uppermost) with: (a) switch on and (b) switch off

When the switch is in position 1 (as shown in figure 7.2 (a)) a current flows from the input side to the inductor. Since the input voltage is higher than the output voltage the inductor current rises and the capacitor charges while the output-voltage grows.

When the switch is shifted to position 2 (as shown in figure 7.2 (b)) the current continues to flow through the inductor at a declining rate and the capacitor voltage decreases because it is being discharged by the power drawn from the load.

In effect the load will receive power with a DC-voltage proportional to the duty-cycle of the switch and an imposed AC-voltage that is dependent on the power required by the load and the size of inductor and capacitor.

**Design equations**

When designing the buck converter three main design equations are needed: The voltage transfer function, the voltage ripple function and the corner frequency as function of the output-filter. The following formulas is cited from [Ned Mohan, 1989] page 172. The voltage transfer function for the buck converter shows a linear dependency on the duty-cycle:

$$\frac{V_o}{V_g} = D \tag{7.3}$$

where:
$V_g$ is the input voltage [V]
$V_o$ is the output voltage [V]
$D$ is the duty-cycle

The ripple on the output voltage can be described with the following equation:

$$\frac{\Delta V_o}{V_o} = \frac{\pi^2(1-D)}{2}\frac{f_c}{f_s} \tag{7.4}$$

where:
$\Delta V_0$ is the ripple on output voltage
$f_c$ is the corner frequency of the output filter [Hz]
$f_s$ is the switching frequency [Hz]

The corner frequency of the output filter can be calculated like this:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \quad [Hz] \tag{7.5}$$

where:
$L$ is the inductance of the inductor [H]
$C$ is the capacitance of the capacitor [F]

As described in chapter 6 on page 41 a converter can operate in two different modes: CCM and DCM. When operating in DCM the conversion ratio of the converter becomes load dependent and therefore in order to make converter control simple and to minimize losses, it is desirable to operate in CCM, which is done by keeping the ripple in the inductor current smaller than the lowest nominal load-average-current. For the buck converter the critical load-current can be described with equation 7.6 [Knopf, 1999] page 32.

$$I_{critical} = \frac{(1-D)\cdot V_o}{2\cdot L\cdot f} \quad [A] \tag{7.6}$$

When the DC-current drawn from the converter decreases below this critical current the converter enters DCM operation.

### 7.2.3 Switching and Switching Losses

As seen on figure 7.2 the buck converter needs to implement a Single-Pole-Double-Throw switch (SPDT). Two different switch implementations have been considered; A switch consisting of a MOSFET transistor and a freewheeling-diode or a switch consisting of two MOSFET transistors which are controlled such that each transistor conducts in one part of the duty-cycle.

The double MOSFET switch is more efficient because it can be implemented with very low conduction loss compared to the diode-switch. But if one of the transistors are subject to a SEU or SEL, this may lead to a short

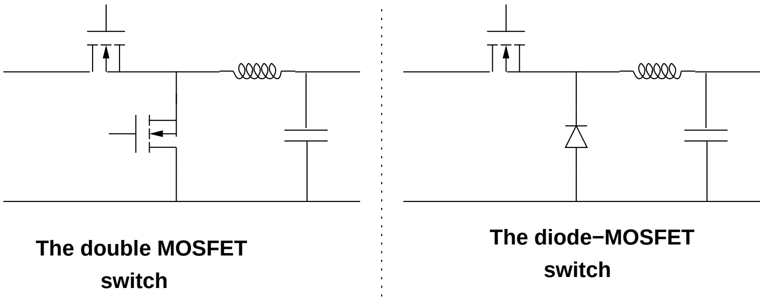


**Figure 7.3:** SPDT switch Realization in the PCC- to the left with double MOSFET realization and to the right with diode-MOSFET realization

circuit to ground. Secondly the double MOSFET-switch is not suited if the converter operates in DCM, because the reverse current that would have been blocked by the freewheeling-diode in a freewheeling-diode switch is free to flow to ground and it is therefore effectively lost.

Because the converter must be able to operate in the discontinuous conduction mode since the protection circuits may turn any load off (because of high current consumption) without any supervisory control that could initiate actions to prevent the converter to enter DCM, it has been chosen to implement the switch with the MOSFET-transistor and a freewheeling-diode as can be seen on figure 7.3.

### The MOSFET and Diode switch

When the transistor is on (i.e. it conducts) a current will flow from the generator side of the buck converter to the filter side and the diode will be reverse biased and thus not conducting. When the transistor is turned off the inductor will continue to conduct current and the diode will therefore become forward biased and conduct.

As MOSFET transistor the HUF76145P3 (datasheet [FAIRCHILD, 2000]) from Fairchild has been chosen, because of its low on-resistance ($R_{on} = 4.5\ m\Omega$ at 25°C (increasing with temperature)) and because of low rise and fall times ($V_{gs} = 10\ V, t_{rise} = 57\ ns, t_{fall} = 38\ ns$). The diode is a Motorola MBR1100 Schottky diode (datasheet [MBR1100, 1999]). A Schottky diode has been chosen to minimize switching losses due to reverse recovery charge[1] and because the lower forward voltage drop of a Schottky device compared to a PN-diode incurs less conduction loss. The open loop forward voltage drop of the MBR1100 at 25 °C is $\simeq 0.3\ V$ at 0.3A and the voltage-drop decreases with temperature.

### Gate-driver Circuitry

In order to switch the MOSFET on and off, a gate voltage must be applied. The threshold voltage of the MOSFET is between 1 to 3 V meaning that a voltage that is at least 3 V higher than the maximum source voltage must be applied in order to operate the MOSFET. When the MOSFET is turned on the source-voltage will be equal to the battery voltage (about 6.0 V - 8.4 V). It is therefore not possible to operate the MOSFET using any voltages that is already found in the PSU.

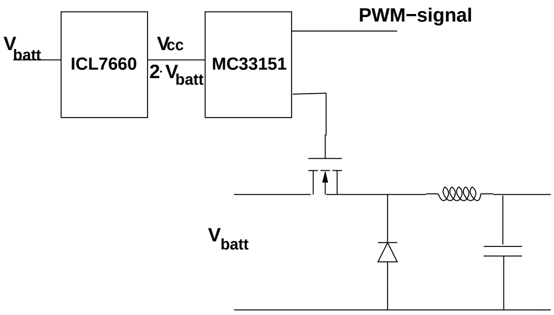


**Figure 7.4:** The PCC with voltage doubler and gate driver applied

Therefore, a gate-driver is needed to operate the MOSFET device. Secondly because a gate-driver is optimized to operate a MOSFET it is able to switch the transistor on or off faster than if the device was directly controlled by for example a micro-controller.

[1] Explained in appendix E on page 191

One type of gate-driver stores energy in either a capacitor or an inductor and switches this energy to a higher potential when it is told to charge the gate of the MOSFET. It has, however, not been possible to find an available gate-driver of this type which both are specified to work from less than 10 V and at the same time withstand the industrial temperature range.

Therefore, it has been chosen to use a voltage doubler device ICL7660 from Intersil (Datasheet: [ICL7660, 1999]) to double the battery voltage. This means that a voltage of between 12.0-16.8 V is available for switching the transistor. The gate is then driven by a low-side gate-driver, which does not switch a voltage to a higher potential. The gate-driver is a MC33151 (Datasheet: [MC33151, 1999]), which exhibits typically rise and fall times of 35 ns for rise and 36 ns for fall. At figure the PCC with voltage doubler and gate-driver are shown.

**Switching Losses**

An equivalent switching diagram of the switch is depicted in figure 7.5.

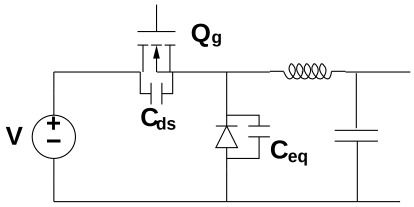


**Figure 7.5:** Model of switching losses

The general theory of the different losses associated with switching of the diode and MOSFET is described in appendix E on page 191 and will not be described here, but the results will be used to calculate the losses in the switch of the converter. The following switch-losses are associated with operation off the SPDT switch described here:

- Transistor turn-on and turn-off loss ($W_{turn\_on}, W_{turn\_off}$)
- Losses due to gate charging ($W_g$)
- Losses due to discharging of the capacitors that are in parallel with the voltage source (i.e. $C_{ds}$ and $C_{eq}$) ($W_{ds}, W_{eq}$)

The total loss inflicted in each switch-cycle (i.e. one turn-on and one turn-off) is:

$$W_{total} = W_{turn\_on} + W_{turn\_off} + W_g + W_{ds} + W_{eq} \quad [J] \tag{7.7}$$

Using the formulas derived in appendix E on page 191 this can be expanded to:

$$W_{total} = \frac{1}{2} \cdot V \cdot I_L \cdot (t_{rise} + t_{fall}) + Q_g \cdot V_g + \frac{1}{2} \cdot V^2 \cdot (C_{ds} + C_{eq}) \quad [J] \tag{7.8}$$

Where:
$V$ : is the applied drain voltage [V]
$V_g$: is the applied gate voltage [V]
$I_L$ : is the average inductor current [A]
$t_{rise/fall}$ : are the rise and fall times for the MOSFET [s]
$Q_g$ : is the charge required to operate the gate [Q]
$C_{ds}$ : is the MOSFET drain-to-source capacitance [F]
$C_{eq}$ : is the input capacitance of the Schottky diode [F]

Since $W_{total}$ is the energy lost in each switch-cycle the switching loss increases linearly with the switching frequency. The power loss because of switching is:

$$P_{switch-loss} = W_{total} \cdot f_{sw} \quad [W] \tag{7.9}$$

where:
$f_{sw}$ : is the switching frequency [Hz]

### 7.2.4 Conduction Losses

The second factor that leads to losses in the converter is conduction losses which are inflicted because of ohmic resistances that are in effect in the converter circuitry because of non idealities in switching and reactive components.

These losses will be analyzed in the following. In order to simplify this analysis it will be assumed that the converter always operate in the Continuous Conduction Mode (CCM). Further, the small-ripple-approximation is used, i.e. voltage- and current-ripples are considered insignificant compared to DC-values. Finally switching effects are not considered since the amount of time used for switching is considered to be small compared to the amount of time where the circuit is conducting normally.

Figure 7.6 depicts the Buck-converter circuit with simple models of the non-ideal ideal components consisting of resistors, voltage sources and the ideal components. Each component model will be analyzed in the following in order to find the loss that is associated with that component.

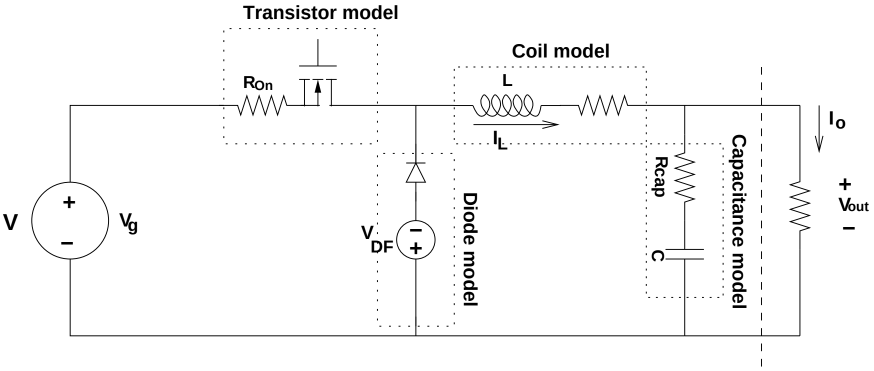


**Figure 7.6:** Circuit diagram with parasitic resistances

**Transistor Loss**

When the MOSFET is turned off ($V_{gs} = 0\ V$) no current flows through the device and hence no power is dissipated. As the MOSFET is turned on a current equal to the current in the load will flow from drain to source. Part of this current will lead to power dissipation because of internal resistance in the inducted channel between drain and source. This resistance is equivaleted with a resistor $R_{on}$ and in one switching interval it dissipates the power:

$$P_{transitor} = R_{on} \cdot I_L^2 \cdot D \quad [W] \tag{7.10}$$

Where:
$I_L$ : is the inductor current [A]

**Diode Loss**

While the diode is reverse biased it conducts no current except for a negliable leakage current. As the diode becomes forward biased when the MOSFET switches off it begins to conduct the load current and therefore begins to dissipate energy because of the voltage drop over the device.

The forward voltage drop of a diode depends unliniarly on the current through the device. In order to calculate the power dissipated one must therefore supply the specific forward voltage drop of the diode for the current that passes through the device when it is conducting. Because of the small-ripple-approximation this current corresponds to the load current. The power dissipation is therefore:

$$P_{diode} = V_f \cdot I_{load} \cdot (1 - D) \quad [W] \tag{7.11}$$

**Inductor Loss**

The windings and core which make up the inductor exhibits ohmic resistance and can be equivaleted with a resistor ($R_{wire}$). Appendix F on page 195, which treats design of inductors, explains how to calculate the equivalent resistance. The power dissipated in this resistance can be divided into two: A conductive loss and a core-loss. All of the load current passes through the inductor and the power dissipated as conductive loss is therefore:

$$P_{wire} = R_{wire} \cdot I_L^2 \quad [W] \tag{7.12}$$

The core-loss depends on the material selected for the core and is in equation F.16 on page 198 described as:

$$P_{core} = K_{fe} \cdot B_{max}^{\beta} \cdot A_c \cdot l_m \quad [W] \tag{7.13}$$

where:
$K_{fe}$ and $\beta$ is constants which can be determined for the selected material.

**Capacitor Loss**
The capacitor is not ideal and is modeled with a resistor ($R_{cap}$) in series. The main factor contributing to this resistance is the loss angle of the capacitor. The loss angle is the deviation of the phase-shift of the capacitor from 90 ° measured at 120 Hz and it gives the following equivalent resistance [AVX, 2001]:

$$R_{cap}(f) = \frac{\tan\delta}{2 \cdot \pi \cdot f \cdot C} \quad [\Omega] \tag{7.14}$$

where:
$\delta$ : is the loss angle [°]
$f$ : is frequency [Hz]
$C$ : is capacitance [F]

The current that passes through the capacitor equivalent resistor each switch-cycle is a fraction of the ripple-current of the inductor. This fraction is the current division between the series resistance of the capacitor and the load impedance. However this current passes the series resistor twice because the capacitor is both charged and discharged due to the ripple current. This yields the loss:

$$P_{capacitor} = R_{cap} \cdot 2 \cdot \left(\sqrt{2} \cdot \Delta I_{L_{peak}} \cdot \frac{R_{cap}}{R_{cap} + R_{load}}\right)^2 \quad [W] \tag{7.15}$$

where:
$\Delta I_{L_{peak}}$ : is the inductor ripple peak current [A]

It should be noted that this model does not account for losses due to current ripples at harmonic frequencies of the switching frequency. However since the energy-spectrum of the current ripple is tightly centered about the switching frequency then the model is deemed adequate.

**Summarized Conduction Loss**
Summing contributions up from the analyzed components a total conduction loss can be found as:

$$P_{conduction} = P_{on} + P_{diode} + P_{wire} + P_{core} + P_{capacitor} \quad [W] \tag{7.16}$$

### 7.2.5 Design Optimization Process

As mentioned in the introduction to this section loss-modeling will be used to select values of the inductance and switching frequency. This is done by implementing the inductor design method[2] and the loss model from the previous subsection in a matlab program that given a set of initial conditions makes a frequency sweep where inductance and losses are calculated and presented in graphs. It is then possible to choose the frequency and corresponding inductance that results in the lowest theoretical loss. The matlab-model can be found on the enclosed CD-ROM[3].

At figure 7.7 the flow chart of the design process are shown. First the static component parameters are set, i.e. the core, transistor, diode and capacitor. Then the static simulation parameters are specified, i.e. the input and output voltage, the maximum allowed ripple at output voltage and inductor current and the load current. The required duty-cycle is then calculated and the design process can begin.

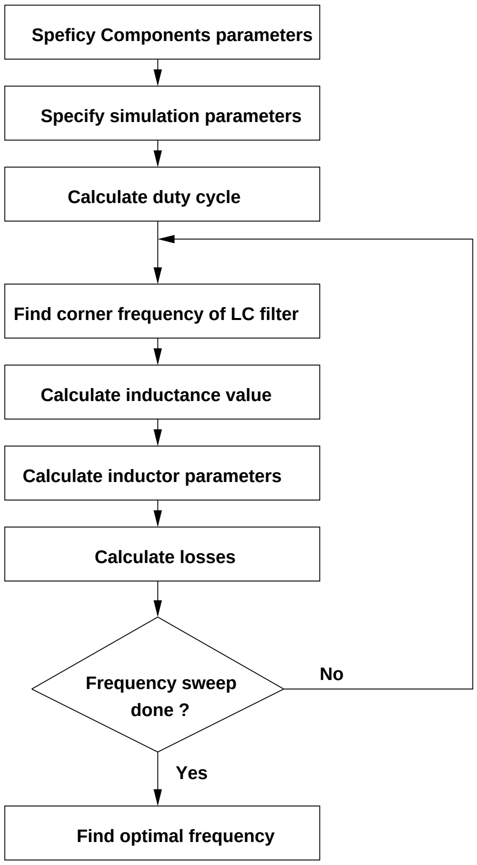

Speficy Components parameters
Specify simulation parameters
Calculate duty cycle
Find corner frequency of LC filter
Calculate inductance value
Calculate inductor parameters
Calculate losses
Frequency sweep done ?
No
Yes
Find optimal frequency

**Figure 7.7:** The design flowchart

The design loop does a frequency sweep from 4 kHz to 100 kHz and at each frequency it designs the inductor and calculates all losses. Finally the losses at all frequencies are compared and the optimal frequency is found.

This model has been used to simulate various choices of components in order to find a set of values that fulfill the following requirements:

1. As high efficiency as possible
2. An output voltage ripple of 1 $\%_{p-p}$.
3. An output current ripple of less than 200 mA which allows the converter to operate in CCM with nominal load

During one day the satellite has about 15 orbits and in 5 of them the satellite is active above Denmark in approximately 12 minutes (see chapter D on page 185). This means that each day the satellite consumes the following energy during the one hour of activity:

$$E_{active} = 2.5\ A \cdot 5\ V \cdot 1 \cdot 60^2\ s = 45\ kJ$$

And in the 23 hours when the satellite is idle it uses:

$$E_{idle} = 0.25\ A \cdot 5\ V \cdot 23 \cdot 60^2\ s = 103.5\ kJ$$

Therefore, the load current which has been selected for the optimization process is the current that corresponds to the idle consumption of the satellite.

The requirement that the output ripple must be less than 1 $\%_{p-p}$ which is less than the requirement of the converter (2 $\%_{p-p}$) is due to the fact that the ripple will be larger when the load current is larger than what is used for the optimization process.

[2] as presented in appendix F on page 195

[3] path

The following paragraphs describe the components that were finally chosen, whereafter the results of the design process with these components are given.

### Component Selection

Because the PSU must be designed to operate in the space environment it is not possible to use capacitors with liquid electrolytes because of out-gassing effects. Therefore, in order to get capacitors with a capacitance higher than about 1 $\mu F$ either tantalum or solid aluminum capacitors must be used[4]. Therefore it has been chosen to use solid aluminum since they generally exhibit smaller loss-angle and therefore less equivalent series resistance. The converter was simulated with different sizes of capacitors and finally a capacitance of $68\ \mu F$ was chosen. A larger value would have resulted in a too large current ripple.

In order to achieve a small implementation of the inductor the core was selected from the criterion; small physical size and then tested with equation F.18 on page 199 to determine if they were capable of operating at maximum load current with out saturating and exhibiting high resistance. The core was selected using [PHILIPS, 2000] as a reference and three core-types were found: RM5, RM6 and RM8, all made of PC40 ferrite with RM8 the largest feasible for the Cubesat. Their data can be seen in table 7.1 (description of parameters in appendix F on page 195).

| | RM 5 | RM 6 | RM 8 |
|---|---|---|---|
| $B_{sat}$ | 450 mT | 450 mT | 450 mT |
| MLT | 24.9 mm | 31 mm | 42 mm |
| Ac | 25 mm$^2$ | 37 mm$^2$ | 63 mm$^2$ |
| Wa | 9.5 mm$^2$ | 16 mm$^2$ | 31 mm$^2$ |
| Size (height, width, length) | 10.4·6.8·14.9 mm | 12.4·8.2·17.9 mm | 16.4·11·23 mm |

**Table 7.1:** Core data for different cores

During the simulations it became clear that the RM8 core was the best suited for the task, because of the high current it must endure when operating at 2 Ω (2.5 A). It was the only core of the three that did not go into saturation when operating at maximum current and with a reasonable air gab. Therefore, this large core will be used for the converter.

After the number of windings and the cross section area of the wire have been determined respectively (using a fill factor of 0.6) the winding resistance and the maximum B-field can be calculated using equation F.11 and F.21 in appendix F on page 195:

$$B_{max} = \frac{LI_{max}}{nA_c} \quad [T] \tag{7.17}$$

$$R_{wire} = \frac{l_w \rho}{2\pi\delta r_w} = \frac{nMLT\rho}{2\pi\delta r_w} \quad [\Omega] \tag{7.18}$$

Using these two values the inductor conduction- and core-loss can be calculated.

The initial simulations showed that one of the major loss effects was the conduction loss in the diode. In order to minimize this and to make sure the diode(s) can handle the current, it was chosen to connect three diodes in parallel and thereby obtain a lower forward voltage drop at the given load current.

### Modeling Results

On figure 7.8; 6 different parameters as functions of switching frequency can be seen. Starting from the top left, the corner frequency of the LC filter can be seen - this value sets the value of the inductor and the capacitor as described in equation 7.5. The top middle figure describes the number of windings on the inductor and on the right is the inductance. The bottom left is the cross section of the wire which influence the winding resistance, seen on the right. In the middle at the bottom the maximum B-field can be seen - this is the parameter that determines the core-loss, together with the frequency (see equation 6.24).

Figure 7.9 show plots of the different loss-contributions and the total loss in the converter vs. frequency. It can be seen on the left graph that the switching loss (dashed line) rises linearly with the frequency. The inductance loss (doth-dashed line) is the major loss-factor in the converter. Finally it can be seen that the core-loss (lowest

[4] Because of limited space on the satellite it is not feasible to use capacitors in parallel

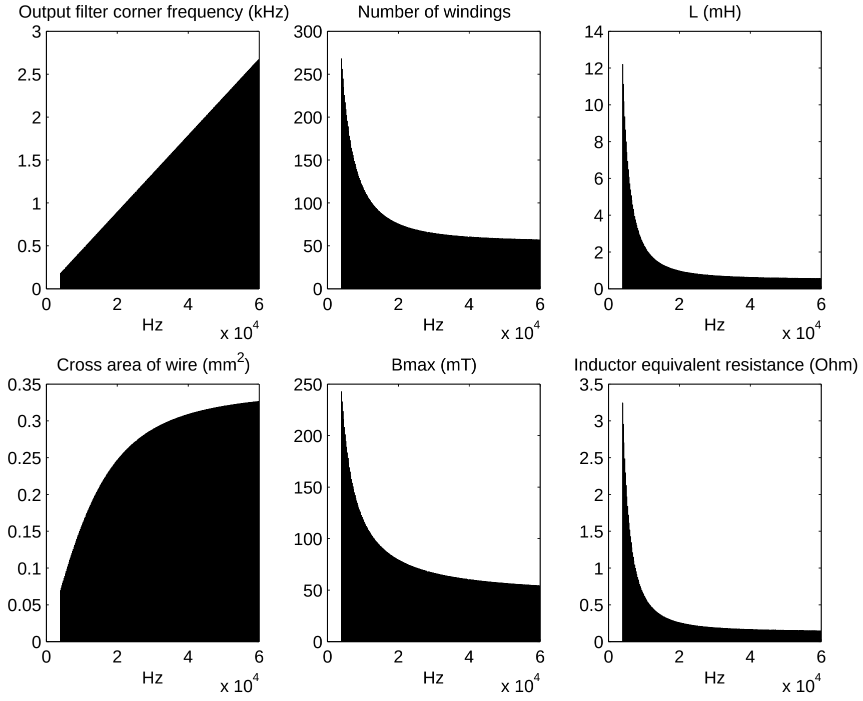


**Figure 7.8:** The results from the modeling - seen from the top left to the right it is the corner frequency of the LC filter, the number of windings on the inductor, the inductance, the cross area of the wire, the maximum B-field and the conduction resistance of the inductor. All values are plotted as a function of frequency measured in Hz.

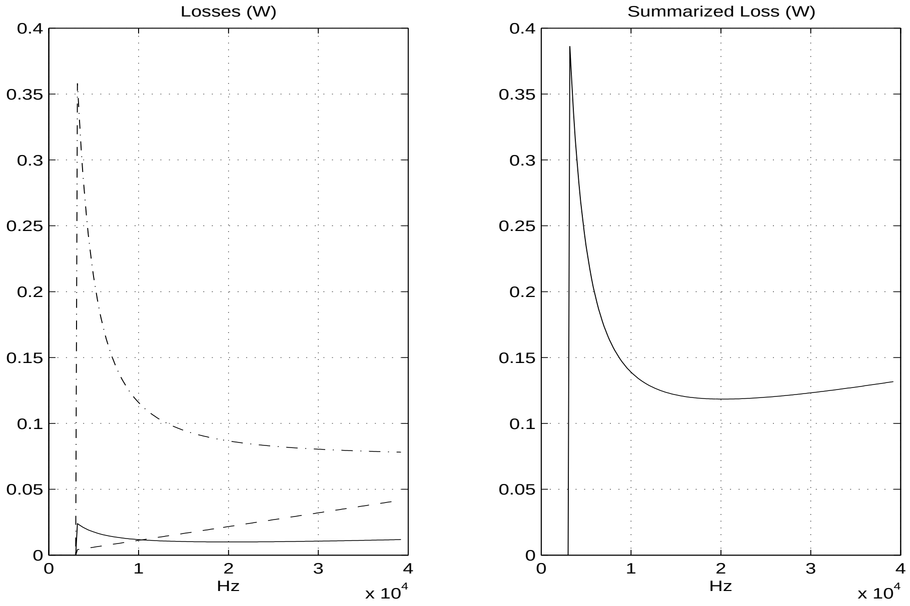


**Figure 7.9:** Different contribution to loss (left) and total loss (right). X-axis is switching frequency and Y-axis is power (W). At left dashed line is switching loss, solid line is core loss and dot-dashed line is conduction loss.

solid line) does not have a large effect on total loss. For low frequencies inductance loss is high because at these frequencies the needed inductance is large which results in a inductor with many windings and therefore a greater resistance.

On the right graph the summarized loss can be seen at it is evident that the optimal switching frequency, according to the model, is about 21 kHz. Along with the graphs the model outputs the parameters for the optimal switching frequency these are given table 7.2 as well as the predefined component values.

| | |
|---|---|
| Switching frequency | 21.0 kHz |
| Converter efficiency | 93.1 % |
| Inductance | 925 $\mu H$ |
| Inductor windings | 73 |
| Inductor air-gap | 550 $\mu m$ |
| $B_{max}$ at 2.5A | 437 mT |
| Diameter of wire | 0.566 mm |
| Inductor resistance | 0.24 Ω |
| Inductor ripple current | 89 mA |
| Capacitor capacitance | 68 $uF$ |

**Table 7.2:** Final results of the optimization process

## 7.3 Simulation

The simulation of the converter was done i Matlab using Simulink and the power toolbox. The Simulink model can be seen on figure 7.10. The purpose of the model is to explore how the converter performs with the previously found component values when the load current is raised to 2.5 A instead of the 0.3 A that the converter was optimized to handle with least possible loss. The simulation should show that the converter is able to operate at this load.

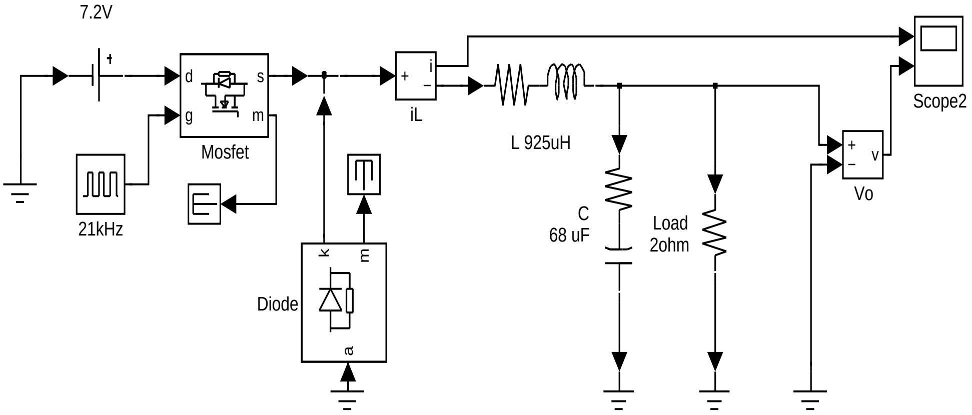


**Figure 7.10:** The Simulink model of the PCC

The model uses the values from the optimization modeling, using a fixed duty-cycle of 69.4% to give a theoretical 5 V output-voltage and thus the converter is unregulated. The simulink model was made from the results of the loss modeling done in the former chapter and therefore it includes resistance in the inductor and capacitor. At figure 7.11 the result of the simulation with $I_{load} = 2.5\ A$ is shown.

It can be seen that the system delivers about 4.6 V, with a very small ripple on the output voltage. The peak to peak ripple on the inductor current can be read to be approximately 100 mA which gives a 50mA ripple to each side of the DC value of the current. Since the expected current drain of the PSU itself is about 50mA, it is clear that the converter will operate in CCM under most circumstances.

## 7.4 Output Filtering

The role of the output filters that are connected to each of the loads are to attenuate any disturbances due to loads being switched on or off. The placement of these filter are shown on figure 7.1. These are needed because the main filter at the converter output is only designed to meet ripple specifications for steady-state operation.

Since the currents for all loads must be measured for house-keeping (see chapter 3 on page 23) a shunt resistor will come in series to each load in order to measure the current - this will be described in section 10.3 on page 111. Now together with a capacitor in parallel between supply and common this shunt resistor can be utilized to implement a first order passive RC-filter.

In order to make this filter as effective as possible this filter should have its corner-frequency as near 0Hz as

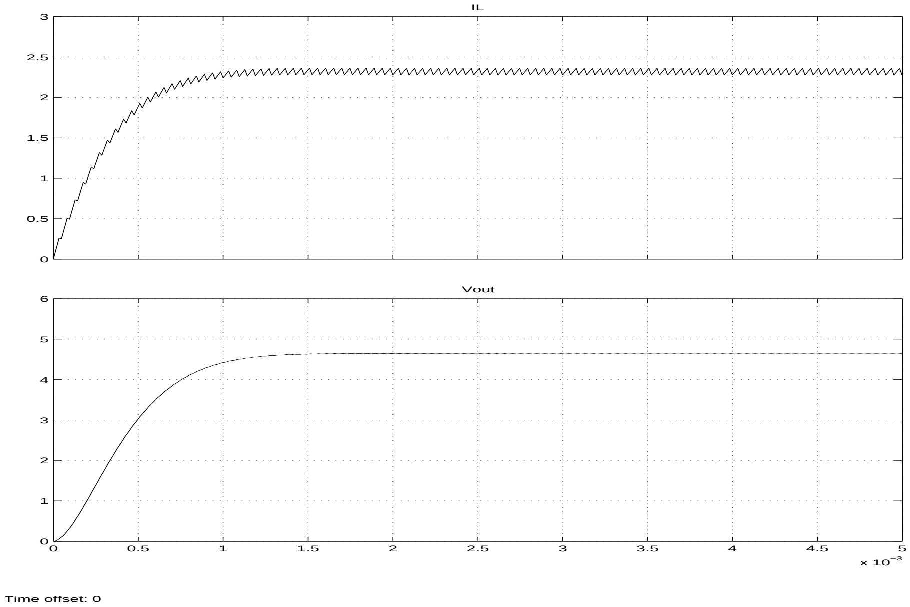


**Figure 7.11:** The results from the simulink simulation. At the top is the inductor current with current [A] at the y-axis and the bottom is the output voltage with voltage [V] at the y-axis. At the x-axis is time measured in seconds

possible since it must block everything else than the direct current. The corner frequency of a RC-filter is given by [Haugen, 1994] page 171:

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} \tag{7.19}$$

Where:
$R$ : is series resistance [$\omega$]
$C$ : is parallel capacitance [F]

The shunt resistor is chosen to 10 $m\Omega$ (see section 10.3 on page 111), it is this low to minimize the voltage-loss that develops across the resistor when it conducts. As can be seen from equation 7.19 it is desirable to have C as large as possible in order to have as low corner frequency as possible. However the largest realistic value in order to conserve circuit-space is about $47uF$. This gives a corner frequency of the low-pas filter of:

$$f_c = \frac{1}{2 \cdot \pi \cdot 10m\Omega \cdot 47\mu F} \simeq 339kHz \tag{7.20}$$

It would be preferable to have a much lower corner frequency, but because of the reasons stated above this is not possible. However because the shunt resistor is very small it may be feasible to expect that the actual series-resistance effecting the filtering is more than the resistor itself, because of converter output resistance and resistance in wires etc. Hence the corner frequency may be lower than previously calculated.

For frequencies below the switching frequency it must be expected that the controller of the converter will make sure to compensate for disturbances during switching.

## 7.5 Implementation of the PCC

This section will provide the complete circuit diagram of the Buck-converter with all components and values and it will describe some of the details regarding implementation of the converter. The circuit diagram is given on figure 7.12. The output-filter design is applied to each connected load (one load shown).

### Inductor realization

When implementing the inductor it became clear that the fill factor of 0.6 was too high for a wire with a diameter of 0.56mm at the RM8 core and trouble to fix enough windings on the coil-former arose. It was therefore decided to use two twisted wires with a diameter of 0.355 mm, using the same number of windings. The inductor resistance at 20kHz could then be measured to 0.31Ω which is only a bit higher than the calculated

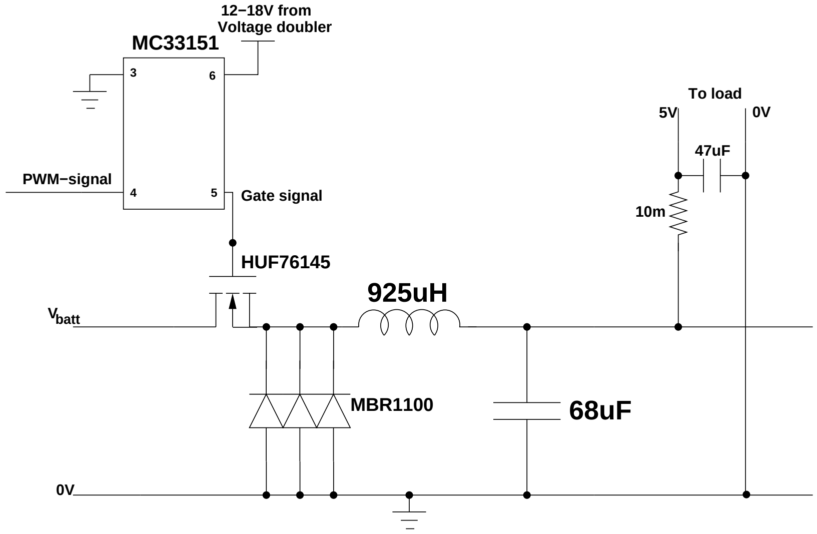


**Figure 7.12:** The complete circuit diagram of the PCC

0.24Ω. This configuration gave a inductance of about 990$\mu$H and the air gap was the raised to 630$\mu$m giving a inductance of about 920$\mu$H.

## 7.6 Test Results and Evaluation

The test was performed according to the testspecification in chapter J.2 on page 214 where three test cases are specified:

1. Output voltage vs. duty-cycle
2. Ripple in steady state
3. Efficiency

The tests were performed without the use of gate-drivers or voltage doublers, because these components was not available at the time when the test was performed - instead a frequency generator was used to drive the gate directly.

### Output voltage vs. Duty-cycle

In the first test the load resistance was set to 2Ω and the input voltage to 7.2 V and the duty-cycle was adjusted until an output of 5V was reached. The same was done with an input voltage of 6 V. The following duty-cycles were found:

**7.2V:** 84.6%

**6V:** 100%

The duty-cycles found in this test are higher than the theoretical values, because of the losses in the non-ideal components of the converter.

At 6 V input voltage the output voltage of 5 V was reached exactly by keeping the MOSFET open through the complete switch period. This is not a problem since as 2Ω is the worst case load the converter does not have to supply more current than what it supplied in this test-case.

### Ripple in Steady State

In the second test the load resistance was set to 2Ω and the input voltage to 7.2 V and the duty-cycle was adjusted until 5 V output voltage was reached. At figure 7.13 the ripple on the output voltage can be seen.

Neglecting the spikes it can be seen that the voltage ripple is below 50 mV which is in accordance with specifications. The spikes that can be seen when the gate signal changes state have amplitudes of maximally 250 mV which is more than the specified allowable ripple of 100 mV. The reason for these spikes are that each

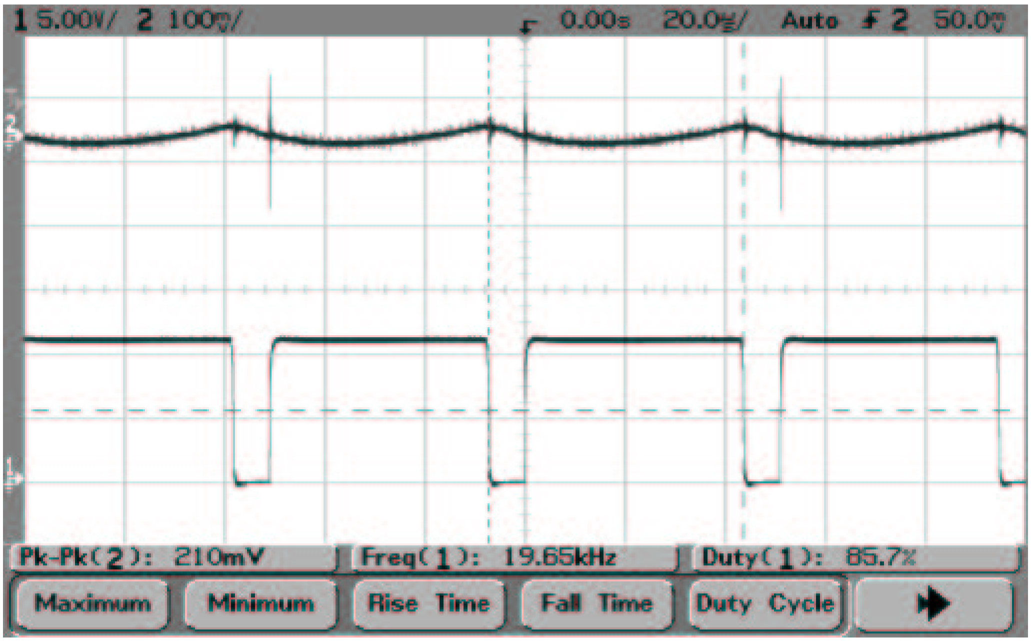


**Figure 7.13:** The output ripple (Upper signal) and gate signal (lower signal) with maximum load and 7.2V input

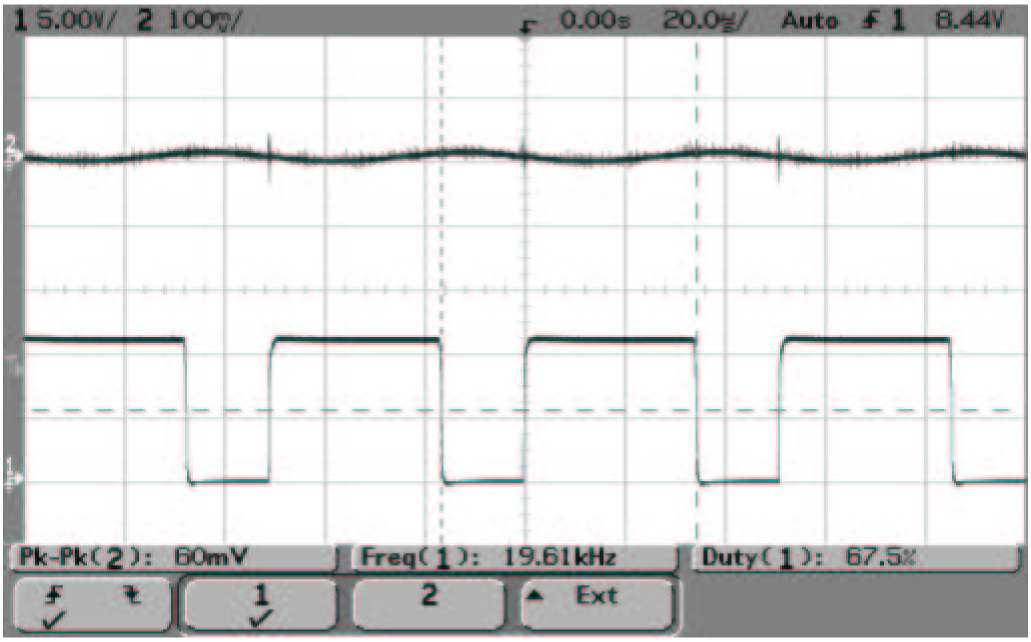


**Figure 7.14:** The output ripple (upper signal) and gate signal (lower signal) with nominal load and 7.2V input

time the switch switches state a step-signal is superimposed on the output LC-filter of the converter and in effect the step-response of the 2nd order filter can be seen on the output voltage.

The same is repeated with 16.7Ω (0.3A) and the result can be seen at figure 7.14.

Again the spikes are visible, but their amplitudes are smaller due to the fact that the converter is less loaded. The amplitude of the spikes are 60 mV which fulfills the requirement. The ripple when the spikes are neglected is less than 25 mV.

**Efficiency**

In the third test the the load resistance was set to 16.7Ω and the input voltage to first 6, then 7.2 and last 8.4 V and the duty-cycle was adjusted until 5V output voltage was reached. The input and output power was then measured and the efficiency was calculated:

$$6V: \quad \eta = \frac{1.58}{1.66}\cdot 100 = 95\%$$

$$\begin{aligned} 7.2V: \quad \eta &= \frac{1.54}{1.62} \cdot 100 = 95\% \\ 8.4V: \quad \eta &= \frac{1.53}{1.59} \cdot 100 = 96\% \end{aligned}$$

The same was repeated with 2Ω as load:

$$\begin{aligned} 6V: \quad \eta &= \frac{12.27}{16.58} \cdot 100 = 74\% \\ 7.2V: \quad \eta &= \frac{12.31}{16.03} \cdot 100 = 77\% \\ 8.4V: \quad \eta &= \frac{12.56}{16.40} \cdot 100 = 77\% \end{aligned}$$

As it can be seen the converter fulfills the requirements at the nominal load with an efficiency of about 95%, but the loss is much larger at maximum load at about 75%. It should however be noted that due to the missing gate-drivers, these efficiencies does not account for losses due to gate-charging.

# Chapter 8
# Overcurrent Protection

## 8.1 Requirements

During this chapter the over current protection circuit, hereafter referred to as OCPC, will be developed. The purpose of the OCPC is to stop the current flow to a certain user in case of a failure. This is done to minimize damage and power loss. The chapter will begin by defining the requirements. The design is then made and finally the design is implemented and tested.

### 8.1.1 Overview

The protection circuits are placed as seen in figure 8.1. The input is the regulated voltage from PCC. The OCPC is a circuit capable of switching off a user if it is drawing too much current. The OCPCs will be placed between the filter and the user. There has to be one OCPC for every user. The MCU has to be able to switch off the individual users.

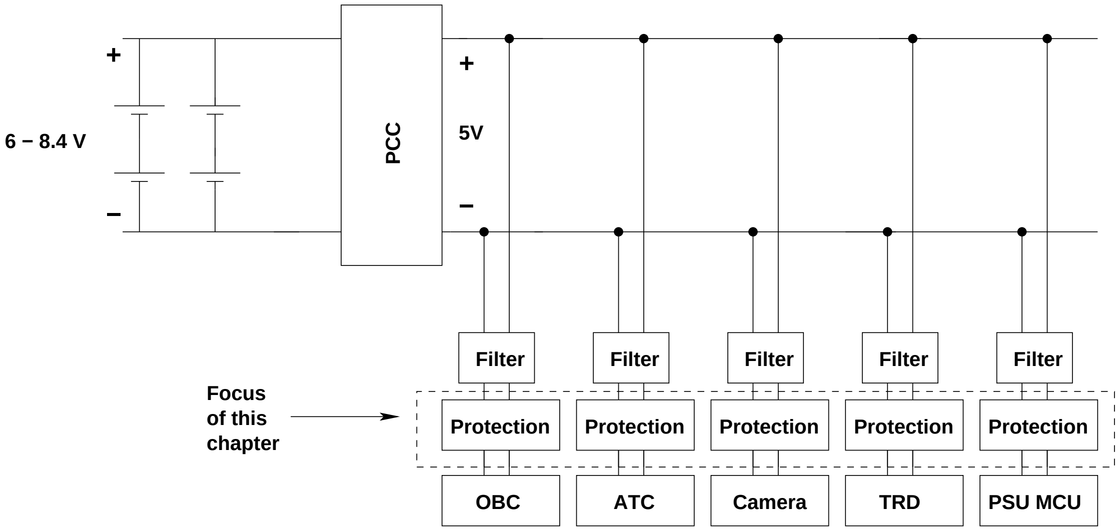


**Figure 8.1:** An overview of the OCPC

The primary purpose for the OCPC is protection in case of a SEL[1] , see chapter 2 on page 21. This occurs if e.g. the non-conducting transistor in a totem pole is hit by an energized particle and therefore starts to conduct. In this case both transistors will conduct and a short-circuit will occur. The protection can be achieved by turning the short-circuiting circuit off and after a while turn it back on. In case a transistor unwantedly becomes conductive because of SEL, the OCPC has to wait and then try to turn the device back on, because there is a possibility that the atoms inside the transistor that was hit by the electron has found their way back and the transistor works again.

The OCPC for the MCU differs from the rest of the OCPCs in two points; the current needed to trigger the OCPC and the fact that the OCPC has to be able to switch back on the MCU, since the MCU can not do it itself.

### 8.1.2 Specific Requirements

In cases where the OCPC has to turn off user, it has to signal the event to the OBC through the MCU. The OBC can then decide whether to leave the equipment switched off or switch it back on. This is done to ensure bringing back of equipment that failed due to SEL without damaging it. After the OCPC has switched off the equipment and sent a signal to the MCU, it awaits the switch back on signal from the MCU. The OCPC then switches back on the equipment expecting that the equipment is back in order.

The function of the over current circuit can be seen i figure 8.2. The OCPC constantly measures the output current. The measured value is compared to an internal reference and if the current flow is too big the device is

[1] Single event latch-up

turned off. In case of a shut down, the MCU is notified and the turned off device is not turned back on until the MCU signals to do so.

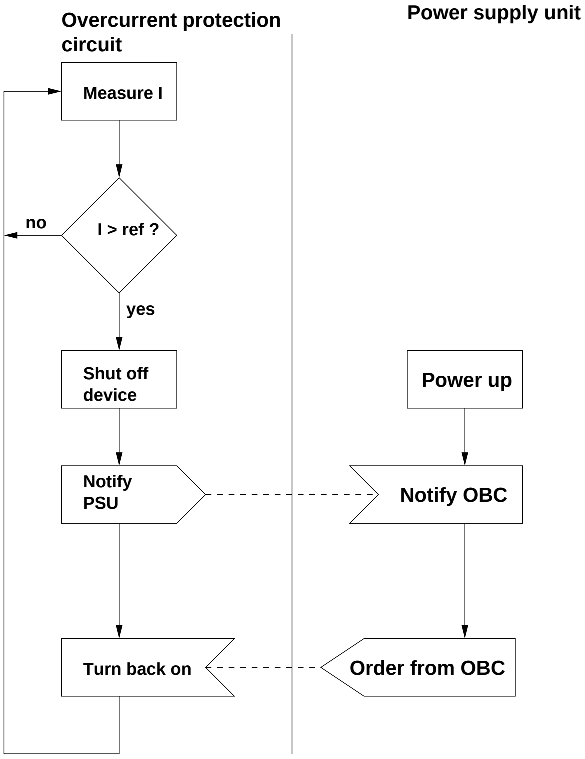


**Figure 8.2:** The flow of the overcurrent circuit. The figure show the different states that the overcurrent circuit can be in

The OCPC for the camera, TRD, ATC and OBC are of the same kind. The only difference is the current needed to switch off the individual user. The different currents and voltages are listed in table N.1 on page 234. The OCPC for the PSU has to be able to reconnect the PSU after a while. If a SEL occurs and the OCPC cuts off the PSU, the OCPC has to try to bring back the PSU.

### 8.1.3 Interface to the MCU

The interface to the MCU consist of two control signals for each OCPC, except the OCPC for the MCU itself. The two control signals are; turn on/off signal from MCU to the OCPC and status on/off from the OCPC to the MCU.

In cases were the OCPC has turned the user off the MCU has to use this signal in order to turn back on the user. The status on/off is used to signal the current status of the OCPC. It is via this signal that MCU can read the status of the OCPC.

### 8.1.4 Interface to the Users

The output from the PSU is 5 V. All users are connected to the 5 V bus. The different currents are listed in table N.1 on page 234. The power used by the PSU refers to the power to the MCU inside the PSU.

## 8.2 Design and Realization

This section describes the realization of the over current protection circuits. The realization of the OCPC for the camera, modem, OBC, TRD and ACS (besides from the current needed to trigger the OCPC) are the same and will be described in the same section. The OCPC for the MCU will, due to its differences, be described in its own section.

### 8.2.1 Realization of OCPC for the Camera, OBC, ACS and TRD

The OCPC for the power bus will be designed in this section. Considering the demands for the over current circuit it is possible to develop the block diagram shown in figure 8.3. The diagram shows the input current on the left hand side; this is also the output from the PC converter. This input is measured and sent via a switch to the user. The switch is controlled by the control logic. The control logic is also connected to the PSU. The comparator compares the measured value with a reference and in case of too high current flow give a signal to the control logic.

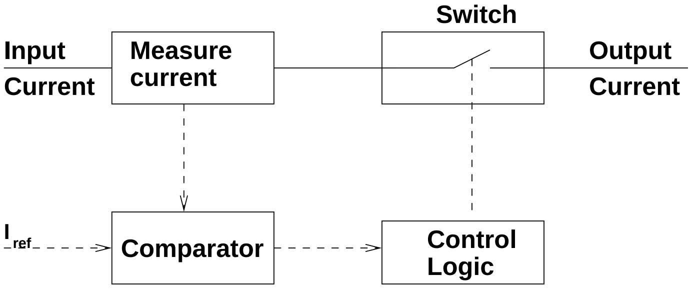


**Figure 8.3:** The function of the different blocks making up the over current protection circuit for the camera, OBC, TRD and ACS

### Measuring

The measuring device is a shunt resistor, in series with the current flow. Measuring the voltage on both sides of this shunt resistor gives a voltage that is proportional to the current flow. The power-loss is dependent on the current and the value of the shunt resistor. Since the current flow cannot be limited, the only thing to minimize in order to limit power loss, is the value of the shunt resistor. Minimizing the value of the shunt resistor lower the power loss, but will also lower the measured value.

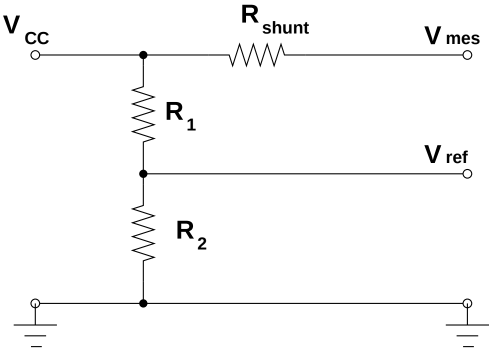


**Figure 8.4:** The shunt resistor and voltage divider. The voltage divider makes the reference voltage, that is to be compared to the measured voltage

Since the OCPC is to be used for users, needing a certain current limitation, it is necessary to compare the measured value to a reference voltage. This reference represents the maximum current flow. Normally a zener diode and a resistor is used, but since the output voltage has a ripple of only 2 %, see section 3.3.1.1 on page 26, a voltage divider, consisting of two resistors, can be used [2]. The solution with the zener diode has the drawback that it depends on the zener-value and can only be controlled stepwise corresponding to the values of the available zener-diodes.

The solution with two resistors as a voltage divider can be seen as $R_1$ and $R_2$ in figure 8.4. This figure also shows the shunt resistor.

### Switch

As switch a PMOS transistor is used. A NMOS generally has a smaller $R_{on}$ and therefore a smaller power loss, but due to the low battery voltage a PMOS is used. This is done in order to ensure that there is a voltage high enough to open the transistor without having too big a power- and voltage loss across the transistor. A SUB75P03-07 is chosen because of its low $R_{on}$ [SUB75P03, 2001]. To save power the switch transistor is used as the shunt resistor and the voltage is measured over this transistor, as can be seen in figure 8.5.

The voltage on the source will be 5 V and the gate will change between 5 and 0 V. At a $V_{cc}$ of 5 V the $R_{on}$ will be approximately 0.01 Ω and therefore the transistor can be used as a shunt resistor.

[2] Assuming that the comparator does not draw any, or a very small, current from the voltage divider

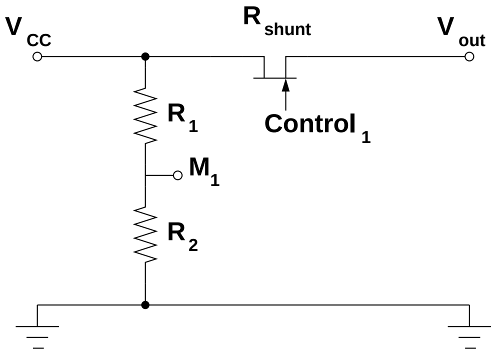


**Figure 8.5:** The OCPC switch. As switch a PMOS transistor is used, the $V_{cc}$ will switch between 0 and 5 V. $R_1$ and $R_2$ makes the reference to the maximum current.

### Comparator

To compare the reference and the current a comparator is used. While choosing this comparator several things were under consideration:

1. Low response time
2. Rail to rail or open drain / collector output
3. Single low supply voltage
4. Possibility of quad housing

Low response time ensures that users will not have time to be damaged before being switched off. Rail to rail in- and output ensures maximum output swing and the possibility of comparing voltages close to $V_{cc}$ and ground of the comparator. A single low supply voltage ensures that the comparator can function even if the supply voltage drops below the lower limit. The reson for possibility of quad housing is to save space on the PCB, by having four comparators housed together. The four comparators housed together have to be part of the OCPC for the OBC, ATC, camera and the TRD user.

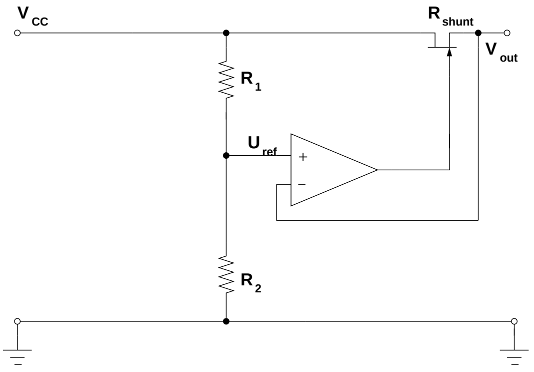


**Figure 8.6:** The comparing of the $V_{out}$ and the current flow. The current flow is converted into a voltage drop over the transistors internal resistance

The final choice is the AD8564 which is the IC that fulfilled the demands above best. This IC has to compare the two voltages $V_{ref}$ and $V_{out}$. If the connected device starts to draw a current the voltage ($V_{out}$) will fall, due to the resistance in the transistor. The $V_{out}$ is 5 V and drops proportionally to the current drawn. For example if the internal resistance in the transistor is 0.004 Ω and the connected device draws a current of 0.1 A, the resulting voltage drop will become:

$$Voltage\ drop = 0.004 \cdot 0.1 = 0.0004\ [V] \tag{8.1}$$

The outgoing voltage, $V_{out}$, will then drop 0.4 $mV$. This drop should then, in the case where the 1 $mV$ is too much, cut off the device.

Since the switch is a transistor of the PMOS type the gate voltage has to be raised if the connected device draws to high current. In situations where the connected device does not draw too high current, the gate voltage has to be lowered in order to open the transistor. Figure 8.6 shows the diagram for the comparator. The relationship, $R_1$ and $R_2$, sets the reference and therefore the maximum current draw allowed to the connected device.

### Control Logic

To perform the task of the control logic a comparator as bistable multi vibrator is used. As can be seen in figure 8.7 the control task is to switch between the two states: On and off. In the on state two things can trigger the OCPC to switch the connected device off:

1. A current greater than the reference

2. A turn off signal PSU MCU

A current greater than reference occurs if the connected device draws to much current. The turn off signal is a signal from the PSU to turn off a certain unit.

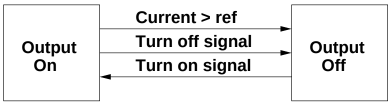


**Figure 8.7:** The two states of the control logic. The MCU is capable of switching off a user and has to reactivate it, in case of a fault

The flow of the control logic can be seen in figure 8.7. If an over current occurs, the only thing that is to be able to switch back on the power is the MCU. During normal operation, the MCU has to be able to switch off the user.

The circuit for the OCPC can be seen in figure 8.8. The circuit is basicly a comparator which can only switch the output to high. In order to switch back on, a positive pulse on the input of C is needed. The input pulse will generate a positive pulse at the positive input of the comparator and thereby switch it back on.

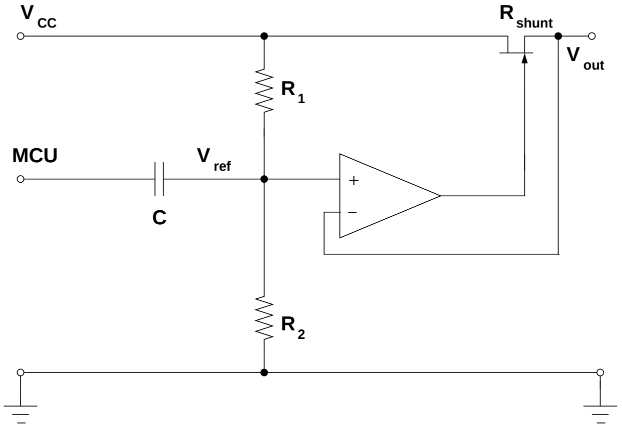


**Figure 8.8:** The overall control logic. The same system will be used for the OBC, ATC, TRD, camera and the COM users

When the system is in steady state, the transistor is on and thereby conducts power to the connected device. If the connected device starts drawing a current, the voltage drop across the transistor increases proportionally with this current, forcing the negative terminal of the comparator down. At a certain level the voltage drop on the negative terminal affects the comparator[3] and the output becomes high, switching off the transistor. Since the MCU monitors the output voltage with a given interval the power fault will be detected. When the MCU decides to switch back on the power, it sends a positive edge on the output and thereby generate a positive edge on the negative terminal of the comparator. This positive edge will set the output of the comparator high and

[3] When the voltage on the negative terminal becomes lower than the one at the positive terminal

allow the transistor to conduct. The pulse from the capacitance will drop and thereby allow the comparator to measure the current - the circuit is back to its initial state.

If the MCU wants to switch off a certain user, it generates a negative edge on the negative terminal of the comparator. This negative edge will act as if the user is drawing too much current and the comparator will switch the transistor off.

**Component Values**

Since the OCPC is the same for the OBC, ATC, camera, TRD and COM, except from the current limit, the same circuit is used for these users. To make the voltage for the positive terminal a voltage divider is used. The value 1 M $\Omega$ is chosen for $R_2$, the value of $R_1$ is different for the individual users. The value of $R_1$ is calculated as follows:

$$V_q = I_{max} \cdot R_{on}\,[V] \tag{8.2}$$

$$V_{cc} - V_q = \frac{V_{cc} \cdot R_2}{R_1 + R_2}\,[V] \tag{8.3}$$

$$R_1 = \frac{V_{cc} \cdot R_2}{V_{cc} - V_q} - R_2\,[\Omega] \tag{8.4}$$

Where:
$V_q$ = The maximum voltage over the transistor [V]
$V_{cc}$ = The voltage from the PCC, this is 5 V $\pm 1$ %
$I_{max}$ = The maximum current drawn by the user [A]

The component values are listed in the table 8.1:

| | $I_{max}$ | $\Delta U$ | $R_1$ | $R_2$ | C |
|---|---|---|---|---|---|
| OBC | 180 mA | 1.8 mV | 360 Ω | 1 MΩ | 100 nF |
| ATC | 200 mA | 2.0 mV | 400 Ω | 1 MΩ | 100 nF |
| Camera | 50 mA | 0.5 mV | 100 Ω | 1 MΩ | 100 nF |
| COM | 30 mA | 0.3 mV | 60 Ω | 1 MΩ | 100 nF |
| TRD | 900 mA | 9.0 mV | 1803 Ω | 1 MΩ | 100 nF |

**Table 8.1:** The component values for the OCPCs

Since normal resistors have a tolerance of 1%, the nearest value is chosen. The value of the capacitance decides the length of the pulse. The length of the pulse should be 4.5 ns or longer, since this is the response time for the comparator. This could be a problem since the power on current to the connected devices is bigger than under normal operation. The exact value of the capacitance can not be determined, before the users are designed or a maximum power on time is set.

In order to calculate the value of the capacitance the start up time[4] for each subsystem must be known, but since this time is not know all capacitances are chosen to 100 nF. The voltage on the MCU side of the capacitance is either 0 or 5 V.

**Power Consumption**

The total power used by one OCPC can be calculated as follows:

$$P_{resistors} = \frac{U^2}{R_1 + R_2}\,[W] \tag{8.5}$$

$$P_{transistor} = I^2 \cdot R\,[W] \tag{8.6}$$

$$\tag{8.7}$$

The different consumptions are listed below:

[4] The time were a subsystem is drawing a large start up current

| | $I_{max}$ | P Comparator | P Resistors | P Transistor | P |
|---|---|---|---|---|---|
| OBC | 180 mA | 40 mW | 25 $\mu$W | 324 $\mu$W | 40 mW |
| ATC | 200 mA | 40 mW | 25 $\mu$W | 400 $\mu$W | 40 mW |
| Camera | 50 mA | 40 mW | 25 $\mu$W | 25 $\mu$W | 40 mW |
| COM | 30 mA | 40 mW | 25 $\mu$W | 9 $\mu$W | 40 mW |
| TRD | 900 mA | 40 mW | 25 $\mu$W | 8.1 mW | 48 mW |
| Total | 1350 mA | 200 mW | 125 $\mu$W | 8.86 mW | 209 mW |

**Table 8.2:** The power used by the OCPC to the different subsystems

### 8.2.2 Realization of OCPC for the PSU

The OCPC has to contain the following users:

1. Bootstrap power supply
2. Over current protection

The bootstrap power supply has to be able to power up the entire PSU. This is necessary because the controlled busses are dependent on the MCU. When the kill switch is deactivated and the power is feed from the solar array and/or the batteries the MCU has to boot up and control the two busses. At the time where the kill switch is deactivated, it is not certain whether the batteries are charged or the solar arrays are capable of producing power.

The circuit in figure 8.9 is working almost as the circuit used for the other users. When the power is on, and the MCU does not draw too much current the output from the comparator is low. Since the system has been working for a while the capacitance has charged so that the voltage over the capacitance, $V_C$, is $V_{cc}$. The voltage at $V_{ref}$ sets up the maximum current the MCU is allowed to draw. If at SEL occurs the voltage at $V_{out}$ will drop immediately and thereby pull down the voltage at the negative terminal of the comparator. The difference of the voltages on the input on the comparator makes the comparator change state an close down the MCU. The change on the output of the comparator and the 0 volt on the $V_{out}$ makes the RC connection start charging and at the point where the negative terminal is slightly higher then the positive the comparator will change state and thereby switch back on the MCU. The switching back on of the MCU is delayed by the RC connection and the delay is calculated by the equation 8.10. Since the charging current for the capacitance is drawn from the output of the comparator is is important to keep $R_3$ as big as possible. The minimum value of this resistor is calculated by the means of equation 8.8.

$$R_3 > \frac{U_{high}}{I_{max}} \Leftrightarrow R_3 > \frac{8.4}{0.0032} \Leftrightarrow R_3 > 2625\ \Omega \tag{8.8}$$

Where:
$U_{high}$ : is the maximum voltage at a high output from the comparator
$I_{max}$ : is the maximum current that the comparator can source

For $R_3$ a 1 M$\Omega$ resistor is used. In order to calculate the value of C equation 10.5 on page 119 is used, in rewritten form:

$$R_3 = \frac{-T}{ln\left(1 - \frac{U_{cap}}{U}\right) \cdot C} \tag{8.9}$$

$$C = \frac{-T}{ln\left(1 - \frac{U_{cap}}{U}\right) \cdot R_3} \tag{8.10}$$

Where:
$U_{cap}$ : is the voltage that the capacitance is charging to, in this case the voltage on that other input terminal of the comparator
$U$ : The high out from the comparator

Since the time the MCU needs in order to discharge probably is not known, the capacitance is chosen to 100 nF. This gives a delay of 2.55 ms, before the MCU will receive power again.

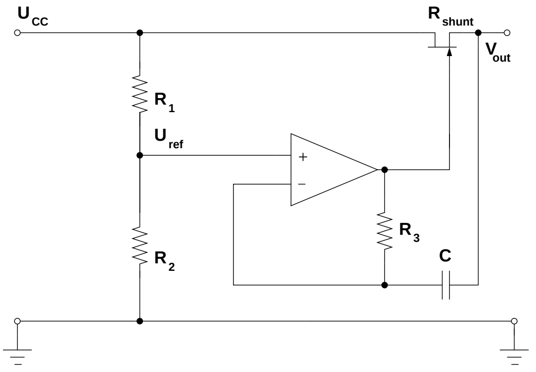


**Figure 8.9:** The OCPC for the MCU. In case of a SEL the voltage at $V_{out}$ will drop immediately, not allowing the capacitance to change its charge

### Component Values for the MCU OCPC

The calculation of the two resistors is performed as calculating the resistors for the OCPC for the other users, see equation 8.4. The values are listed in table 8.3. In order to calculate the values of $R_3$ and C, the time in which the MCU is to be kept unpowered must be known.

| $R_1$ | $R_2$ | $R_3$ | $C$ |
|---|---|---|---|
| 60.004 Ω | 1 MΩ | 1 MΩ | 100nF |

**Table 8.3:** The power used by the OCPC to the different subsystems

## 8.3 Implementation

During the implementation is was discovered that the comparator could not compare two voltage correctly if they were close to the $U_{cc}$. Since the two voltages to be compared both are close to 5 V, and since the last converter has a voltage drop of minimum 1 V it was decided to connect the OCPC directly to the batteries instead of to the 5 V bus. This made the OCPCs functionally but will result in a greater power consumption.

As is evident from the tables on current consumption in this chapter, the solution designed use too much power to be implemented on the Cubesat. It should however be noted that the design presented here uses the fastest available comparator in order to switch off power as fast as possible. Other comparators, which uses many times less current than the AD8564 used here, can be used in the same design, but they will result in longer switch off times.

Therefore, if this solution is to be implemented on the Cubesat then an individual comperator for each load should be selected. This comperator must be selected as a trade-off between current consumption, reaction times and importance of the subsystem.

## 8.4 Test Results and Evaluation

In this section the result from the module-test of the OCPC will be described. The test was made according to a test plan that can been seen in appendix J.3 on page 216. In the test the following three functions were tested:

1. Power up close down
2. MCU control
3. Over current switching

All of these test are performed with at variable resistor as load.

### Power Up Close Down

The test was done according to the test plan in appendix J.3.3.1 on page 216. The test was performed several times and all times gave the right output, in this case none.

**MCU Control**
The test was done with at control signal of 10 kHz. The test showed that the circuit is capable of being switched off by a negative edge and to be switched on by a positive edge and thereby being able to be controlled by the MCU.

**Overcurrent Switching**
The test was performed as if the load was the ATC. As $R_1$ a 402 Ω resistor was used and a 1 MΩ for $R_2$. The load was then increased (the current flow was raced by decreasing the resistor). The OCPC switched off the load at a current flow just about 180 mA. The early switching off is due to noice from the power supply.

# Chapter 9
# Converter Control Design

## 9.1 Overview

In this chapter the control of the MPPTC and the PCC will be designed. First a general model of the system will be discussed in order to identify the different modes the controller have to operate in. After that first the MPPTC and next the PCC will be analyzed in order to identify the dynamic transfer functions of the converters. These transfer functions will then be used to find control types and parameters. Then the controllers will be designed and tested through simulation.

## 9.2 General Control Model

The two converters have different tasks to perform and therefore need different control. The job of the MPPTC is to perform maximum power point tracking on the solar cells and to charge the batteries. When the batteries is fully charged the charging must be stopped, i.e. the output current from the MPPTC must be limited in order to keep the voltage of the batteries constant. The reference of the control then shifts from the input current to the output voltage (battery voltage). Thus the control of the MPPTC can operate in two basic modes:

1. Maximum power point tracking

   - Operation for a current and voltage on the input ($V_{in}$ and $I_{in}$)

2. Specific Output Voltage

   - Operation for the maximum battery voltage on the output ($V_{bat}$)

The PCC controller has the task of keeping the output voltage constant and therefore it can only operate in one mode:

1. Constant output voltage

   - Operation for a specific voltage on the output ($V_{bus}$)

In both cases the only controllable input is the duty cycle on the switch. The regulation in its basic form can be seen on figure 9.1.

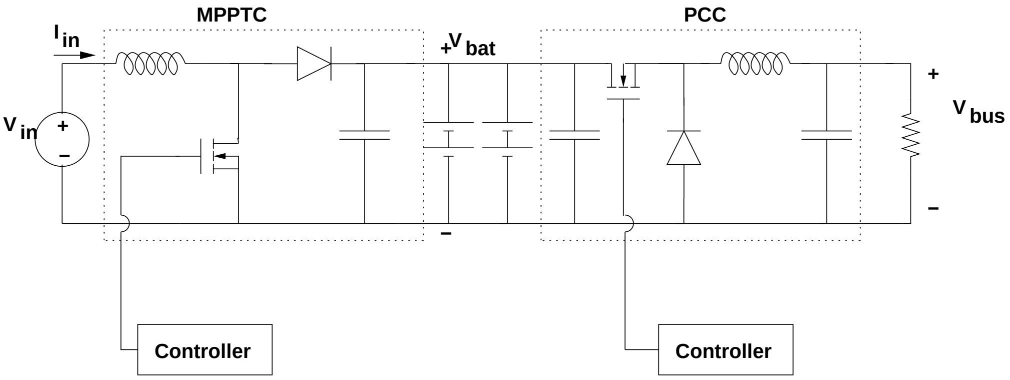


**Figure 9.1:** The two converters with controllers

In the following the transfer function of a system is denoted $G(s)$, the open-loop transfer function of a system is denoted $G_o(s)$ and the transfer function of a controller is denoted $D(s)$. The closed-loop transfer function of a system is denoted $T(s)$ and a sensor transfer function is denoted as $H(s)$.

### 9.2.1 Control Types

There are two basic methods to control a system, either with or without feedback (see figure 9.2).

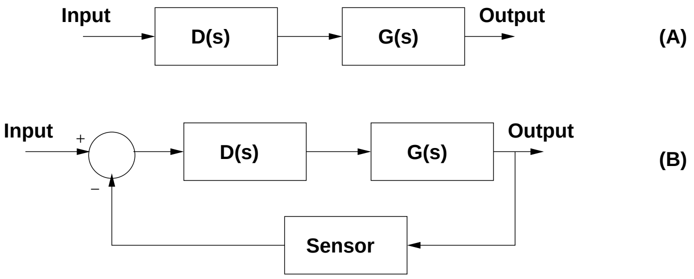


**Figure 9.2:** Two different kinds of control: Open-loop (A) and Closed-loop (B)

When feedback is not used it is essential to have a good knowledge of the system that must be controlled, because there is no way of dynamically adjusting the controller when it is operating. The lack of feedback makes the control system easy to construct and it also gives a "stable system", however since it is difficult to know everything about the system it will often also give an inaccurate output. This control method is also known as open-loop control.

When feedback is used there is not a great need for precise knowledge of the system that needs to be controlled, because errors are suppressed through the feedback loop. The feedback loop however can cause difficulties with stability in the system and thereby increase the risk of oscillations in the system. In this case it is clear, that because of all the unknown factors in the system, like changing loads and temperatures, a feedback loop also known as a closed-loop system is needed.

The closed-loop system shown at figure 9.2 is known as a series controller because there is only one feedback loop in the system, i.e. from the output of the system. This means that all disturbances in the system are compensated for at only one point. It is however possible to place several series controllers inside each other, and thereby creating what is know as a cascaded controller (see figure 9.3).

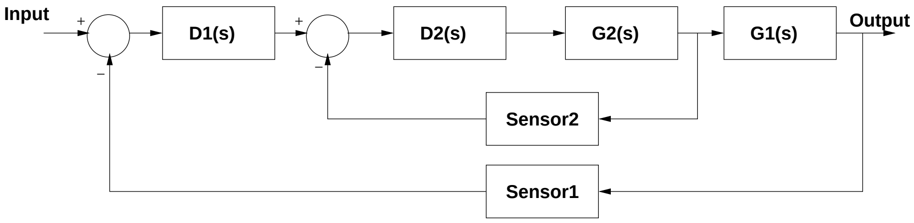


**Figure 9.3:** A cascaded controlled system divided into two subsystems

The cascaded controller gives in most cases a more robust system which is more tolerant of dynamic disturbances, because of the several feedback loops [Haugen, 1994] page 512. When using cascaded control it is important that the inner loop is significantly faster that the outer loop and if this is not the case it is better to use only one loop [Heilmann, 1992] page 234.

### 9.2.2 The Basic PID Controller

One of the most common ways to control a system is to use the PID controller, which consists of three different controllers: A proportional controller (P), an integral controller and a derivative controller (D). The PID controller transfer function is given as [Gene F. Franklin and Emami-Naeini, 1994] page 185:

$$u(t) = K\Big(e(t) + \frac{1}{T_I}\int_{t_0}^{t} e(t)\;dt + T_D\frac{de(t)}{dt}\Big) \;\rightarrow D(s) = K(1+\frac{1}{T_I s} + T_D s) \tag{9.1}$$

It is possible to derive different kinds of controllers from the PID and they are examined following.

#### The basic P controller

This controller multiplies the input signal with a constant. Thus, its transfer function is:

$$u(t) = Ke(t) \;\Rightarrow D(s) = K \tag{9.2}$$

Where:
$K$ is the gain of the controller.

The P controller is the simplest of all the controllers, but it has some disadvantages. A system with a proportional control will in most cases be unable to reject constant disturbances, i.e. a step, and the system will therefore show a steady state error.

**The basic PI controller**
The proportional-integral controller can be used to eliminate the steady state error. The transfer function of the I-controller is:

$$u(t) = K\Big(e(t) + \frac{1}{T_I}\int_{t_0}^{t} e(t)dt\Big) \rightarrow D(s) = K(1 + \frac{1}{T_I s}) \tag{9.3}$$

Where:
$T_I$ is the integral time, that is the time it takes the integrator output to reach a value of $K$ when the input to the controller is unity.

A disadvantage of the PI controller is that the integral part slows down the transient response of the controller.

**The basic PD controller**
The proportional-derivate controller is used to increase the damping and the stability of the system. It can eliminate a change in an error, but will not correct a constant error as the PI controller. It has the following transfer function:

$$u(t) = K\Big(e(t) + T_D\frac{de(t)}{dt}\Big) \rightarrow D(s) = K(1 + T_D s) \tag{9.4}$$

Where:
$T_D$ is the derivative time.

A PD controller improves the transient response, but this ability can cause instability in system with large amounts of noise.

## 9.3 Control Systems for MPPTC and PCC

Because of the different tasks for the converters a different approach is used for the controller design for the two. The MPPTC will be controlled directly by the MPPT-controller, while the PCC will be controlled by more conventional control methods. When controlling DC-DC converters it is desirable to do this through current controlled loops because this increases the stability of the controlled converter. Therefore, the PCC will be controlled with a closed-loop cascaded control scheme, where the inner loop will be current controlled.

**Control of MPPTC**
The role of the MPPTC in the PSU is to draw the maximum power out of the solar cells using MPPT, and secondly to keep the batteries from over-charging. To maximize the power, it can be measured and used as input in the MPPT-control while the output controls the duty cycle of the MPPTC. This will result in a change in the voltage level over the solar array, and the current being drawn from it. The measurement of the power would traditionally be done by measuring the voltage over the panels and the current being drawn from them. This configuration can be seen in figure 9.4, as the dotted measuring points.

Because the output of the MPPTC is directly linked to a battery, it can be assumed that over a short time period the voltage level will remain constant. Therefore, a variation in the measured output current (measured at the non dotted current measurement point on figure 9.4), will be proportional to the variation in the output power from the MPPTC. The MPPT can then attempt to get the maximum current out of the MPPTC, which will also give a maximum output power.

The method of measuring the power through just the output current of the MPPTC, has several advantages over the double measuring point method, the main one being that only one measuring point is needed for the MPPT. This means that less computation is needed in the MPPT algorithm. The disadvantage of this method is the assumption that the voltage is constant,which is an approximation, since there is an expected voltage ripple of $2\%_{p-p}$ (see chapter 6 on page 41). This again affects the precision of the power measurement. In spite of this, the single measuring point method is chosen because of its simplicity.

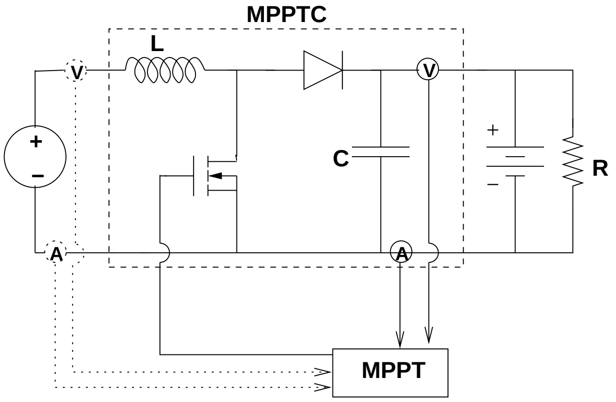


**Figure 9.4:** The MPPTC with control system with two different measuring point sets

To keep the MPPTC stable, the frequency of the MPPT must be adapted to the settling time of the MPPTC. Therefore the systems response in output current to a change in the duty cycle needs to be found. To do this the following transfer function is needed:

$$MPPT: \; G \;=\; \frac{\hat{i}_o(s)}{\hat{d}(s)} \tag{9.5}$$

To achieve the secondary task of preventing over-charging of the battery, it is necessary to measure the voltage level of the battery. This measurement point can be seen as the solid line voltage measurement point on figure 9.4.

**Control of the PCC**

The inner loop of the cascaded control of the PCC controls the duty cycle of the converter from the inductor current. The outer loop controls from the output voltage and delivers the reference for the inner loop. In order to improve the performance of the controller a feed forward is introduced in the control scheme. This feed forward is the output current which is fed directly into the fast inner loop. Thereby changes in the load current can be detected and compensated for before the voltage on the output capacitor can change and the slower outer loop detects the changes. The control of the PCC can be seen on figure 9.5.

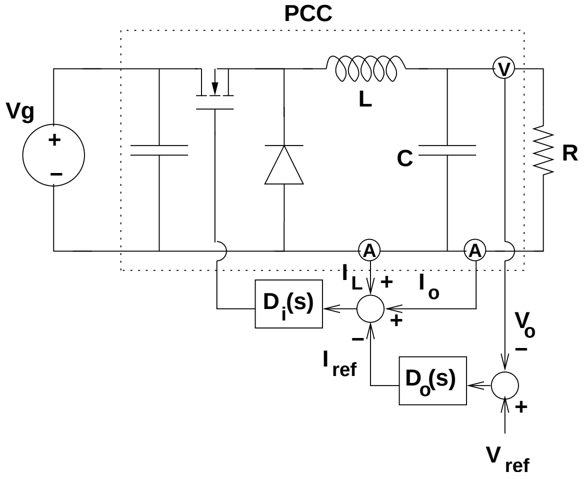


**Figure 9.5:** The PCC with controller system

From the above description, two transfer functions for the PCC can be identified as necessary for the control (see figure 9.6):

$$Inner\ loop: \; G_i \;=\; \frac{\hat{i}_L(s)}{\hat{d}(s)} \tag{9.6}$$

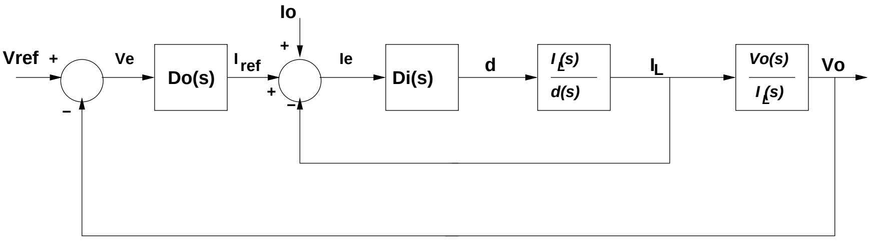


**Figure 9.6:** Block diagram of the PCC and its controller

$$Outer\ loop: \ G_o = \frac{\hat{v}_o(s)}{\hat{i}_L(s)} \tag{9.7}$$

## 9.4 Transfer Functions

In order to control the converters it is essential to find their transfer functions for the converters. This is done through an AC analysis of the two converters. In a small signal analysis, the average frequency values is studied and the switch ripple is ignored. The average value of a signal $x(t)$ over an interval $T$ is described:

$$\langle x(t) \rangle_T = \frac{1}{T} \int_{t-T}^{t} x(\tau) d\tau \tag{9.8}$$

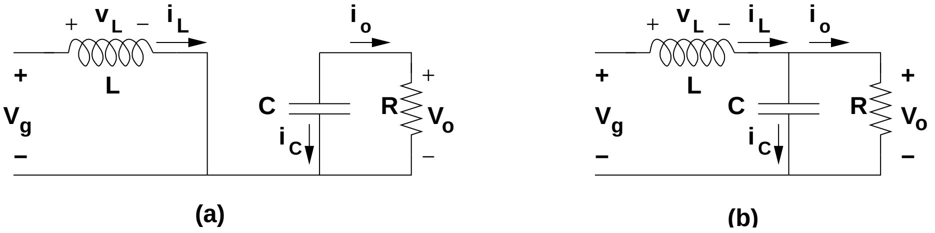
When looking at one switch period ($T_s$) it is important to state the relationship between the duty cycle (the on time of the switch) and its complementary (the off time of the switch):

$$d'(t) = 1 - d(t) \tag{9.9}$$

The dynamic modeling described in the two following sections is based primarily on the approach introduced in [Erickson, 1999] page 198. In this approach assumes that the converters operate in CCM and in the following only the ideal converters will be considered, i.e. conduction losses, etc. will be neglected.

### 9.4.1 Dynamic Modeling of MPPTC

The converter is operating in CCM mode, and the switching ripple is neglected. Therefore, the small ripple approximation is used. Figure 9.7 shows an operating Boost converter.



(a) (b)

**Figure 9.7:** Diagram of the MPPTC converter operating in the two possible switch states

The inductor voltage and capacitor current are examined accordingly in both state a and b:

$$a: \quad v_L(t) = v_g(t) \tag{9.10}$$

$$i_C(t) = -i_o(t) = -\frac{v_o(t)}{R} \tag{9.11}$$

$$b: \quad v_L(t) = v_g(t) - v_o(t) \tag{9.12}$$

$$i_C(t) = i_L(t) - i_o(t) = i_L(t) - \frac{v_o(t)}{R} \tag{9.13}$$

The inductor voltage is given as:

$$\langle v_L(t) \rangle_{T_s} = L \frac{d \langle i_L(t) \rangle_{T_s}}{dt} \tag{9.14}$$

If the expressions for the inductor voltage during a complete switch period (equation 9.10 + 9.12) are inserted into equation: 9.14 the following are obtained:

$$\begin{aligned}
L\frac{d\langle i_L(t)\rangle_{T_s}}{dt} &= \langle v_g(t)\rangle_{T_s} d(t) + (\langle v_g(t)\rangle_{T_s} - \langle v_o(t)\rangle_{T_s})d'(t) \Leftrightarrow \\
L\frac{d\langle i_L(t)\rangle_{T_s}}{dt} &= \langle v_g(t)\rangle_{T_s}(d(t) + d'(t)) - \langle v_o(t)\rangle_{T_s} d'(t) \Leftrightarrow \\
L\frac{d\langle i_L(t)\rangle_{T_s}}{dt} &= \langle v_g(t)\rangle_{T_s} - \langle v_o(t)\rangle_{T_s} d'(t) \qquad (9.15)
\end{aligned}$$

Next the capacitor current is considered. The capacitor current is:

$$\langle i_C(t)\rangle_{T_s} = C\frac{d\langle v_o(t)\rangle_{T_s}}{dt} \qquad (9.16)$$

If the expressions for the capacitor current (equation 9.11 + 9.13) are inserted into equation 9.16 the following are obtained:

$$\begin{aligned}
C\frac{d\langle v_o(t)\rangle_{T_s}}{dt} &= -\frac{\langle v_o(t)\rangle_{T_s}}{R} d(t) + \left(\langle i_L(t)\rangle_{T_s} - \frac{\langle v_o(t)\rangle_{T_s}}{R}\right) d'(t) \Leftrightarrow \\
C\frac{d\langle v_o(t)\rangle_{T_s}}{dt} &= -\frac{\langle v_o(t)\rangle_{T_s}}{R}(d(t) + d'(t)) + \langle i_L(t)\rangle_{T_s} d'(t) \Leftrightarrow \\
C\frac{d\langle v_o(t)\rangle_{T_s}}{dt} &= -\frac{\langle v_o(t)\rangle_{T_s}}{R} + \langle i_L(t)\rangle_{T_s} d'(t) \qquad (9.17)
\end{aligned}$$

In the AC-analysis every time dependent quantity is modeled as a DC-component and a much smaller time dependent AC-component. This yields the following expressions:

$$\begin{aligned}
\langle v_L(t)\rangle_{T_s} &= V_L + \hat{v}_L(t) \quad where: V_L >> \hat{v}_L(t) \\
\langle v_g(t)\rangle_{T_s} &= V_g + \hat{v}_g(t) \quad where: V_g >> \hat{v}_g(t) \\
\langle v_o(t)\rangle_{T_s} &= V_o + \hat{v}_o(t) \quad where: V_o >> \hat{v}_o(t) \\
\langle i_L(t)\rangle_{T_s} &= I_L + \hat{i}_L(t) \quad where: I_L >> \hat{i}_L(t) \\
\langle d(t)\rangle &= D + \hat{d}(t) \quad where: D >> \hat{d}(t) \\
\langle d'(t)\rangle &= 1 - \langle d(t)\rangle = 1 - D - \hat{d}(t) \\
&= D' - \hat{d}(t) \quad where: D' >> \hat{d}(t) \qquad (9.18)
\end{aligned}$$

It is assumed that the AC-component is much smaller in magnitude than the DC-component and therefore the equations presented above are considered linearizations of the average value signals. If the expressions from 9.18 are inserted into equation 9.15 and 9.17 and thereby linearised, they yield:

$$\begin{aligned}
L\frac{d(I_L + \hat{i}_L(t))}{dt} &= V_g + \hat{v}_g(t) - (V_o + \hat{v}_o(t))(D' - \hat{d}(t)) \qquad (9.19) \\
C\frac{d(V_o + \hat{v}_o(t))}{dt} &= -\frac{(V_o + \hat{v}_o(t))}{R} + (I_L + \hat{i}_L(t))(D' - \hat{d}(t)) \qquad (9.20)
\end{aligned}$$

These two equation can then be expanded:

$$\begin{aligned}
L\Big(\frac{dI_L}{dt} + \frac{\hat{i}_L(t)}{dt}\Big) &= \underbrace{V_g - V_oD'}_{DC} - \underbrace{\hat{v}_o(t)D' + V_o\hat{d}(t) + \hat{v}_g(t)}_{1st\ order AC} + \underbrace{\hat{v}_o(t)\hat{d}(t)x}_{2nd\ order AC} \qquad (9.21) \\
C\Big(\frac{dV_o}{dt} + \frac{\hat{v}_o(t)}{dt}\Big) &= \underbrace{I_LD' - \frac{V_o}{R}}_{DC} - \underbrace{\frac{\hat{v}_o(t)}{R} - I_L\hat{d}(t) + \hat{i}_L(t)D'}_{1st\ order AC} - \underbrace{\hat{i}_L(t)\hat{d}(t)}_{2nd\ order AC} \qquad (9.22)
\end{aligned}$$

Since it was assumed that the AC-components were of much smaller magnitude than the DC-components, it is then clear that the second order AC-terms are negligible. The derivative of DC-terms are zero and therefore by definition the DC-terms on the right side have to equal zero as well. That leaves only the 1. order AC-terms:

$$\begin{aligned}
L\frac{d(\hat{i}_L(t))}{dt} &= \hat{v}_g(t) - \hat{v}_o(t)D' + V_o\hat{d}(t) \qquad (9.23) \\
C\frac{d(\hat{v}_o(t))}{dt} &= -\frac{\hat{v}_o(t)}{R} - I_L\hat{d}(t) + \hat{i}_L(t)D' \qquad (9.24)
\end{aligned}$$

**Transfer Function of MPPTC**

In order to find the transfer functions of the MPPTC the equations 9.23 and 9.24 are Laplace transformed.

$$\begin{aligned} sL\hat{i}_L(s) &= \hat{v}_g(s) - \hat{v}_o(s)D' + V_o\hat{d}(s) && (9.25)\\ sC\hat{v}_o(s) &= -\frac{\hat{v}_o(s)}{R} - I_L\hat{d}(s) + \hat{i}_L(s)D' && (9.26)\end{aligned}$$

Equation 9.25 is solved for $\hat{i}_L(s)$.

$$\hat{i}_L(s) = \frac{\hat{v}_g(s) - \hat{v}_o(s)D' + \hat{d}(s)V_o}{sL} \qquad (9.27)$$

This is inserted into equation 9.26 gives:

$$\begin{aligned} sC\hat{v}_o(s) = -\frac{\hat{v}_o(s)}{R} - I_L\hat{d}(s) \quad &+ \quad \left(\frac{D'}{sL}\hat{v}_g(s) - \frac{D'^2}{sL}\hat{v}_o(s) + \frac{V_oD'}{sL}\hat{d}(s)\right) \Leftrightarrow \\ \hat{v}_o(s)\left(sC + \frac{1}{R} + \frac{D'^2}{sL}\right) = \frac{D'}{sL}\hat{v}_g(s) \quad &+ \quad \left(-I_L + \frac{V_oD'}{sL}\right)\hat{d}(s) \Leftrightarrow \\ \hat{v}_o(s)\left(\frac{s^2LC + s\frac{L}{R} + D'^2}{sL}\right) = \frac{D'}{sL}\hat{v}_g(s) \quad &+ \quad \left(\frac{-s\cdot I_LL + V_oD'}{sL}\right)\hat{d}(s) \Leftrightarrow \\ \hat{v}_o(s) = \left(\frac{D'}{s^2LC + \frac{L}{R}s + D'^2}\right)\hat{v}_g(s) \quad &+ \quad \left(\frac{-s\cdot I_LL + V_oD'}{s^2LC + \frac{L}{R}s + D'^2}\right)\hat{d}(s) && (9.28)\end{aligned}$$

With the principles of superposition the transfer function $\frac{\hat{v}_o(s)}{\hat{d}(s)}$ can be obtained:

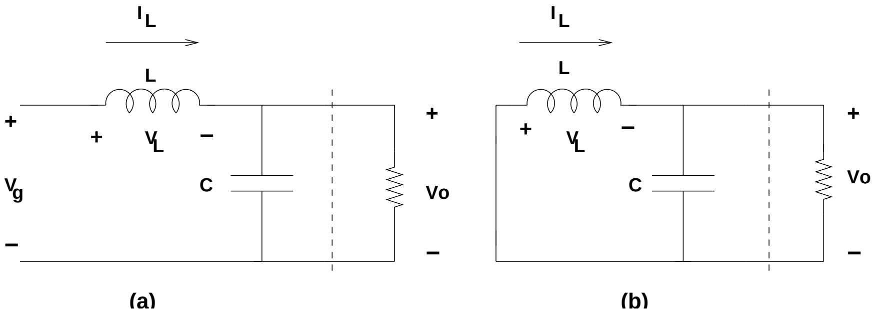
$$\left.\frac{\hat{v}_o(s)}{\hat{d}(s)}\right|_{\hat{v}_g(s)=0} = \frac{-s\cdot I_LL + V_oD'}{s^2LC + \frac{L}{R}s + D'^2} \qquad (9.29)$$

Since the desired transfer function is $\frac{\hat{i}_o(s)}{\hat{d}(s)}$ equation 9.29 is divided by the load resistance R to obtain the output current. This gives:

$$\begin{aligned} \frac{\hat{i}_o(s)}{\hat{d}(s)} &= \frac{\frac{-s\cdot I_LD' + V_oD'}{R}}{s^2LC + \frac{L}{R}s + D'^2} \Leftrightarrow \\ \frac{\hat{i}_o(s)}{\hat{d}(s)} &= \frac{-s\cdot I_LD' + V_oD'}{s^2\cdot LCR + s\cdot L + D'^2R} && (9.30)\end{aligned}$$

### 9.4.2 Dynamic Modeling of PCC

As described in chapter 7 on page 55 the PCC can operate in two states, see figure 9.8.



(a) (b)

**Figure 9.8:** Diagram of the PCC in the two operating states

The inductor voltage and the capacitor current is expressed by the following equations in respectively state a and b, with a switch period of $T_s$:

$$a: \quad i_C(t) \quad = \quad C\frac{dv_o(t)}{dt} = i_L - i_o \qquad (9.31)$$

$$v_L(t) = L\frac{di_L(t)}{dt} = v_g(t) - v_o(t) \tag{9.32}$$

$$b: \quad i_C(t) = C\frac{dv_o(t)}{dt} = i_L - i_o \tag{9.33}$$

$$v_L(t) = L\frac{di_L(t)}{dt} = -v_o(t) \tag{9.34}$$

From this the average value of the capacitor current and inductor voltage can be found during a complete switch period:

$$\langle i_C(t)\rangle_{T_s} = C\frac{d\langle v_o(t)\rangle_{T_s}}{dt} = d(t)(\langle i_L\rangle_{T_s} - \langle i_o\rangle_{T_s}) + d'(t)(\langle i_L\rangle_{T_s} - \langle i_o\rangle_{T_s}) \tag{9.35}$$

$$\langle v_L(t)\rangle_{T_s} = L\frac{d\langle i_L(t)\rangle_{T_s}}{dt} = d(t)(\langle v_g(t)\rangle_{T_s} - \langle v_o(t)\rangle_{T_s}) - d'(t)(\langle v_o(t)\rangle_{T_s}) \tag{9.36}$$

The average value equations presented are then linearized using the assumptions presented in section 9.4.1:

$$C\frac{d(V_o + \hat{v}_o)}{dt} = (I_L + \hat{i}_L(t)) - (I_o + \hat{i}_o(t)) \tag{9.37}$$

$$L\frac{d(I_L + \hat{i}_L(t)}{dt} = (D + \hat{d}(t))(V_g + \hat{v}_g(t)) - (V_o + \hat{v}_o(t)) \tag{9.38}$$

The equations are expanded:

$$C\Big(\frac{dV_o}{dt} + \frac{\hat{v}_o(t)}{dt}\Big) = \underbrace{I_L - I_o}_{DC} + \underbrace{\hat{i}_L(t) - \hat{i}_o(t)}_{1st\ order AC} \tag{9.39}$$

$$L\Big(\frac{dI_L}{dt} + \frac{\hat{i}_L(t)}{dt}\Big) = \underbrace{DV_g - V_o}_{DC} + \underbrace{D\hat{v}_g(t) + \hat{d}(t)V_g - \hat{v}_o(t)}_{1st\ order AC} + \underbrace{\hat{d}(t)\hat{v}_g(t))}_{2nd\ order AC} \tag{9.40}$$

As described in section 9.4.1 the 2nd order AC-terms and all DC-terms are neglected yielding the small signal model of the buck converter:

$$C\Big(\frac{\hat{v}_o(t)}{dt}\Big) = \hat{i}_L(t) - \hat{i}_o(t) \tag{9.41}$$

$$L\Big(\frac{\hat{i}_L(t)}{dt}\Big) = D\hat{v}_g(t) + \hat{d}(t)V_g - \hat{v}_o(t) \tag{9.42}$$

**Transfer Function of PCC**

The small signal model of the PCC represented by equation 9.41 and 9.42 are Laplace transformed:

$$sC\hat{v}_o(s) = \hat{i}_L(s) - \hat{i}_o(s) \tag{9.43}$$

$$sL\hat{i}_L(s) = D\hat{v}_g(s) + \hat{d}(s)V_g - \hat{v}_o(s) \tag{9.44}$$

In equation 9.43 $\hat{v}_o(s)/R$ is substituted for $\hat{i}_o(s)$ and it is then solved for $\hat{v}_o(s)$:

$$sC\hat{v}_o(s) = \hat{i}_L(s) - \frac{\hat{v}_o(s)}{R} \Leftrightarrow \tag{9.45}$$

$$\hat{v}_o(s) = \frac{\hat{i}_L(s)}{sC + \frac{1}{R}} \tag{9.46}$$

Equation 9.46 is the substituted into equation 9.44 and solved for $\hat{i}_L(s)$ :

$$sL\hat{i}_L(s) = D\hat{v}_g(s) + \hat{d}(s)V_g - \frac{\hat{i}_L(s)}{sC + \frac{1}{R}} \Leftrightarrow$$

$$\left(sL + \frac{1}{sC + \frac{1}{R}}\right)\hat{i}_L(s) = D\hat{v}_g(s) + \hat{d}(s)V_g \Leftrightarrow$$

$$\left(\frac{s^2CL + s\frac{L}{R} + 1}{sC + \frac{1}{R}}\right)\hat{i}_L(s) = D\hat{v}_g(s) + \hat{d}(s)V_g \Leftrightarrow$$

$$\hat{i}_L(s) = \frac{D(sC + \frac{1}{R})}{s^2LC + s\frac{L}{R} + 1}\hat{v}_g(s) + \frac{V_g(sC + \frac{1}{R})}{s^2LC + s\frac{L}{R} + 1}\hat{d}(s) \tag{9.47}$$

From equation 9.46 and 9.47 using the principle of superposition and the assumption that the average changes of the input current are very small, the transfer functions can be found:

$$\frac{\hat{v}_o(s)}{\hat{i}_L(s)} = \frac{1}{sC + \frac{1}{R}} \tag{9.48}$$

$$\left.\frac{\hat{i}_L(s)}{\hat{d}(s)}\right|_{\hat{v}_g(s)=0} = \frac{V_g(sC + \frac{1}{R})}{s^2LC + s\frac{L}{R} + 1} \tag{9.49}$$

## 9.5 Controller Design

As mentioned earlier in this chapter the controller design for the two converters will be very different from each other and therefore so will the design method. Because of the structure of the MPPT a traditional analysis is not needed, but as for the design of the controller to the PCC a design method has to be chosen.

There are three basic methods for designing a controller: Frequency response design, Root locus design and State space design. The first of the design methods is the simplest of the three and functions as a graphical trial and error method where bode plot are used to determine the compensation. The second method is also a graphical method using root-locus plot to determine the compensation, while the third method uses linear algebra. In the following the Frequency response design method will be used because of its simplicity and ability to provide good design in cases with uncertainties in the model [Gene F. Franklin and Emami-Naeini, 1994] page 337. The design goals are a stable system that is capable of suppressing changes in the load and input.

The approach for the design is as follows:

1. Determine requirements for control. There are three types of requirements to be determined for a control system: Steady state error (low frequency gain), stability (phase margin and damping) and speed (bandwidth, rise time and settling time).

2. Evaluate necessary compensation. Here the open- and closed-loop bode plots as well as the step response are used.

3. Apply compensation in form of P, PI, PD or PID to fulfill requirements.

4. Reiterate step 2 and 3 until all requirements are fulfilled.

In order to ease the design a Matlab file is created, which plots the uncompensated and compensated open-loop bode plot, closed-loop bode plot and step response. It is then possible for the designer through the iterative process to adjust the parameters and immediately see the result. The file can be found on the enclosed CDROM[1]. In the following documentation the iterative process will not be described, since this would be quite comprehensive, but instead will the result of the design will be discussed.

There is one general design criteria that apply for both converters and that is the sampling frequency of the controllers. The sampling frequency depends heavily on the bandwidth (i.e. speed of the system: settling and rise time) which is defined as the 3 dB corner frequency of the system and is a very important criterion, because it puts demands on the speed of the MCU. It is stated that the sampling frequency must be between 2 to 40 times the system bandwidth [Gene F. Franklin and Emami-Naeini, 1994] page 648. Thus, it is essential to keep the bandwidth as low as possible. The lower sampling frequency that is chosen, the more the stability of the system is jeopardized. A sampling frequency that is 15 times larger that the bandwidth decreases the phase margin with 10°[Haugen, 1994] page 565.

When choosing a low sampling frequency compared to the frequencies to the measured signal the problem of aliasing can arise and therefore a low-pass prefilter on the sampled signal is necessary. According to [Haugen, 1994] page 569 the corner frequency of the filter must be at most 1/4 of the samplings frequency when using a 1st order filter. This filter also helps to damp the ripple caused by the switching. If the corner frequency of the filter is not situated more than 1 decade above the bandwidth of the system it will affect the system and therefore it should be taken into account when designing the controller.

In the following the designs will be done first without taking the filters into account, and a samplings frequency and filter corner frequency will be determined. Then the filters will be applied and their effect on the system will be considered and if necessary the controller design will be reiterated.

[1] CDROM/Matlab/PCC_CONTROL.m

### 9.5.1 Controller for MPPTC

To carry out the MPPT-regulation a Perturbation and Observation (P&O) algorithm will be used. The reason for choosing the P&O algorithm is its efficiency and simplicity [Knopf, 1999] page 66. An illustration of the P&O algorithm functionality is shown in figure 9.9B.

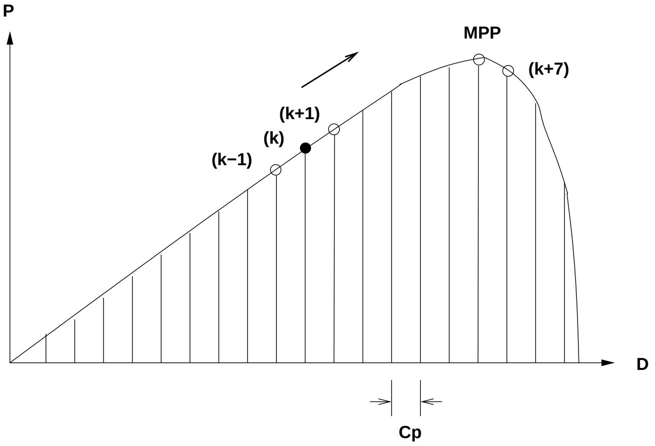


**Figure 9.9:** The functionality of the MPPT algorithm which moves position point $k$ step by step to the MPP. $C_p$ represents the duty cycle interval between the points

The control will begin by setting the duty cycle at a default value of 0.4, which is between the maximum and minimum expected duty cycle (see section 6.3.1 on page 45), and then measure the current at point ($k$). It will then change the duty cycle by $C_p$, and measure the current in the new point ($k+1$). It then compares it with the current measured in $k$. A rise in the current will then change the duty cycle in the same direction as the previous change. A decrease in the current will change the duty cycle in the opposite direction of its previous change. The size of these changes is given by $C_p$. After the duty cycle is changed, a new current measurement is made, and compared to the previous in a similar manner. This repeats itself in a never-ending loop.

In figure 9.9 the duty cycle will move to the right, to point $k+1$ and keep heading into that direction until it measures a lower current (here at point $k+7$), it will then change direction. When the MPP is found, the duty cycle will oscillate back and forth between $k+5$, $k+6$, and $k+7$. This will continue, until the MPP moves, which typically would be caused by a change in illumination.

As mentioned $C_p$ determines the step between each position on the x-axis. If the step is big the system will respond fast but it will not be particularly precise. On the other hand, when the step is small the system will be slower, but more precise.

Figure 9.10 shows a flow chart of the MPPT algorithm as it will be used. It begins by setting or measuring a number of default start values, which are needed later in the algorithm. Both the start measurement of V(k-1), and the later measurement of V(k) are not used for the actual MPPT algorithm, but for the over-charge algorithm. The MPPT algorithm then measures the current I(k) and calculates the change in current and in duty cycle since the last measurement. The change in current ΔI(k) then decides the direction of the next step in the duty cycle, so that if the power (here measured as a current) has increased, the direction is unchanged, but if the power has decreased or not changed, the step direction is reversed. This is implemented in the ΔD(k)>0 at the bottom of the flowchart, where after the new duty cycle is calculated.

To avoid over-charging the battery an over-charge algorithm is inserted into our MPPT algorithm (see figure 9.10). This secondary algorithm measures the battery voltage and keeps it from rising above the maximum battery voltage of 8.4 V (see chapter D on page 185). Figure 9.11 shows a flowchart of the over-charge algorithm.

First the algorithm checks to see if the battery voltage V(k) is below 8.4 V. If this is not the case, the voltage difference ΔV(k) is calculated, and the variable OCP is set to one. This is done to show that the over-charge regulation is being initiated. The direction of ΔV(k) is then used to determine the change in the maximum duty cycle ΔMaxD(k). For a negative ΔV(k) the maximum duty cycle MaxD(k)is unchanged, whereas for non negative ΔV(k) the maximum duty cycle is halved. From this point on the following in the flowchart is the implementation of limiting the duty cycle to the maximum duty cycle. Is V(k) bellow 8.4 V, it is checked whether the over-charge regulation was used during the last cycle. If so the duty cycle is set to its default value of 0.4. If not the duty cycle is left unchanged. In both cases OCP is set to 0. The reason for choosing

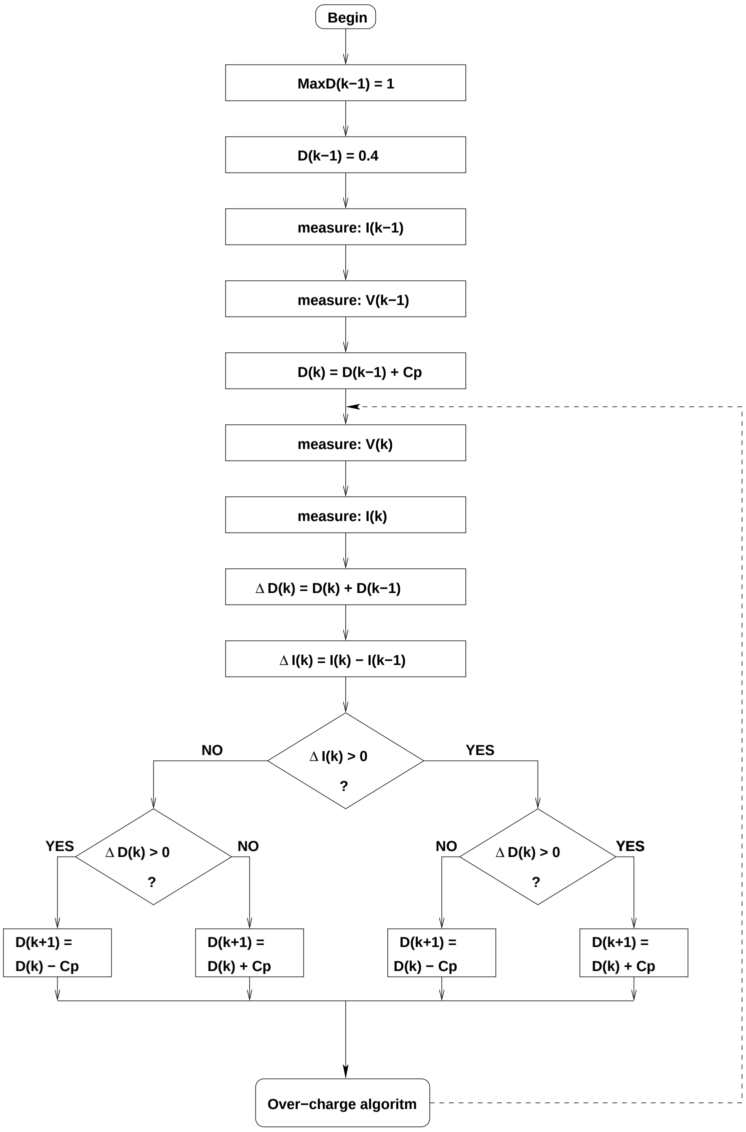


**Figure 9.10:** Flow chart of the P&O algorithm

over-charge algorithm is that it is very simple to implement in the software.

**Calculation of Interval and Frequency**

To find the interval ($C_p$) between the measurement points and how often the MPPT loop is to be carried out, certain considerations have to be taken into account. Since the MPPT algorithm expects a stable current input, it can not make a new measurement, before the MPPTC is stabilized. Therefore, the time between regulations must be larger than the settling time of the MPPTC. Using equation 9.30 the response on the output current for a 0.01 step on the duty cycle can be found. The step response can be seen at figure 9.12.

As can be seen from the figure 9.12, the settling time of the MPPTC is $<$ 10 ms. This means that the control frequency is maximum $\frac{1}{10ms} = 100Hz$. To be sure that the MPPT can follow the MPP when this moves, a minimum step size has to be computed. According to [930 and 931, 2001] the satellite will turn with a max speed of 3 rounds per minute. The voltage MPP is estimated to move between 3.4 to 4.2V during one maximum/minimum cycle. During this time the maximum and minimum duty cycle is expected to be 0.524 and 0.233 (see section 6.3.1 on page 45) There will be 4 of these cycles for each $360^o$ turn. With a frequency of 100 Hz, the minimum step size $C_p$ can be found as:

$$C_p \cdot 100 > \frac{4 \cdot 3}{60} \cdot (0.595 - 0.3) = 5.9 \cdot 10^{-4}$$

This means that $C_p$ has to be larger than $5.9 \cdot 10^{-4}$. To keep a safety margin, the step size $C_p$ is chosen to $1 \cdot 10^{-3}$.

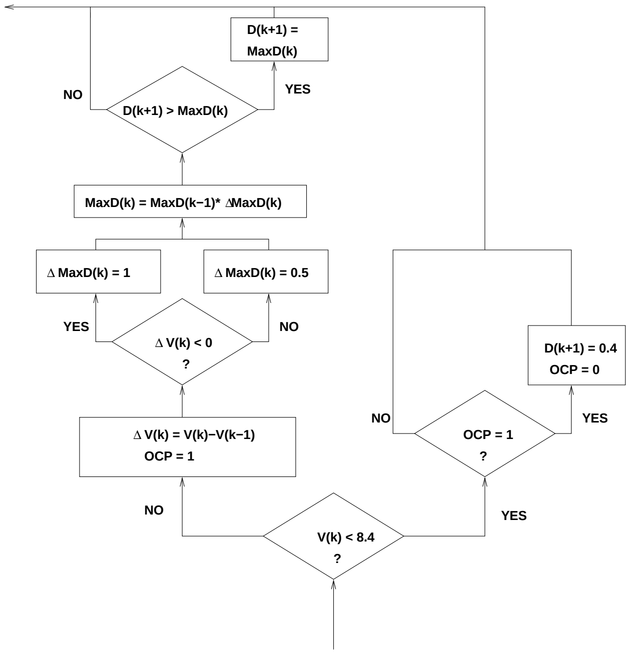


**Figure 9.11:** Flow chart of the over-charge algorithm

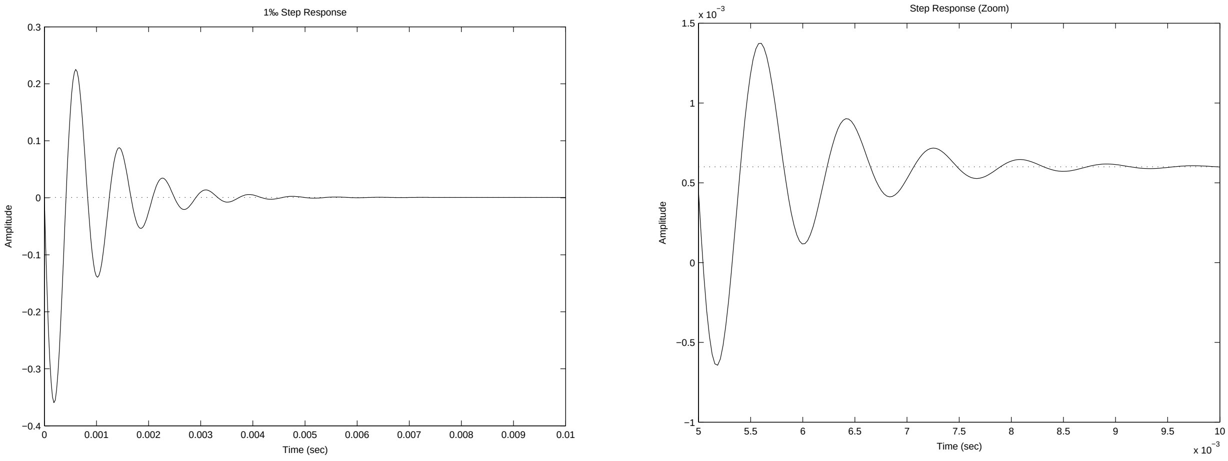


**Figure 9.12:** The step response of the MPPTC, with zoom to the right

As described in section 9.3 the MPPTC is designed for 2% output voltage ripple and therefore also has a 2% current ripple is expected. This ripple comes from the switching of the converter, and will therefore be at the switching frequency of 50 kHz (see chapter 6 on page 41) or higher. To avoid this ripple from affecting our DC power/current measurement, the ripple on these measurements must be dampened. Caused by the lack of data on our solar panels, it cannot be said exactly how much a ripple of a given size will effect the running of the solar panels, but a good guess is that if the 2 % ripple is dampened by approximately 40 dB, this effect will not have to be taken into account. Therefore a dampening of 40 dB is needed at 50 kHz. This filtering will be made by the measurement circuitry (see section 10.3 on page 111). In table 9.1 the key values for the MPPT can be found.

| | |
|---|---|
| Step size ($C_p$) | 0.001 |
| Samplings frequency | 100Hz |
| Filter corner frequency | 500Hz |

**Table 9.1:** MPPT key values

### 9.5.2 Controllers for PCC

It is a requirement for the PCC to keep a zero steady state error on the output as well as a small overshot of only 5% and the settling time must be below 20 ms (requirements derived from chapter 3 on page 23).

For the inner loop it is important to keep small rise times and settling times without getting too large a bandwidth in order to keep the converter well controlled and reducing the effect of sudden transients in the outer loop. For both loops the overshoot should be kept below 5%.

The requirement for a zero steady state error will be taken care of in the outer loop, but to ease the task the steady state error in the inner loop will be kept below 20%.The requirements for the inner and outer loop are specified in table 9.2

| | | |
|---|---|---|
| Inner loop | Maximum overshoot $M_p$ | 5% |
| Inner loop | Maximum settling time $t_{s1\%}$ | 10 ms |
| Inner loop | Maximum steady state error $e_{ss}$ | 20% |
| Outer loop | Maximum overshoot $M_p$ | 5% |
| Outer loop | Maximum settling time $t_{s1\%}$ | 20 ms |
| Outer loop | Maximum steady state error $e_{ss}$ | 0 |

**Table 9.2:** PCC controller requirements

**Controller for Inner Loop**

First the uncompensated open loop of the inner loop, which is equal to equation 9.48, is determined using the values for the PCC found in chapter 7 on page 55. For the intervals, mean values are chosen for operating values, which is done, because the controller must be able to operate in the complete range of the system. However the controller must be tested in the complete spectrum of the converter to ensure a full operating range. The operating point values can be found in table 9.3.

| $R$ | $D$ | $V_g$ | $L$ | $C$ |
|---|---|---|---|---|
| $17\Omega$ | $0.7$ | $7.2V$ | $925\mu H$ | $68\mu F$ |

**Table 9.3:** PCC operating point values

The values are then inserted into the uncompensated open loop transfer function which yields:

$$\begin{aligned} G_{oi}(s) &= \frac{sCV_g + \frac{V_g}{R}}{s^2LC + \frac{L}{R}s + 1} \Rightarrow \\ G_{oi}(s) &= \frac{s4.90 \cdot 10^{-4} + 0.423}{s^2 6.29 \cdot 10^{-8} + 5.4 \cdot 10^{-5}s + 1} \end{aligned} \tag{9.50}$$

The function is the plotted (see figure 9.13) and it can be observed that the cross-over frequency is placed around 6600 rad/s (1.05 kHz). If only P compensation is used on the system the bandwidth will rise to a very large value and therefore different types of compensation must be considered. Because of the allowed steady state error an PI term, i.e. the integrator, in the controller is not a necessity, but it can still be a good idea to apply one in this case, because of its ability to decrease the bandwidth [Gene F. Franklin and Emami-Naeini, 1994] page 183. The task is then to apply a PI compensation to the system and thereby reduce the bandwidth and steady state error, without decreasing the speed of the system too much.

The PI compensator which is described by equation 9.3 is then applied by first adding a integrator (1/s) to the system yielding a DC gain of infinite. The phase lifting zero is then placed at 4000 rad/s (637 Hz), equal to a $T_i$ of $15.7 \cdot 10^{-4}$ s, and it is then possible to determine a suitable gain factor for the system. This gain factor was chosen to 1 yielding the response seen at figure 9.14. The phase margin can now be read to about 62°at the open loop bode plot. The compensated open loop that was found is:

$$\begin{aligned} G_{oi}(s) &= D_i(s)G_i(s) = \Big(K_i \frac{s + \frac{1}{T_i}}{s}\Big)\Big(\frac{sCV_g + \frac{V_g}{R}}{s^2LC + \frac{L}{R}s + 1}\Big) \Leftrightarrow \\ G_{oi}(s) &= \frac{s^2K_iCV_g + s(\frac{K_iCV_g}{T_i} + \frac{K_iV_g}{R}) + \frac{K_iV_g}{T_iR}}{s^3LC + \frac{L}{R}s^2 + s} \Rightarrow \end{aligned}$$

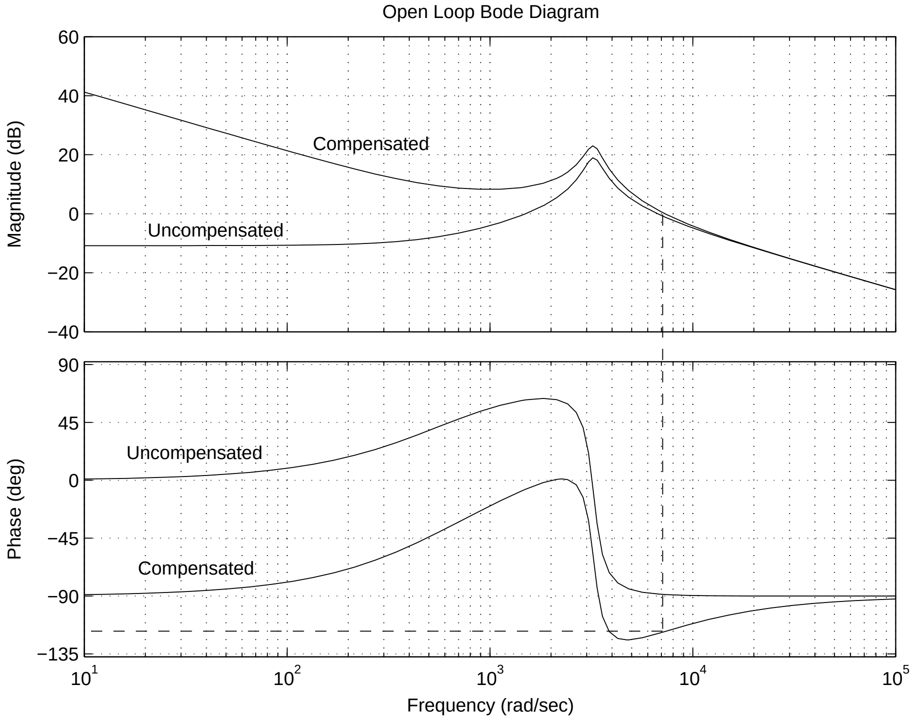


**Figure 9.13:** The open loop bode plot of the PCC inner loop with the phase margin marked by a dashed line

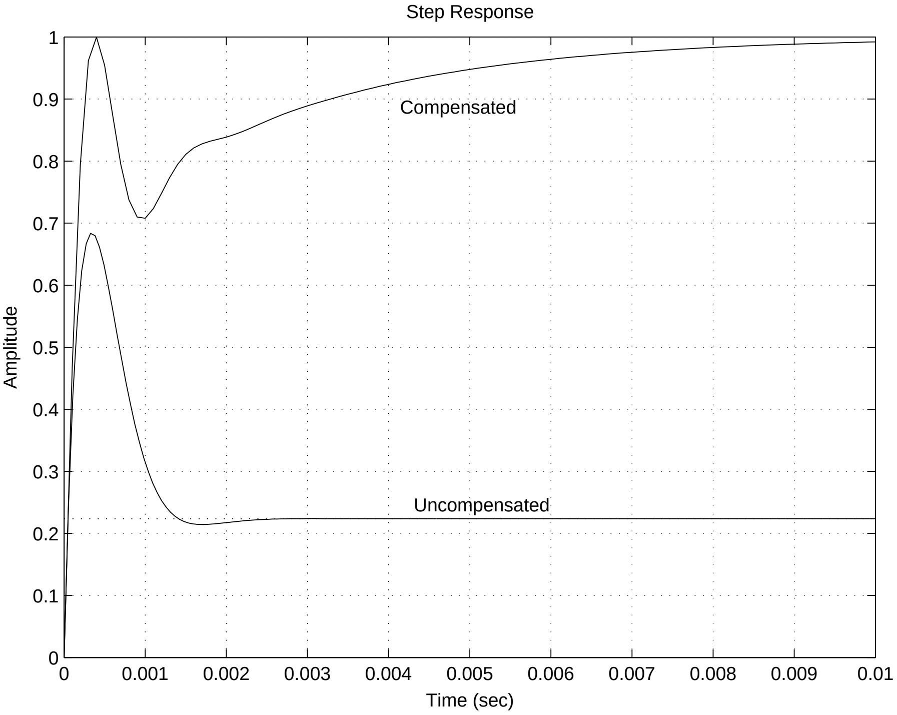


**Figure 9.14:** The step response of the PCC inner loop

$$G_{oi}(s) \quad = \quad \frac{s^2 4.9 \cdot 10^{-4} + s2.82 + 2075}{s^3 6.29 \cdot 10^{-8} + s^2 5.44 \cdot 10^{-5} + s} \tag{9.51}$$

The compensated closed loop that was found is:

$$T_i(s) \quad = \quad \frac{G_{oi}(s)}{1 + G_{oi}(s)} = \frac{\frac{s^2 K_i C V_g + s(\frac{K_i C V_g}{T_i} + \frac{K_i V_g}{R}) + \frac{K_i V_g}{T_i R}}{s^3 LC + \frac{L}{R} s^2 + s}}{1 + \frac{s^2 K_i C V_g + s(\frac{K_i C V_g}{T_i} + \frac{K_i V_g}{R}) + \frac{K_i V_g}{T_i R}}{s^3 LC + \frac{L}{R} s^2 + s}} \Leftrightarrow$$

$$
\begin{aligned}
T_i(s) &= \frac{s^2V_gCK_i + s(\frac{V_gK_i}{R} + \frac{CV_gK_i}{T_i}) + \frac{V_gK_i}{T_iR}}{s^3LC + s^2(\frac{L}{R} + K_iV_gC) + s(\frac{V_gK_i}{R} + \frac{K_iCV_G}{T_i} + 1) + \frac{V_gK_i}{RT_i}} \Rightarrow \\
T_i(s) &= \frac{s^2 4.9\cdot 10^{-4} + s2.82 + 2075}{s^3 6.29\cdot 10^{-8} + s^2 5.4\cdot 10^{-4} + s3.82 + 2075}
\end{aligned}
\tag{9.52}
$$

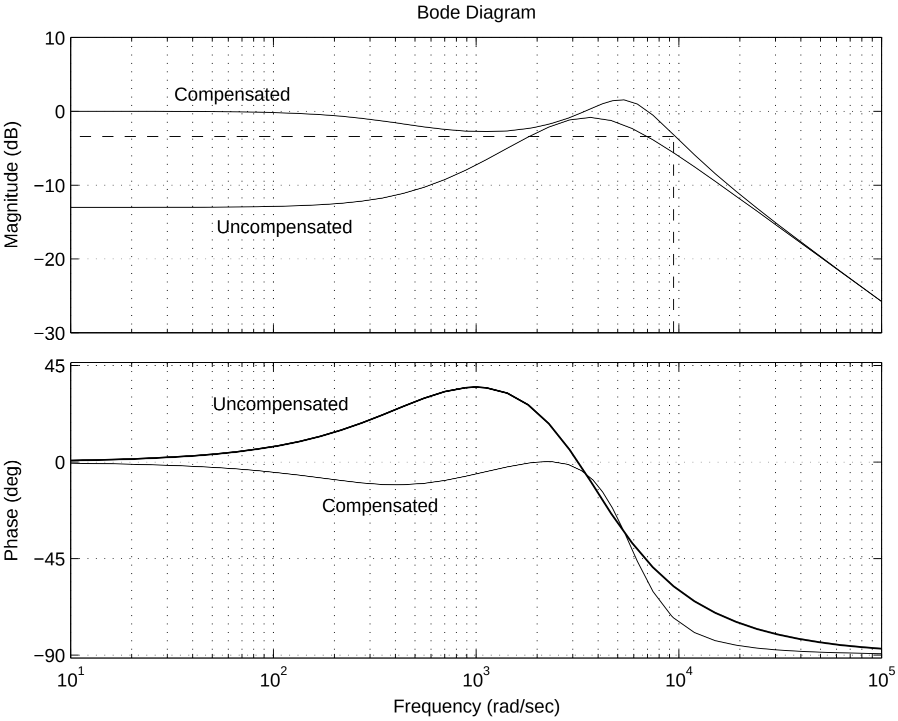


**Figure 9.15:** The closed loop bode plot of the PCC inner loop with the bandwidth marked by a dashed line

On figure 9.15 the closed loop bode plot for the inner loop is shown and the bandwidth of the system can be read to about 9300 rad/s (1.5 kHz). In conclusion this yields a system that has an adequate phase margin, a relative small bandwidth and a zero steady state error. It can be seen that there is no overshot and that the settling time is below 10 ms and thus all requirements for the inner loop are fulfilled with the following compensation used:

$$
D_i(s) = 1\frac{s+4000}{s}
\tag{9.53}
$$

The summarized results for the inner loop compensator are shown in table 9.4.

| $K_i$ | $T_i$ | BW | $M_p$ | $T_r$ | $T_s$ | e$_{ss}$ |
|---|---|---|---|---|---|---|
| 1 | 2.5$\cdot 10^{-4}$ $s$ | 1.5kHz | 0% | 0.25 ms | 9 ms | 0 |

**Table 9.4:** The results for the inner loop compensator

### Controller for outer loop

The uncompensated outer open loop is:

$$
\begin{aligned}
G_{oo}(s) &= T_i(s)G_o(s) \Leftrightarrow \\
G_{oo}(s) &= \Big(\frac{s^2V_gCK_i + s(\frac{V_gK_i}{R} + \frac{CV_gK_i}{T_i}) + \frac{V_gK_i}{T_iR}}{s^3LC + s^2(\frac{L}{R} + K_iV_gC) + s(\frac{V_gK_i}{R} + \frac{K_i}{T_i}CVg + 1) + \frac{V_gK_i}{RT_i}}\Big)\Big(\frac{1}{sC + \frac{1}{R}}\Big) \Leftrightarrow \\
G_{oo}(s) &= \frac{s^2V_gCK_i + s(\frac{V_gK_i}{R} + \frac{CV_gK_i}{T_i}) + \frac{V_gK_i}{T_iR}}{a1s^4 + a2s^3 + a3s^2 + a4s + a5} \Rightarrow \\
G_{oo}(s) &= \frac{s^2 4.9\cdot 10^{-4} + s2.82 + 2075}{s^4 4.27\cdot 10^{-12} + s^3 4.07\cdot 10^{-8} + s^2 2.91\cdot 10^{-4} + s0.37 + 122.1}
\end{aligned}
\tag{9.54}
$$

Where:
$a1 = LC^2$
$a2 = \frac{LC}{R} + K_iV_gC^2 + \frac{LC}{R}$
$a3 = \frac{CV_gK_i}{R} + \frac{CK_i}{T_i}CVg + C + \frac{L}{R^2} + \frac{K_iV_gC}{R}$
$a4 = \frac{V_gK_i}{R^2} + \frac{K_i}{RT_i}CVg + \frac{1}{R} + \frac{CV_gK_i}{T_iR}$
$a5 = \frac{V_gK_i}{T_iR^2}$

As it can be seen on the open loop equation 9.54 the steady state error is non-zero when using the Final Value Theorem ($s \rightarrow 0$) and therefore a PI compensation is also needed for the outer loop. The open loop can be seen plotted at figure 9.16.

As with the inner loop a integrator is added to the system and also a phase lifting zero is applied at 500 rad/s (80Hz), which yields a $T_o$ of 0.0124 s. The system has now become quite underdampened and therefore a low $K_o$ of 0.1 is chosen to dampen the system and also decreases the bandwidth. On figure 9.16 the new phase margin can be read to about 90 °. This yields a compensated open loop of:

$$\begin{aligned}
G_{oo}(s) &= D_o(s)T_i(s)G_o(s) \Rightarrow \\
G_{oo}(s) &= \Big(K_o\frac{s+\frac{1}{T_o}}{s}\Big)\Big(\frac{s^2V_gCK_i + s(\frac{V_gK_i}{R} + \frac{CV_gK_i}{T_i}) + \frac{V_gK_i}{T_iR}}{a1s^4 + a2s^3 + a3s^2 + a4s + a5}\Big) \Leftrightarrow \\
G_{oo}(s) &= \frac{K_iK_oV_g(s^3C + s^2(\frac{1}{R} + \frac{C}{T_i} + \frac{C}{T_o}) + s(\frac{1}{RT_o} + \frac{C}{T_iT_o} + \frac{1}{T_iR}) + \frac{1}{T_iT_oR})}{a1s^5 + a2s^4 + a3s^3 + a4s^2 + a5s} \Rightarrow \\
G_{oo}(s) &= \frac{s^34.90\cdot10^{-5} + s^20.31 + s349 + 1.04\cdot10^5}{s^54.27\cdot10^{-12} + s^44.07\cdot10^{-8} + s^32.91\cdot10^{-4} + s^20.37 + s122.1} \qquad (9.55)
\end{aligned}$$

The compensated closed loop can then be calculated:

$$\begin{aligned}
T_o(s) &= \frac{G_{oo}}{1+G_{oo}} \Leftrightarrow \\
T_o(s) &= \frac{\frac{K_iK_oV_g(s^3C+s^2(\frac{1}{R}+\frac{C}{T_i}+\frac{C}{T_o})+s(\frac{1}{RT_o}+\frac{C}{T_iT_o}+\frac{1}{T_iR})+\frac{1}{T_iT_oR})}{a1s^5+a2s^4+a3s^3+a4s^2+a5s}}{1+\frac{K_iK_oV_g(s^3C+s^2(\frac{1}{R}+\frac{C}{T_i}+\frac{C}{T_o})+s(\frac{1}{RT_o}+\frac{C}{T_iT_o}+\frac{1}{T_iR})+\frac{1}{T_iT_oR})}{a1s^5+a2s^4+a3s^3+a4s^2+a5s}} \Leftrightarrow \\
T_o(s) &= \frac{K_iK_oV_g(s^3C + s^2(\frac{1}{R} + \frac{C}{T_i} + \frac{C}{T_o}) + s(\frac{1}{RT_o} + \frac{C}{T_iT_o} + \frac{1}{T_iR}) + \frac{1}{T_iT_oR})}{b1s^5 + b2s^4 + b3s^3 + b4s^2 + b5s + b6} \Rightarrow \\
T_o(s) &= \frac{s^34.90\cdot10^{-5} + s^20.31 + s349 + 1.04\cdot10^5}{s^54.27\cdot10^{-12} + s^44.07\cdot10^{-8} + s^33.41\cdot10^{-4} + s^20.67 + s471 + 103800} \qquad (9.56)
\end{aligned}$$

Where:
$b1 = LC^2$
$b2 = \frac{LC}{R} + K_iV_gC^2 + \frac{LC}{R}$
$b3 = \frac{CV_gK_i}{R} + \frac{CK_i}{T_i}CVg + C + \frac{L}{R^2} + \frac{K_iV_gC}{R} + K_iK_oV_gC$
$b4 = \frac{V_gK_i}{R^2} + \frac{K_i}{RT_i}CVg + \frac{1}{R} + \frac{CV_gK_i}{T_iR} + \frac{K_iK_oV_g}{R} + \frac{K_iK_oV_gC}{T_i} + \frac{K_iK_oV_gC}{T_o}$
$b5 = \frac{V_gK_i}{T_iR^2} + \frac{K_iK_oV_g}{RT_o} + \frac{K_iK_oV_gC}{T_iT_o} + \frac{K_iK_oV_g}{T_iR}$
$b6 = \frac{K_iK_oV_g}{T_iT_oR}$

The step response can be seen at figure 9.17 and it is clear that the compensated response is well within the requirements with at overshot of 0%, a settling time of less than 10 ms and no steady state error.

At figure 9.18 the bandwidth of the compensated system can be read to about 500 rad/s (80 Hz) and it can be concluded that the system fulfills the requirements with the following compensation used:

$$D_i(s) = 0.06\frac{s+500}{s} \qquad (9.57)$$

The results are summarized in table 9.5.

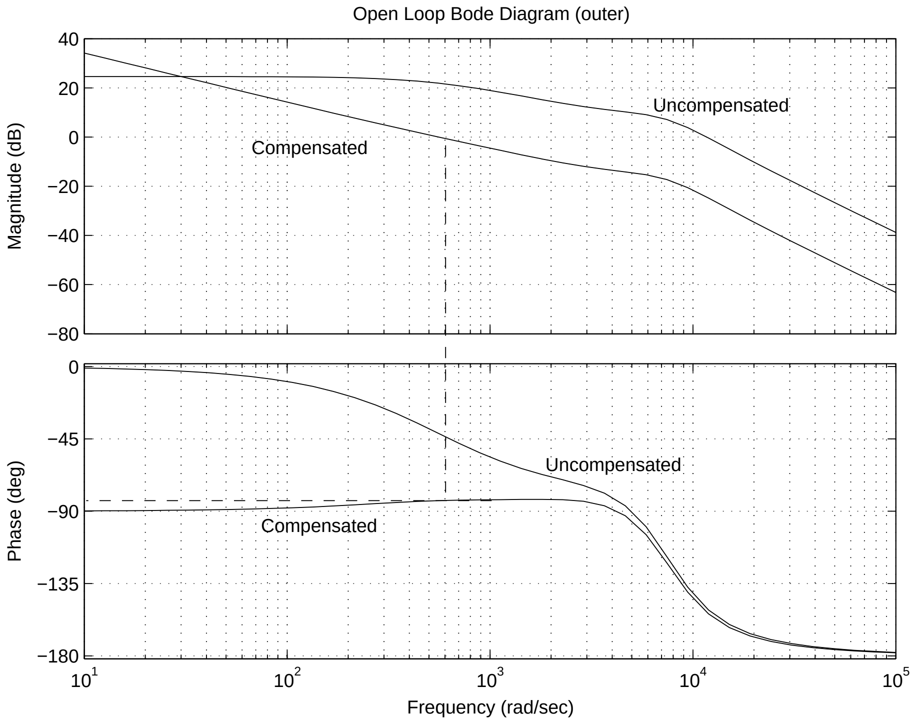


**Figure 9.16:** The open loop bode plot of the PCC outer loop with the phase margin marked by a dashed line

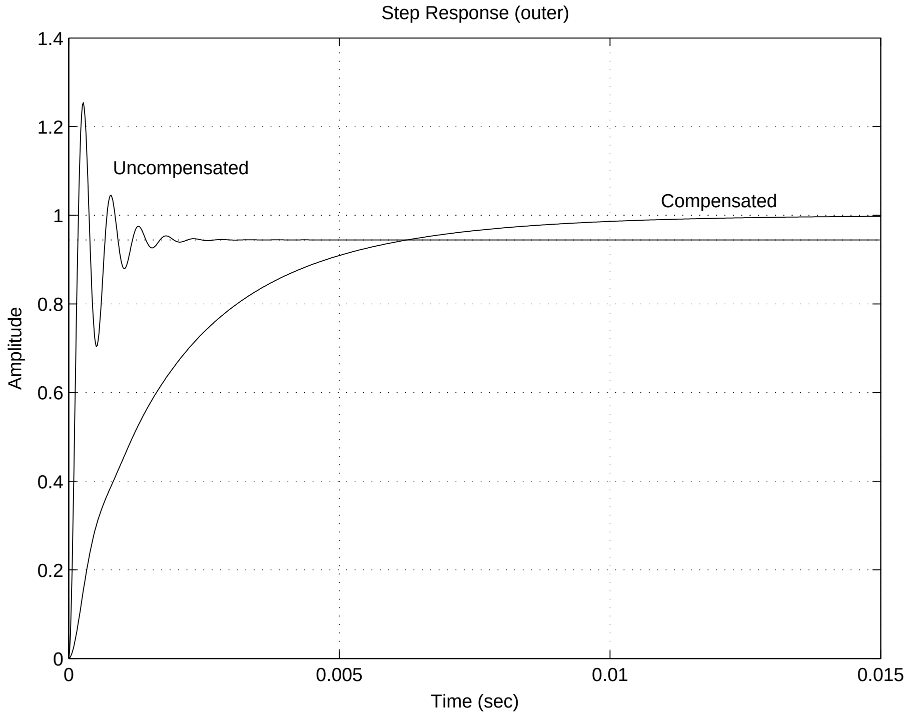


**Figure 9.17:** The step response of the PCC outer loop

| $K_i$ | $T_i$ | BW | $M_p$ | $T_r$ | $T_s$ | e$_{ss}$ |
|---|---|---|---|---|---|---|
| 0.06 | $2{\cdot}10^{-3}$ | 80Hz | 0% | 4.5 ms | 9 ms | 0 |

**Table 9.5:** The results for the outer loop compensator

### Sampling frequency and Prefiltering

As mentioned in the beginning of this section the sampling frequency ($f_s$) must be from between a factor 2 and 40 of the bandwidth ($f_{bw}$) and at least a factor 10 is recommended [Gene F. Franklin and Emami-Naeini, 1994],

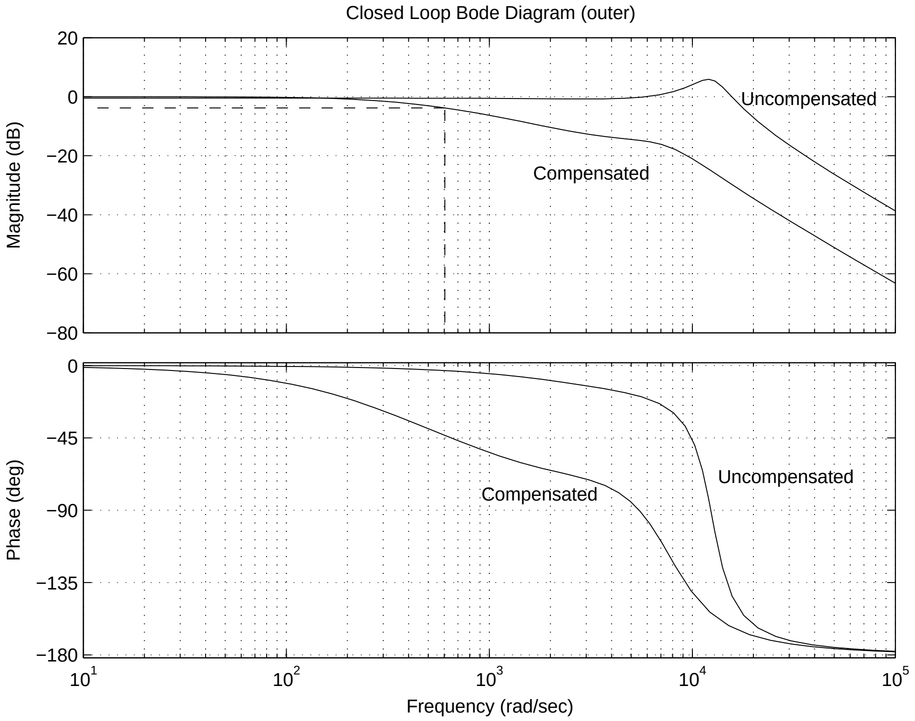


**Figure 9.18:** The closed loop bode plot of the PCC outer loop with the bandwidth marked by a dashed line

page 606:

$$10f_{bw} \leq f_s \leq 40f_{bw} \tag{9.58}$$

The relationship between the sampling frequency and the corner frequency of the prefilter ($f_c$) is [Haugen, 1994] page 569:

$$f_c = \frac{1}{4}f_s \tag{9.59}$$

From equation 9.58 and 9.59 the sampling frequency and prefilter corner frequency of the inner and outer loop can now be determined as shown in table 9.6:

| | Inner Loop | Outer Loop |
|---|---|---|
| Bandwidth | 1.5 kHz | 80 Hz |
| Factor | 12 | 20 |
| Samplings frequency | 18 kHz | 1.6 kHz |
| Filter corner frequency | 4.5 kHz | 400 Hz |

**Table 9.6:** PCC sampling and filter requirements

The design is then tested for the effect of the filters on the system and on figure 9.19 the step responses for both loops with the filter can be seen. Both responses still look fine, but the overshoot on the inner loop have ascended to about 15% and the gain factor ($K_i$) is then lowered to 0.6, bringing the overshot down to 5% again. Also the gain of the outer loop are lowered a bit to 0.05.

## 9.6 Digitization of the controllers

The transfer function for the continuous systems must be digitized to enable the MCU to perform the control. The transfer functions are transformed from the continuous s-plane to the discrete z-plane, with the following transformation:

$$D(s) = Z\{D(s)\} \tag{9.60}$$

The digitization method used in the following is know as the Matched Pole Zero (MPZ) method which is described by the following three points [Gene F. Franklin and Emami-Naeini, 1994] page 620:

1. Map poles and zeros using the relation $z = e^{sT}$, where $T$ is the samplings period.

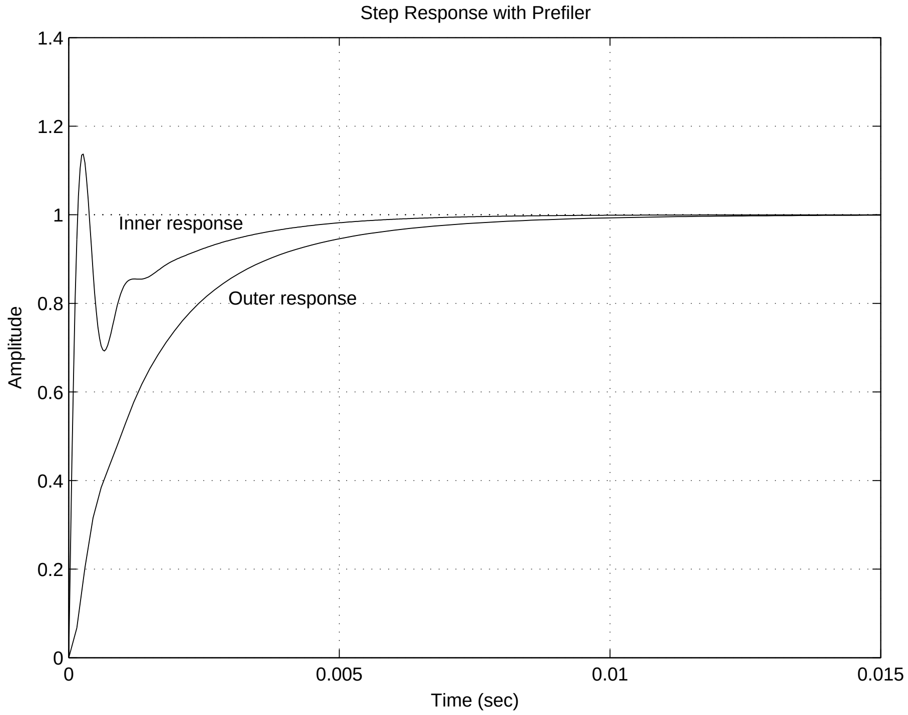


**Figure 9.19:** The step responses for the inner and outer loop with filter added

2. Multiply $(1+z^{-1})$ to the numerator until numerator and denumerator are of equal order
3. Set DC gain of D(z) equal to that of D(s) by using the discrete Final Value Theorem.

### 9.6.1 Digitization of the MPPTC controller

The MPPT algorithm can be considered to be a row of logical choices and therefore it does not need further digitization to be implemented in the MCU.

### 9.6.2 Digitization of the PCC controller

The control for the PCC has been chosen to be two PI-controller described by equation 9.53 and 9.57 (remembering the changes made after adding the filters), which followingly will be z-transformed using the MPZ method described above. First the inner loop with a samplings frequency of 18kHz ($T = 5.56 \cdot 10^{-5}$) is taken through step 1 and 2:

$$\begin{aligned}
D_i(s) &= 0.6\frac{s+4000}{s} \rightarrow \\
D_i(z) &= 0.6\frac{z-e^{-4000 \cdot 5.56 \cdot 10^{-5}}}{z-1} \Rightarrow \\
D_i(z) &= 0.6\frac{z-0.80}{z-1} \Rightarrow \\
D_i(z) &= 0.6\frac{1-0.80z^{-1}}{1-z^{-1}} \qquad (9.61)
\end{aligned}$$

The 3. step is then completed by using the discrete Final Value Theorem and letting $z \rightarrow 1$ in the above equation and it can be seen that the discrete DC gain is infinite like the continuous DC gain and thus no adjustment is needed. The same is done for the outer loop ($T = 6.25 \cdot 10^{-4}$) yielding the following:

$$\begin{aligned}
D_o(s) &= 0.05\frac{s+500}{s} \rightarrow \\
D_o(z) &= 0.05\frac{z-e^{-500 \cdot 6.25 \cdot 10^{-4}}}{z-1} \Rightarrow \\
D_o(z) &= 0.05\frac{z-0.73}{z-1} \Rightarrow
\end{aligned}$$

$$D_o(z) \quad = \quad 0.05\frac{1-0.73z^{-1}}{1-z^{-1}} \tag{9.62}$$

Again no DC gain adjustment is required. At figure 9.20 the discrete step response for both inner and outer loop can be seen and if it is compared to the continuous step responses at figure 9.14 and 9.17 it can be seen that there are no major differences in the responses.

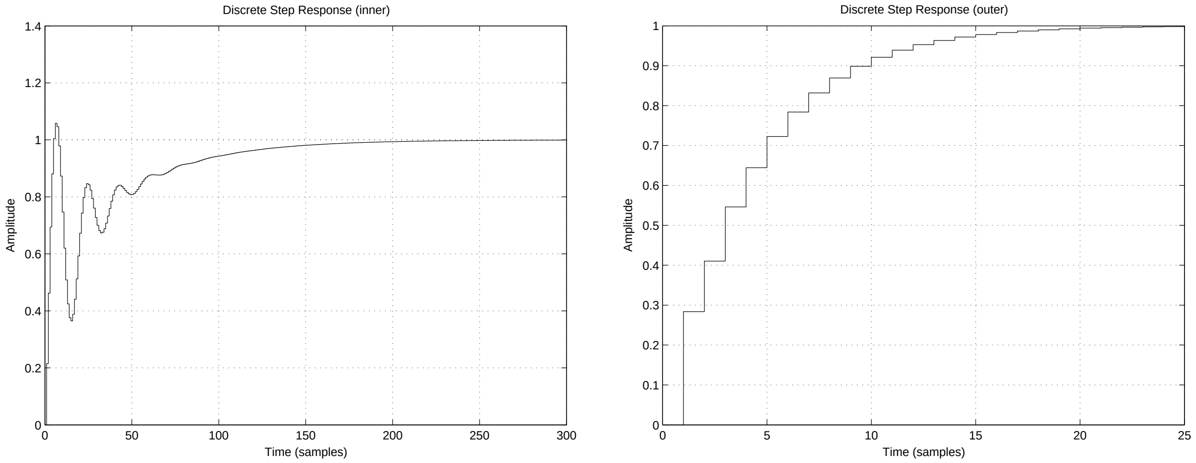


**Figure 9.20:** The discrete step response for the inner and outer loop

## 9.7 Simulation of the Controller Design

To test the controllers simulations are carried out on the Simulink models created in chapter 6 on page 41 and 7 on page 55. The simulation models can be found in appendix K on page 223 and on the enclosed CDROM[2] for both the MPPTC and PCC controller.

### 9.7.1 Simulation of MPPTC Controller

In order to simulate the MPPTC controller a model of the solar array and the batteries are needed. The batteries are simulated by a voltage source of 7.2V on the output which gives a constant output voltage. The solar array are modeled in form of a second order equation which, even though it only roughly describes the solar array, can be used for simulation purposes.

$$I = -0.041V^2 + 0.85 \tag{9.63}$$

On figure 9.21 the I(V) and P(V) plot for this equation can be seen.

The MPP can then be calculated from equation 9.63 and in table 9.7 are the data, which the simulation should find, shown.

| $V_{MPP}$ | $I_{MPP}$ | $P_{MPP}$ | Duty |
|---|---|---|---|
| 2.629V | 0.566A | 1.490W | 0.63 |

**Table 9.7:** The MPPT data for the solar cell model

On figure 9.22 and 9.23 the result of the simulation are shown with respectively the power/duty cycle and the voltage/current.

As it can be seen the MPPT steps the duty cycle between 0.667 and 0.669, which result in the power shifting between 1.486 and 1.489. The power is very close to the calculated which indicates a good efficiency of the MPPT, but the duty cycle is a bit higher which is due to the losses in the MPPTC. Also the current and voltage are very close to the calculated values.

On figure 9.24 a simulation with a change in the solar cell equation and it can be seen that the MPPT quickly finds a new operating point. At figure 9.25 a simulation where the MPPT start with a duty cycle of 0.57 and then tracks the MPP to a duty cycle of about 0.67.

[2] CDROM/Matlab/Simulink

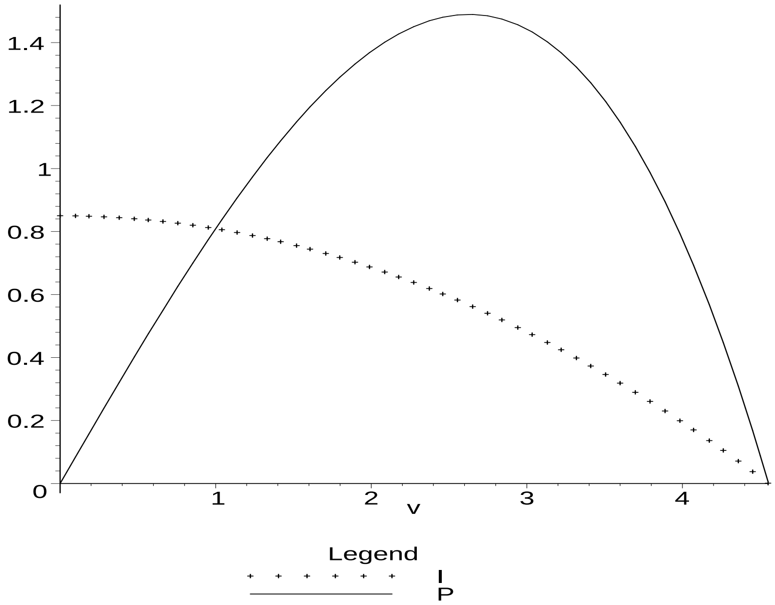


**Figure 9.21:** The I(V) and P(V) plot of the solar cells model

### 9.7.2 Simulation of PCC controller

The simulations are to be carried out using prefiltering and the discrete compensations. First a startup of the converter is simulated with 7.2V on the input and 25Ω as load. Secondly a load change is tested to simulate a subsystem switching in and out. A realistic scenario is that the converter is operating at nominal load (17Ω = 0.3A) and then the TRD is switched on and after a while it is switched off again, a change to 3Ω and back. Another scenario is that the converter is operating at nominal load and then all loads except the OBC are switched out and in again. The last thing that is simulated is how the controller acts when the converter operates in DCM. The simulation scenarios are shown in table 9.8.

| Description | Dynamic Changes | Graphs |
|---|---|---|
| Startup of the converter | None | figure 9.26 |
| Load change to simulate TRD switching in and out | R: 17Ω → 2Ω → 17Ω | figure 9.27 |
| Load change to simulate all loads switching out and in | R: 17Ω → 50Ω → 17Ω | figure 9.29 |
| Load change to test DCM | Vg: 20Ω → 100Ω → 20Ω | figure 9.30 |

**Table 9.8:** Simulation scenarios for the PCC

#### Startup of the converter

At figure 9.26 the simulated startup of the converter is depicted and it is clear that the control does not fulfill the requirement of a 5% overshot maximum. It is therefore recommended not to turn on the loads before the output voltage is stabilized during startup of the PSU.

#### Load Change 1

At figure 9.27 the simulated load change from 17Ω to 2Ω and back again is depicted and again it can be seen that the requirement of a 5% overshot maximum is not fulfilled. When the output current jumps from 0.3 A to 2.5 A a voltage overshot of 2 V occur and when the current jumps back again the overshot is 2.5 V. This problem however will be reduced in there real implementation because of the output filters that will prevent sudden changes on the regulated bus and thereby give the controller more time to adjust the converter.

In order to test whether the feed forward improves the system the same load change from 17 to 2 Ω was also made without the feed forward and the result can be seen at figure 9.28. It is clear that the feed forward improves the response of the system ”significantly”.

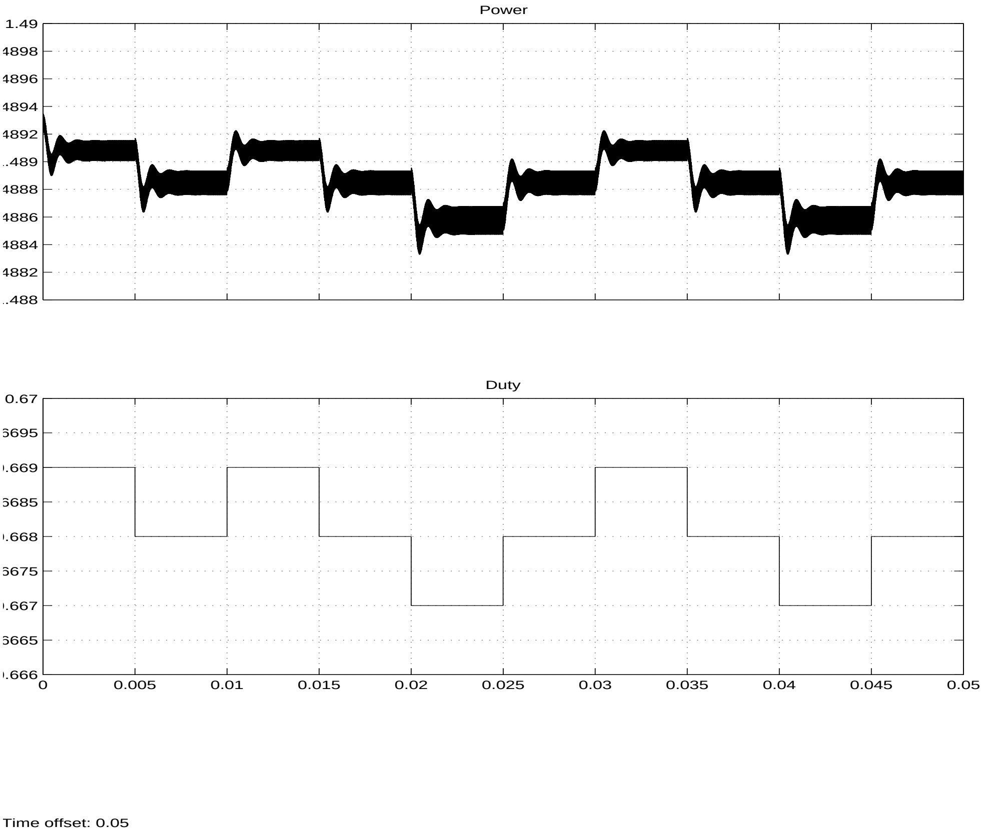


**Figure 9.22:** The result of the MPPT in form of power and duty cycle

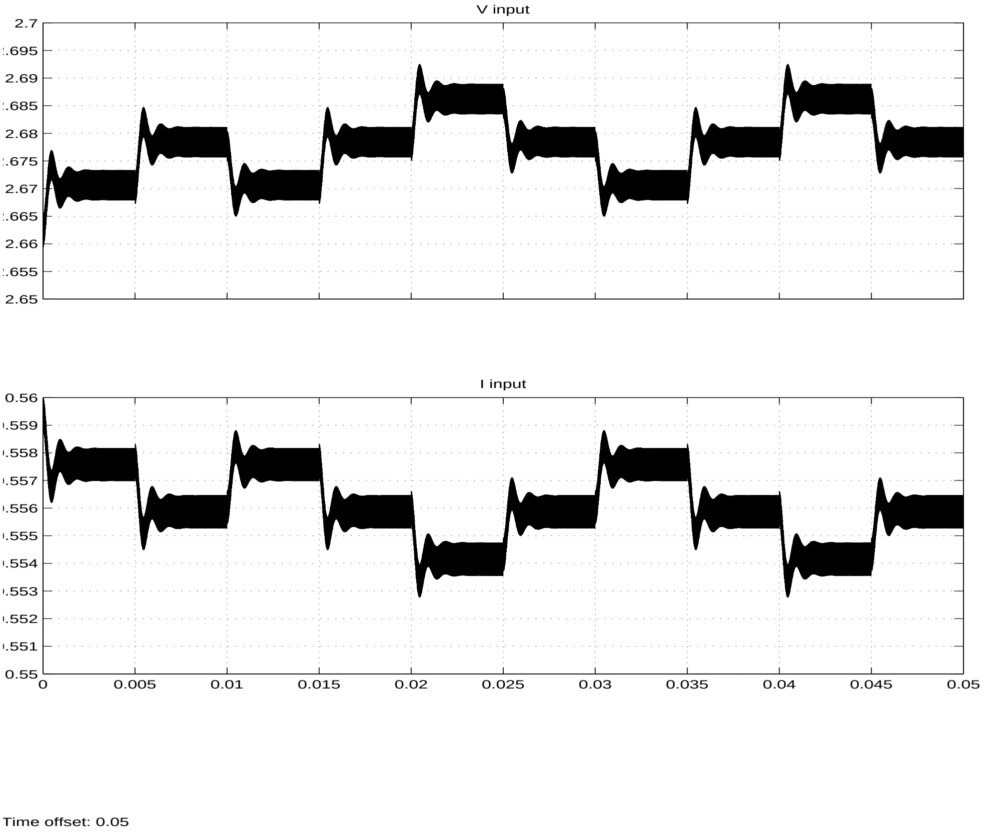


**Figure 9.23:** The result of the MPPT in form of voltage and current

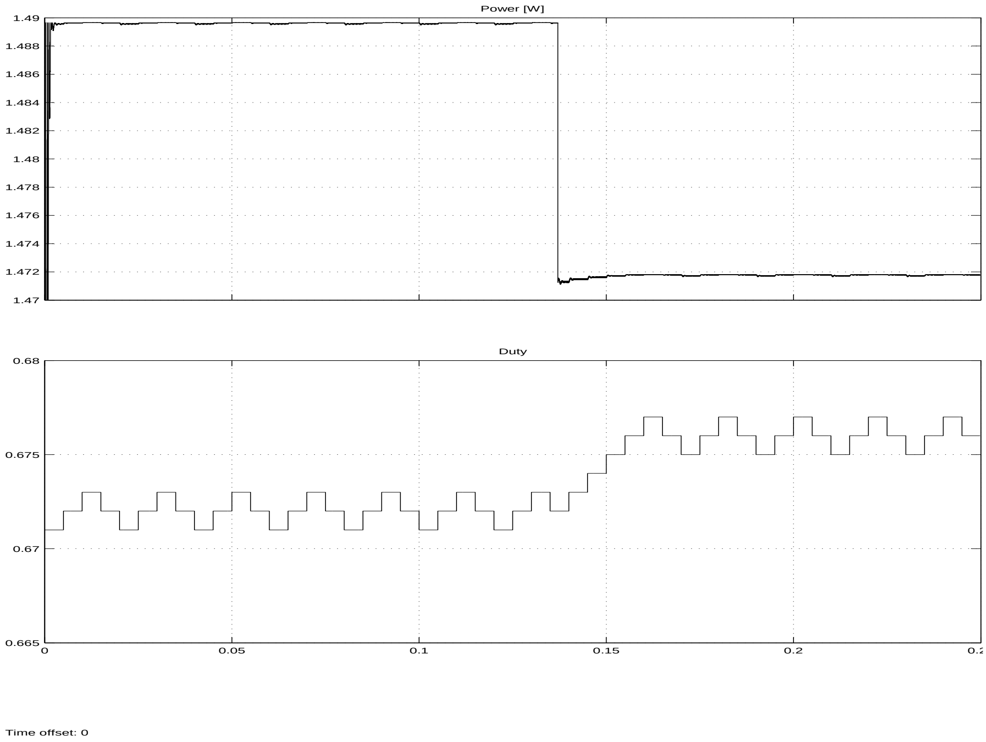


**Figure 9.24:** The result of the MPPT in form of power and duty cycle with a change in solar cell-equation

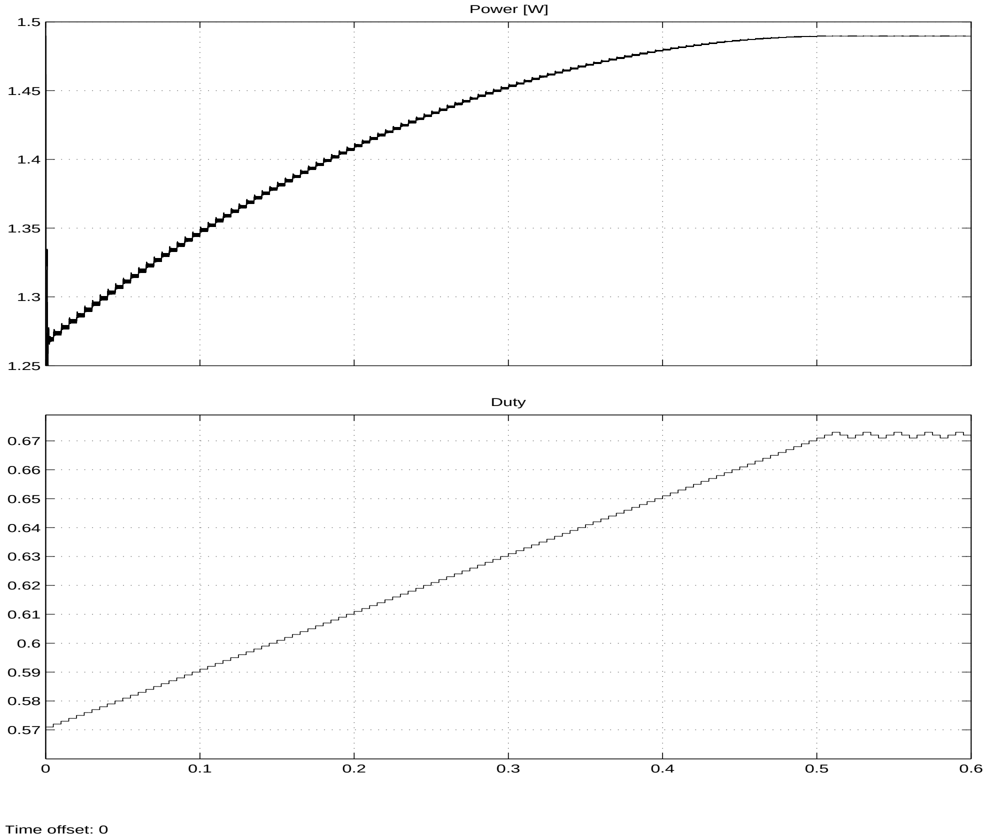


**Figure 9.25:** The result of the MPPT in form of power and duty cycle where the MPPT finds the maximum power point

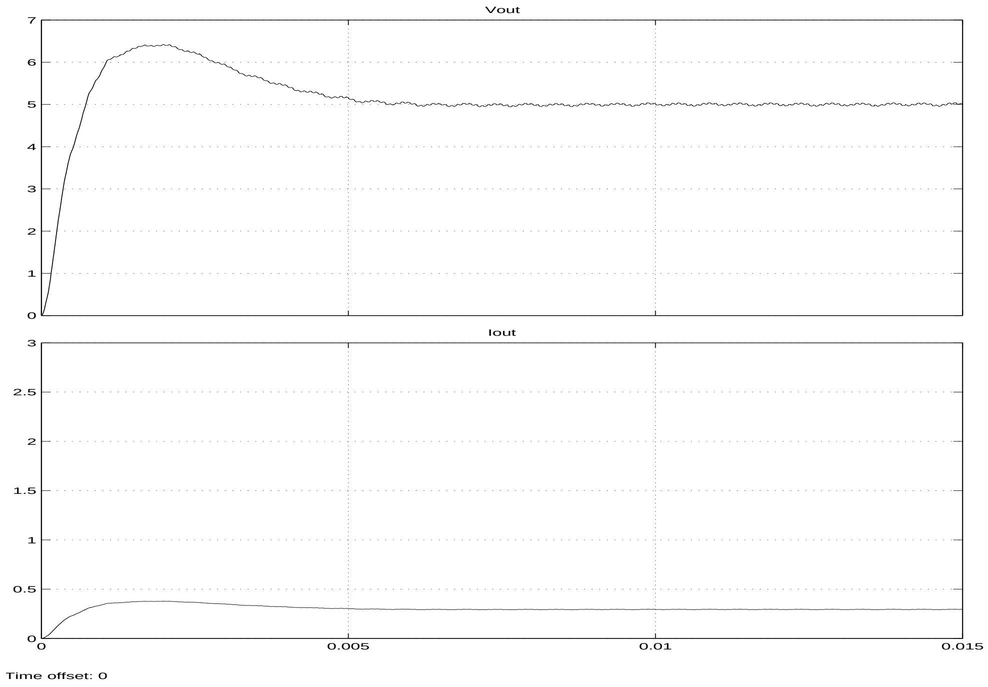


**Figure 9.26:** The output voltage and current during a startup

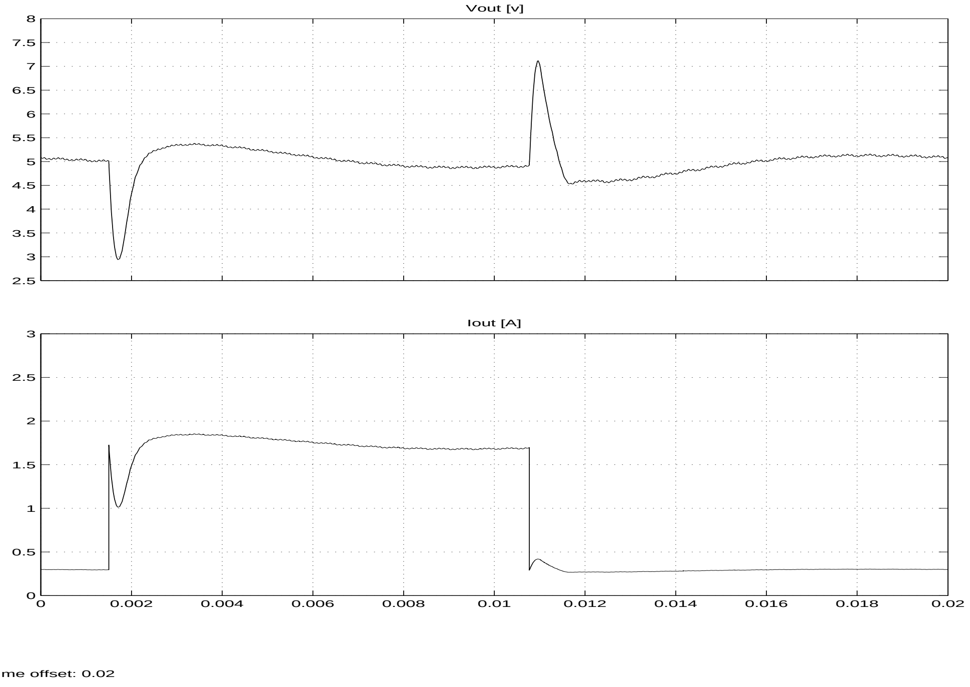


**Figure 9.27:** The output voltage and current during a load change from 17Ω to 2Ω and back

### Load Change 2

At figure 9.29 the simulated load change from 50Ω to 17Ω and back again is depicted and is can be seen that the changes in the output voltage are quite small and as described above they can be expected to be even smaller with the output filters added.

### Load Change 3

At figure 9.30 the simulated load change from 20Ω to 100Ω and back again is depicted and it is clear that the converter enters DCM when operating with a load resistance of 100Ω. It can be seen that the controller keeps the voltage ripple within the requirements, even without output filter and that it in spite of the converter operation in DCM managed to keep the output voltage stable.

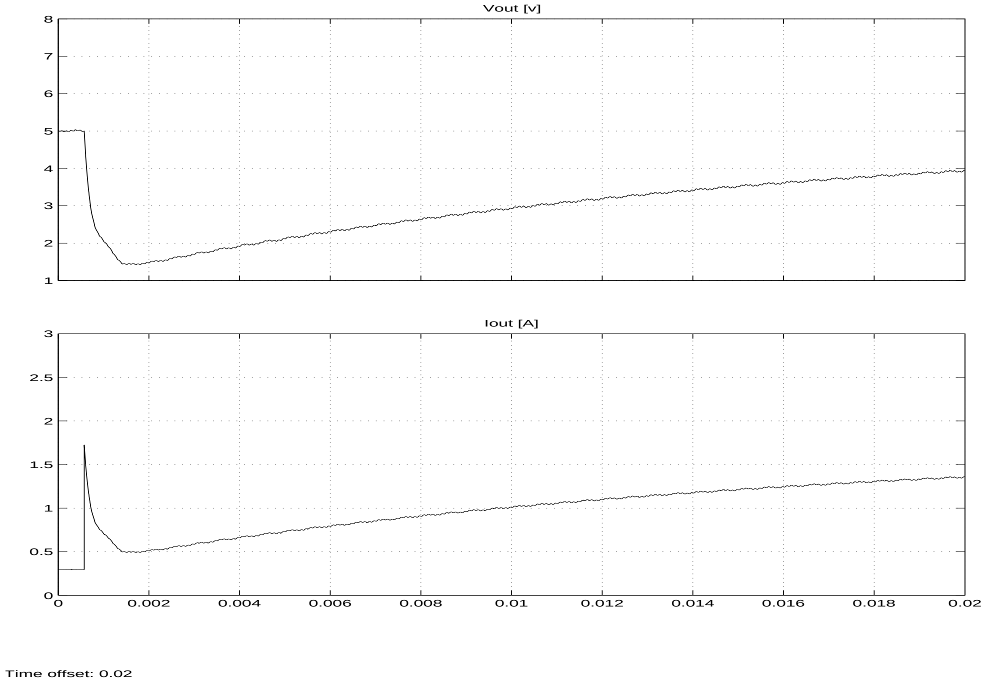


**Figure 9.28:** The output voltage and current during a load change from 17Ω to 2Ω without feed forward

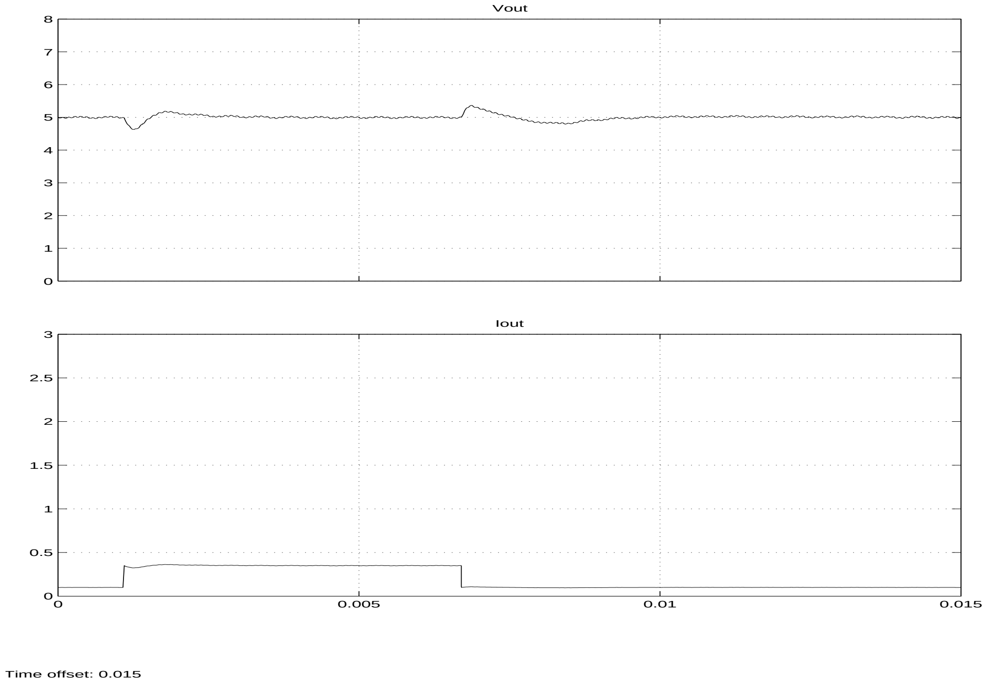


**Figure 9.29:** The output voltage and current during a load change from 50Ω to 17Ω and back

### 9.7.3 Results of Simulations

To sum up the results of the simulations it can be concluded that the MPPT fulfill the requirements and is capable of finding and obtaining the MPP for the simulated solar cell model. The controller for the PCC however is not capable of keeping the output voltage (bus voltage) within the specified requirements when exposed to large steps in load. In table 9.9 the results for the PCC controller are summarized:

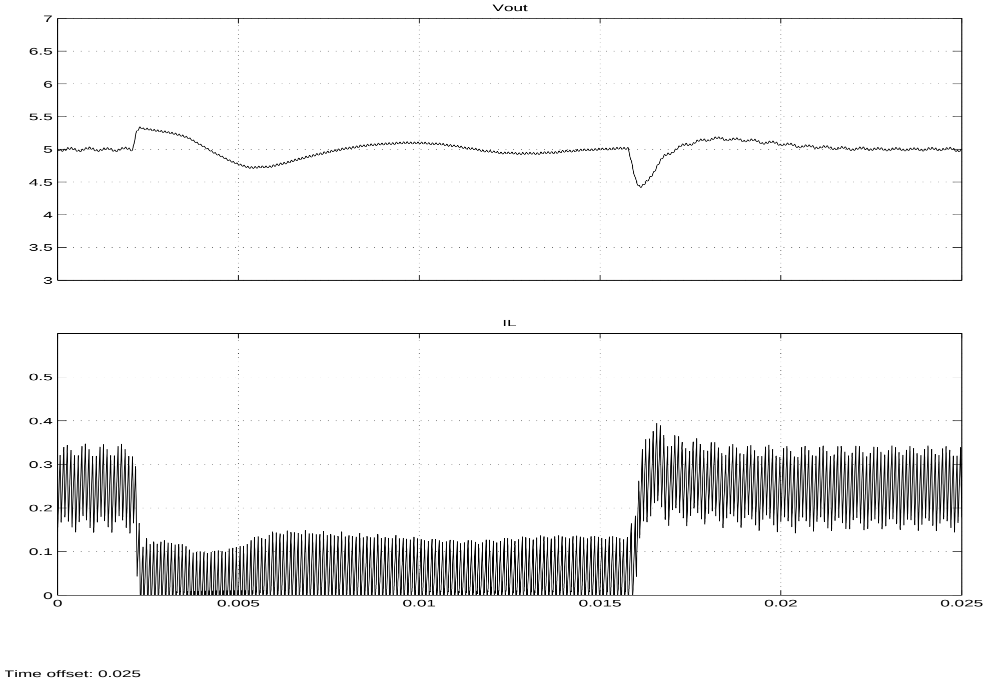


**Figure 9.30:** The output voltage and inductor current during a load change from 20Ω to 100Ω and back

| | | |
|---|---|---|
| PCC overshot | >40% | Requirement not met |
| PCC settling time | <10 ms | Requirement met |
| PCC steady state error | 0 | Requirement met |

**Table 9.9:** The results for the PCC controller

The large overshot on the bus voltage however will be less when implemented with output filters. The final test of the control will be done in connection with the software, i.e. the control will be tested together with the module test of the software.

# Chapter 10
# Digital Hardware

## 10.1 Requirements

In this chapter the digital hardware will be presented. The purpose of the digital hardware is to measure currents and voltages of certain points in the hardware topology, and to measure temperatures of point in the satellite that is exterior to the PSU itself. It must also communicate with the OBC and collect housekeeping information. Out from the measured points the controller must regulate the two converters in order to make a stable output to the power bus.

### 10.1.1 Specific Requirements for the Digital Hardware

The following requirements for the microcontroller will has been identified through discussion on how to fulfill the requirements stated in 3 on page 23 and the control loops designed in chapter 9 on page 81.

1. Internal ADC-unit capable of sampling at a rate of at least 45kHz. The multiplexed inputs the better. The resolution must be 10bit or better.
2. I$^2$C-Bus communication line
3. Temperature range from -40 °C to 80 °C
4. Must be able to start up from solar cell voltages alone, i.e. a voltage less than about 4 V.
5. 2 x Pulse-Width-Modulators (PWM) to control the duty-cycle of each converter

## 10.2 Design of Digital Hardware

In this section the digital hardware will be developed. The microcontroller ADuC812 (Data sheet is [ADuC812, 2001]) meets the requirements in the previous section 10.1.1, except for the requirements regarding PWM-modulators. This processor is build around the Intel 8051 8bit core and it has the following important peripherals:

- ADC-unit 12bit accuracy and 8 analog inputs
- I$^2$C bus hardware driver
- Two DAC outputs

The ADuC812 does not have any PWM-modulators and therefore it has been chosen to implement these externally from the MCU in such a way that they work independently from the MCU. This means that it will be possible for the OCPC circuits to shut-down the digital hardware due to e.g. an SEU without the voltages on the regulated busses are influenced to a large degree, during the reset of the MCU. This helps to make the complete PSU more robust. To implement this the two DACs will be used.

The ADuC812 has a very capable ADC-unit, which is connected to 8 internally multiplexed analog inputs. The following calculations show the maximum sample rate of the device at the chosen clocks peed of 16 MHz. The ADC internal clock is derived from the system clock.

- Optimum performance is obtained with ADC clock between 400 kHz and 3 MHz
- Total ADC conversion time is 15 ADC clocks, plus 1 ADC clock for synchronization, plus the selected acquisition time (worst case 4 ADC clocks).

To operate the ADC-unit in the frequency range described above the ADC-clock is divided by scaling down the system clock with a factor of eight giving 2 MHz. The total ADC conversion time is then 20 ADC clock cycles (best acquisition time), this yields a conversion time of:

$$T_{AD} = 20 \cdot \frac{1}{2\ MHz} = 100\ \mu s \qquad (10.1)$$

This yields a frequency of:

$$f_{AD} = \frac{1}{100\,\mu s} = 100\,kHz \tag{10.2}$$

### 10.2.1 Starting up the PSU

When the Cubesat is released into space, a kill-switch is activated to start up the PSU. This switch will activated both the solar array and the batteries at the same time. When starting up the PSU, the microcontroller is powered directly from the batteries or the solar array. The batteries and solar array are activated by a kill-switch that is released when the satellite is deployed in space. When the satellite is activated by the kill-switch, the PSU's microcontroller will be supplied either by the batteries or the solar array.

By using some diodes this can be designed such that the digital hardware will select where to draw its current from either solar cells, battery or the main power bus depending on the voltages at these points. This will allow the PSU to start-up even on flat (or damaged) batteries and when the PSU operate normally it uses power from the main power bus, which has the most stable voltage.

The design of this circuitry can be seen in figure 10.1. Across the batteries a Zener Diode with zener voltage of 3.9 V is placed. Across the solar array a Schottky diode, see data sheet [BAT85, 2000], with a forward voltage of approximately 0.3 V will be placed and the same Schottky diode will be placed over the power bus. In the figure the Protection Circuit is shown in connection with the MCU.

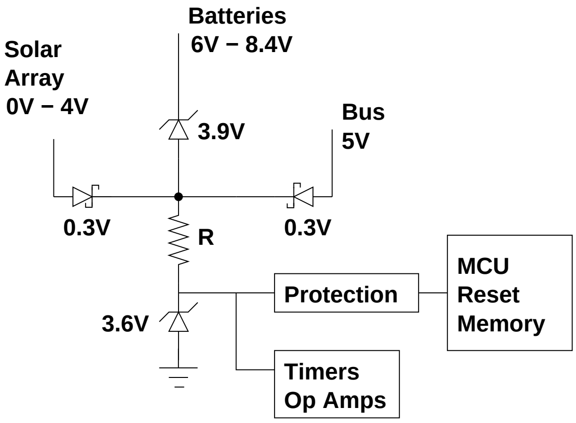


**Figure 10.1:** Starting up the PSU, by using two Schottky diodes and a zener diode

The maximum voltage level of the batteries is 8.4 V, because of the zener voltage of 3.9 V the supply voltage to the MCU will be 4.5 V. The maximum voltage from the solar array is 4.0 V and with a forward voltage of 0.3 the supply voltage will be 3.7 V. It is now clear that when the kill switch i activated, the MCU be supplied either by the solar array or the batteries, this is depending on the voltage levels. This is an advantage for the PSU, that it can start up without power from the batteries and then only use the power from the solar array. If we assume that the power bus has been stable for some time and will suddenly be shut down by the Protection Circuit, then the MCU will still run because the supply to the MCU will be automatically switched over and use the power from the batteries or the solar array. The only circuit that will not be dependent on the protection is the timers, which may not be shut down, because they must be able to generate PWM signal to the converters for some time period.

In the figure 10.1 the resistor **R** will now be determined. The reason for using a resistor here is to prevent short circuiting. The maximum voltage level from the power bus is 5 V and because of the Schottky diode with a forward voltage of 0.3 V the voltage will be 4.7 V. The devices the must be supplied are using 3.6 V and the current that passes through the circuits is estimated to about 50 mA[1], see section 10.7.

Then the resistors value can be determined, using the current consumption of 50 mA, to be:

$$R = \frac{V_{max} - V_{min}}{I_{DD}} \Longrightarrow R = \frac{4.7\,V - 3.6\,V}{50\,mA} = 22\,\Omega \tag{10.3}$$

[1] This value has been re-estimated as the design process has proceeded

### 10.2.2 Power-On Reset Operation for MCU

An external power-on reset (POR) circuit must be designed to reset the ADuC812. This circuit must hold the reset pin asserted (high) whenever the power supply ($DV_{DD}$) is below 2.5 V. The $DV_{DD}$ must remain above 2.5 V for at least 10 ms before the reset signal is negated (low). Whenever the supply voltage is below 2.5 V, the reset must be asserted. This voltage level is called threshold voltage ($V_t$). Furthermore, in the data sheet for the MCU, it says that the external POR circuit must be operational down to 1.2 V or less, see data sheet [ADuC812, 2001] page 41. POR and power off (power cycling) is required because of the use of external memory [ANALOG, 2001], it e.g. the software fails then the power to the MCU must be shut down by the Protection Circuit. After some time the MCU is tried to be powered up and it will be automatically reset by the POR circuit. The POR circuit is required to clear all the registers in the MCU.

**Specific Requirements for the POR**

From the above we can set up the specific requirements for the POR:

- Low supply voltage - down to 1.2 V or less
- Reset is active high
- Minimum Threshold voltage ($V_t$) 2.5 V

The POR circuit that will be used in the PSU in Cubesat to reset the MCU is MCP101 from Microchip, see data sheet [MCP101, 2001]. This circuit meets the previous requirements. The POR circuit must be connected with the Protection Circuit as shown in figure 10.2. For further information about how the Protection Circuit is working, see chapter 8 on page 71.

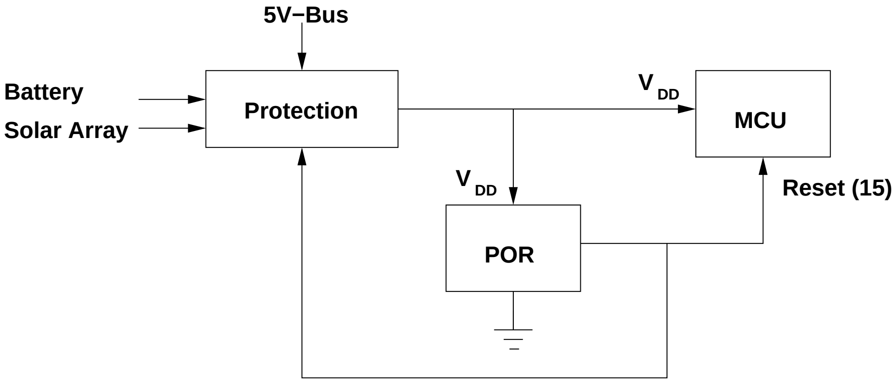


**Figure 10.2:** POR circuit in connection with the MCU

### 10.2.3 External Memory

In this subsection the interface between the MCU and external memory will be described and how to select which code space (internal or external program memory) to begin executing instructions.

The internal program memory on the ADuC812 is 8 KB flash (EEPROM). This program memory will be used under developing the software, but when all tests are completed the software will be implemented in the external memory. The reason for implementing the software in external memory rather than in internal, is that in space applications some data in flash memory can be erased if it is hit by a high energy particle, see chapter 2 on page 21.

On the ADuC812 there is a pin called $\overline{EA}$, which is used to select code space. The $\overline{EA}$ pin must be tie high to access external memory and low if internal memory is used. To select external or internal memory (high or low, respectively) a jumper will be used in the prototype. But in the PSU which is sent into space, the external memory must be used and therefore the $\overline{EA}$ pin will be tied high.

**Specific Requirements for the External Memory**

The following will describe the specific requirements for the external memory that must be used. Because of the amount of software which must be used to control the converters a minimum amount of memory is needed. The voltage level that the memory must be operating in must be very close to the MCU's voltage levels. The specific requirements for the external memory is the following:

- Minimum 8 KB PROM

- Voltage levels between 2.7 V and 5.5 V

The interface between the MCU (ADuC812) and external memory will now be described. The external memory must be connected as shown in figure 10.3. The MCU can only access program memory less that 64 KB. Two port on the ADuC812 will be used (port 0 and port 2), the ports are 8 bit wide, this will give 16 I/O lines. Port 0 (P0) is used to multiplex the address/data bus. First it emits the low byte of the program counter (PCL) as an address. During the time where the PCL is valid on P0, the address latch enable (ALE) clocks this byte into an latch. In this time port 2 (P2) emits the high byte of the program counter (PCH) and $\overline{PSEN}$ (Program Store Enable) strobes the PROM (OE - output enable) and the code byte is read into the ADuC812 [ADuC812, 2001] page 40.

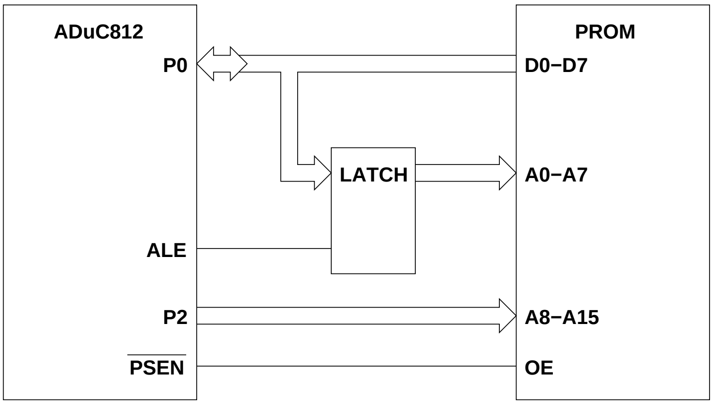


**Figure 10.3:** External Program Memory Interface

### 10.2.4 Interface with the I$^2$C bus and configuration

This subsection will describe the implementation of an I$^2$C (inter integrated circuit) and how it must communicate with the OBC by using the microcontrollers I$^2$C compatible interface.

To establish a connection with the I$^2$C-bus, only two bus lines are required– a serial data line (SDA) and a serial clock line (SCL), see figure 10.4. These lines are bi-directional, meaning that the master and slave can operate as transmitter and receivers.

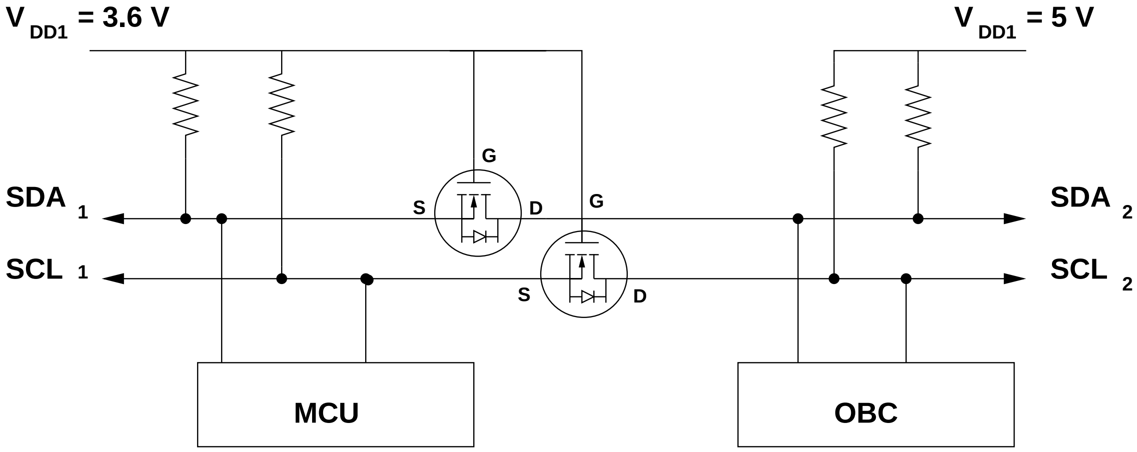


**Figure 10.4:** The I$^2$C interface with the OBC, where the ADuC812 is slave and OBC is master

The SCL is always transmitted from master to slave. As seen in the figure 10.4, pull-up resistors are needed on both SCL and SDA lines. These lines are separated by two MOSFES's, this is done because the use of different voltages levels [Semiconductors, 2000].

## 10.3 Interfacing to the Internal ADC

In order to control the converters and collect housekeeping information a number of currents, voltages and temperatures must be measured. Figure 10.5 depicts the placement of all current and voltage measurement points in the PSU-topology. In the figure all measurement points have an associated identifier which consists of an unique number an a letter which identifies if the value to be measured is a voltage (V) or a current (I). On the figure the PSU is depicted to be connected to the main power bus which will be the case under normal operation.

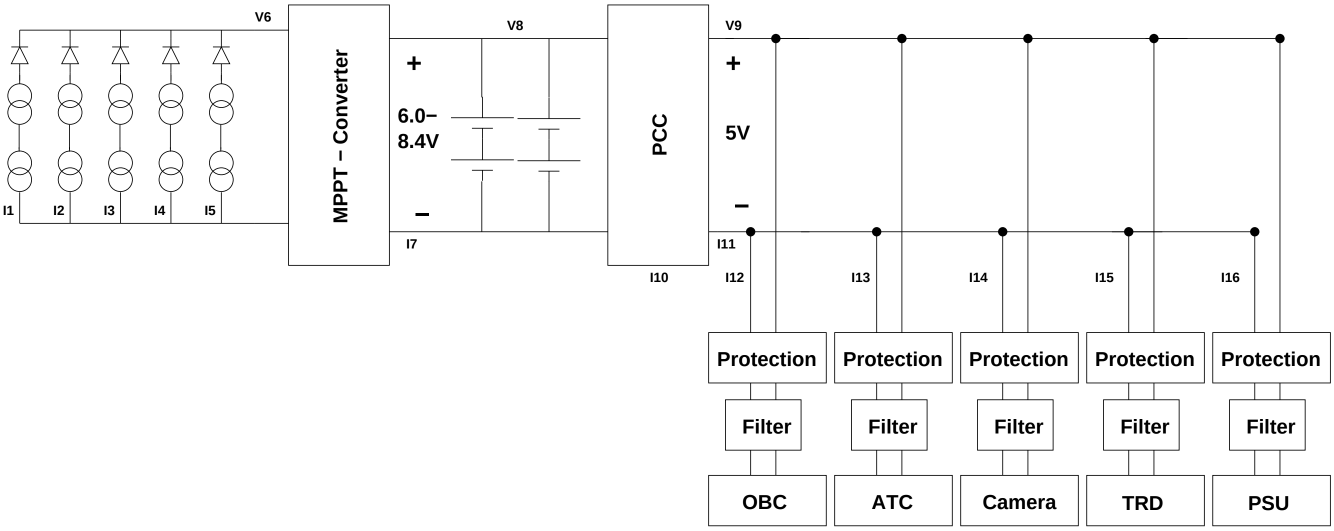


**Figure 10.5:** Measurement points in the PSU

The points to be measured for control of converters have been determined from the chapter on converter control (see chapter 9 on page 81) and the points to be measured for housekeeping have been determined from the requirements specification (see chapter 3 on page 23), which also states that 7 temperatures in the satellite must be measured by the PSU. Table 10.1 lists all measurement points along with ranges of values and needed frequency.

In the following it will be described how these signals will be measured. First the components used will be described and then the specific design that allows all point to be measured will be designed.

### 10.3.1 The ADC and External Components

The ADuC812[2] is called a mircroconverter rather than a microcontroller because it integrates both a MCU core and a data-acquisition system that incorporates both an analog multiplexer and an advanced ADC. The following will describe both the internal ADC-system as well as external components that are needed to interface the 23 measuring points to the 8 analog inputs of the ADuC812.

### 10.3.2 ADuC812 ADC

The internal analog multiplexer is capable of multiplexing 9 inputs to the 12-bit ADC and 8 of these can be accessed externally through port 1 of the MCU. The 9th input is from the internal temperature sensor and can thus be used to measure the temperature of the PSU (T21). The internal structure of the multiplexer and ADC are depicted in figure 10.6.

The sampling process consists of two stages. First the input signal is latched from the analog input port to the internal holding capacitor this is done in what is called the acquisition time, which is programmable. Then next SW1 and SW2 switches and the ADC begins to convert the voltage held at the holding capacitor using successive approximation.

When the multiplexer chooses another input port the charge held in the holding capacitor may propagate backwards and cause a transient on the

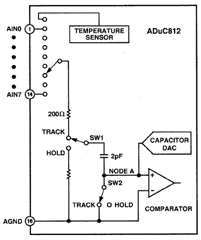


**Figure 10.6:** Internal ADC structure

[2] All specifications from datasheet: [ADuC812, 2001]

| Identifier | Description | Physical Range | Frequency [Hz] |
|---|---|---|---|
| I1 | Current through solar-panel 1 | 0-0.2A | 1/5 |
| I2 | Current through solar-panel 2 | 0-0.2A | 1/5 |
| I3 | Current through solar-panel 3 | 0-0.2A | 1/5 |
| I4 | Current through solar-panel 4 | 0-0.2A | 1/5 |
| I5 | Current through solar-panel 5 | 0-0.2A | 1/5 |
| V6 | Voltage across parallel connection of solarcells | 0-6V | 1/5 |
| I7 | MPPTC return current | 0-0.5A | 100 |
| V8 | Voltage across parallel connection of batteries | 0-10V | 100 |
| V9 | Voltage on main power bus | 0-7V | 1600 |
| I10 | PCC inductor current | 0-3A | 18000 |
| I11 | PCC return current | 0-3A | 18000 |
| I12 | OBC current | 0-3A | 1/5 |
| I13 | ATC current | 0-3A | 1/5 |
| I14 | Camera current | 0-3A | 1/5 |
| I15 | TRD current | 0-3A | 1/5 |
| I16 | PSU current | 0-3A | 1/5 |
| T17 | OBC temperature | -55-150° | 1/5 |
| T18 | ATC temperature | -55-150° | 1/5 |
| T19 | Camera temperature | -55-150° | 1/5 |
| T20 | TRD temperature | -55-150° | 1/5 |
| T21 | PSU temperature | -55-150° | 1/5 |
| T22 | FREE temperature | -55-150° | 1/5 |
| T23 | FREE temperature | -55-150° | 1/5 |

**Table 10.1:** Table of all points that must be measured by the PSU

input signal which in effect will cause the next sample to be inaccurate.
In order neglect this problem the input should be buffered and filtered as depicted in figure 10.7.

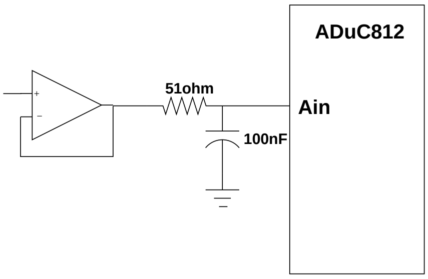


**Figure 10.7:** ADC input filtering

This filter will filter away the possible transient due to discharging of the holding capacitor, but by choosing the right values of the capacitor and resistor one can also use the filter as Nyquist sampling filter. When this is done care should be taken in order to select a resistor value that is small enough not to disturb the sampling. This is done by selecting the resistor value such that the voltage drop that develops across the resistor when the leakage current into the ADC is 10 $\mu A$ does not exceed the voltage of the Least Significant Bit (LSB).

In order to operate, the ADC requires a stable reference voltage between 2.3 V to Vcc. This can be supplied either externally or an internal voltage reference of 2.5 V can be used. It has been chosen to use the internal reference since it is more stable than what can easily be generated externally considering the wide range of supply voltages that will be used for the digital hardware. This means that all voltages at the analog inputs must be in the interval 0-2.5 V.

**Multiplexers**

As described the ADuC812 is only capable of multiplexing 8 external signals to the ADC. Therefore, external analog multiplexers are required in order to measure all 23 values. A part from being able to multiplex signals these components must live up to the following critical requirements:

- must be available in the industrial graded temperature range
- must be able to operate from 3.3 V
- should consume very little power

The ADG704, ADG708 and ADG706 from Analog Devices (Datasheet is: [ADG708, 2000] and [ADG704, 1999]) live up to these requirements. They are 4:1, 8:1 and 16:1 multiplexers respectively. They operate from 1.8-5.5 V and uses typically 0.001 $\mu A$ at 5.5 V. Further, they have a on-resistance of a few ohms.

**Op-Amps**

Operational amplifiers are required to condition and buffer the signals to the analog inputs of the ADuC812. The same requirements as stated above for the multiplexers apply for the Op-Amps. The following two Op-Amps have been found.

The OP181/OP281 (Datasheet: [OP181, 1996]) are micro-power Op-Amps from Analog Devices that have 1 and 2 amplifiers per. chip respectively. Each amplifier consumes a maximum current of 4 $\mu A$, but the unity-gain-bandwidth-product (UGBP) of the amplifier is only 95 kHz, which means that at signals of 10 kHz the open-loop gain is only about 20 dB. Therefore, at high frequencies there is not enough gain to suppress the Op-Amp offset voltage, which typically is 1.5 mV. This amplifier will therefore be used to condition signal for the measuring points of low frequencies i.e. those of 1/5 Hz.

The other Op-Amp is the OP191/OP291 (Datasheet:[OP191, 2000]) (also 1 & 2 amplifiers respectively). While each amplifier uses 300 $\mu A$ they have an UGBP of 3 MHz and a typically offset voltage of 700 $\mu V$. Therefore, these amplifiers are better suited for measuring signals with higher frequencies and they will therefore be used to condition the signals that will be used for converter control.

### 10.3.3 Allocation of Analog Inputs

As can be seen on table 10.1 there are two distinct types of measurements; High frequent measurements for control and low frequent measurements for collection of housekeeping data. It can also be observed from the table that the housekeeping measurements all fall in groups of similar measurements i.e. same physical range whereas the measurements for control are more diverse. On figure 10.8 it can be seen how the analog input-ports of the MCU are utilized together with analog multiplexers and Op-Amps in order to measure all 23 points from table 10.1.

As it can be seen from the figure generally all housekeeping measurements are channeled to input **AN0** on the MCU through multiplexers and all control measurements are all connected directly to the analog inputs of the MCU.

It can also be seen that the housekeeping measurements are grouped such that each multiplexer multiplexes one group of similar measurements that need the same amplification. One exception from this is the PSU load current (I16) and the current through through solar panel 5 (I5), these could not be fitted together with the similar measurements on the 4:1 multiplexers.

The reason for connecting all control points directly to the MCU is that their physical range of measurement vary and it is therefore beneficial to provide each point its own amplification in order to achieve higher accuracy and better utilization of the measurement range of the ADC. Further, since the control points are measured more often than the housekeeping data, there is less overhead associated with programming the multiplexers when the control points are not multiplexed.

### 10.3.4 Specific Design and Transducers

The following describes exactly how the physical values are measured for each group of measurements and the specific choice and design of transducers as well as input amplification and filtering circuitry.

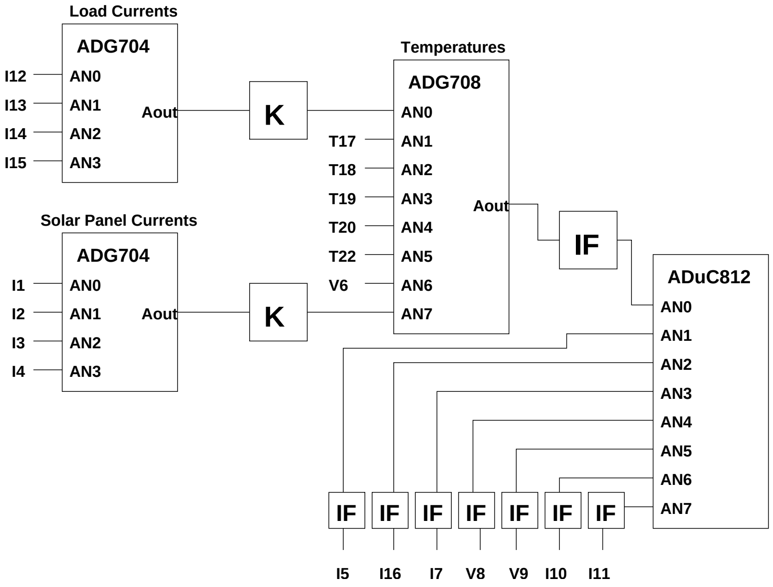


**Figure 10.8:** Allocation of signals to multiplexers. K are amplifiers and IF are input filters. Multiplexer control signals are not shown

### Housekeeping Measurements of Temperatures

The temperature of the PSU itself is measured (T21) using an internal temperature transducer in the ADuC812. The remaining temperature measurements are performed by LM35CZ transducers that produces a voltage proportional to the temperature of the chip in the range of -55°to 150°.

Since the LM35 generates a negative voltage at a negative temperature and since tha internal ADC in the MCU only measures positive values, it is necessary to adjust the output voltage from the LM35 sensor. This is done by moving the common. The common has to be moved minimum 550 mV since then the output voltage will be 0 at -55°and 2.05 V at 150°.

The demand for the reference to be used as common for the LM35 temperature sensors are: Temperature independce, current independce and low power consumption.

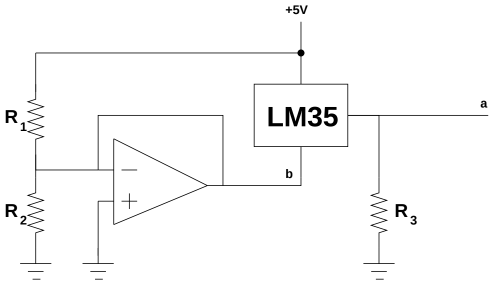


**Figure 10.9:** Using the LM35 for temperature measurements

Moving the common requires a stable reference voltage at the required offset voltage. This could be made with at diode but since diodes are temperature dependent and will change their voltage drop according to the temperature and current, another solution must be used.

The solution chosen for this task is a Op-Amp as a buffer for a voltage divider. This can be seen in figure 10.9. The resistors $R_1$ and $R_2$ makes the reference voltage and the Op-Amp buffers it, ensuring that the current drawn by the transducers will not change the reference voltage.

In order to measure all the different temperatures all the LM35 transducers will share the same **b** point which is placed on the PSU circuit board. All transducers that are placed outside the PSU are then connected with this voltage and the supply voltage from the PSU. This means that the temperature of a subsystem can be measured even when that system is not powered. The voltage at **b** is considered to be constant.

Since the transducer LM35 needs minimum 4 V in order to operate it is chosen to supply it with 5 V from the power-bus. This means that the offset voltage at **b** must be 1 V. This means that the output amplitude af the transducers vill be minimum: 1-0.55 = 0.45 V and maximum 1+1.5 = 2.5 V.

The voltage divider, $R_1$ and $R_2$ makes the reference voltage that has to be 1 V.

$$U_{ref} = U_{cc} \cdot \frac{R_2}{R_1 + R_2} \leftrightarrow R_1 = 5 \cdot \frac{1M}{4} - 1M = 250k\, Omega$$

The resistor $R_3$ is according to the data sheet to calculated as in equation 10.3.4.

$$R_3 = \frac{V_{ref}}{50 \cdot 10^{-6}} = \frac{1}{50 \cdot 10^{-6}} = 20k\,\Omega$$

As Op-Amp the OP191 is used. On the reference output from this Op-Amp, which are to be connected 5 LM35s, each sinking less then 60 $\mu$A through the Op-Amp, giving a maximum current of 300 $\mu$A through the Op-Amp. Since the Op-Amp can sink 8.4 mA this is considered to be safe.

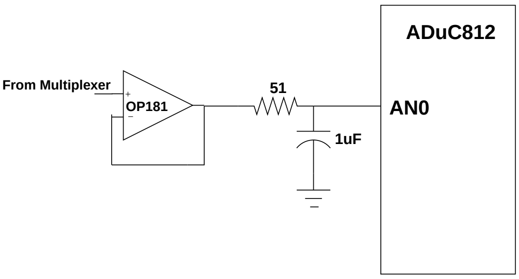


**Figure 10.10:** Input circuitry for housekeeping measurements

The RC-filter part of the input circuitry is not required to act as a Nyquist sampling filter in this case, since all the housekeeping data that are measured through this filter only vary slowly (e.g. temperatures) or they are already being sampled far below the Nyquist rate (Currents). The filter is therefore only required to filter away the switching transient of the internal multiplexer. The complete circuit is shown on figure 10.10.

The value of the capacitor is somewhat larger than the 100 nF that was proposed in subsection 10.3.2. This value has been selected in order to lower the corner frequency of the RC-filter such that it helps to filter away high-frequent noise on the signal. The corner frequency is:

$$f = \frac{1}{2\pi \cdot 51\Omega \cdot 1uF} \simeq 3.1\ kHz$$

The value of the filter-resistor is kept small since a larger value may disturb the signal being measured because of the leakage current into the MCU of maximally 10 $\mu$A, which leads to a voltage drop over the resistor of:

$$U = 10\ \mu A \cdot\ 51\Omega = 510\ \mu V$$

This value is less than the voltage corresponding to a change of the least significant bit, which is:

$$U_{LSB} = \frac{2.5\ V}{4096} = 610\ \mu V \tag{10.4}$$

The Op-Amp is the ”slow” OP181 and is chosen because of the low frequency requirements for these measurements. Since all 14 housekeeping signals which are being measured at 1/5Hz pass through this amplifier the frequency requirement is:

$$F = 14 \cdot 0.2\ Hz = 2.8\ Hz$$

The OP181 draws maximally 10 nA from the inputs and this current is drawn through the ADG708 which has an on-resistance of maximally 5 Ω, this yield a voltage drop of:

$$U = 10\ nA \cdot 5\ \Omega = 50\ nV$$

Because this voltage drop is far below the LSB voltage it is of no consequence.

### Housekeeping Measurements of Solar-Panel Currents

In order to measure the current through each solar panel (I1-I5) a resistor of 100 $m\Omega$ is placed in series with each solar panel between the ground terminal and the panel. The current of maximally 0.5 A which passes through this resistor causes a voltage drop of:

$$U_{resistor} = 100m\ \Omega \cdot 0.5\ A = 50\ mV$$

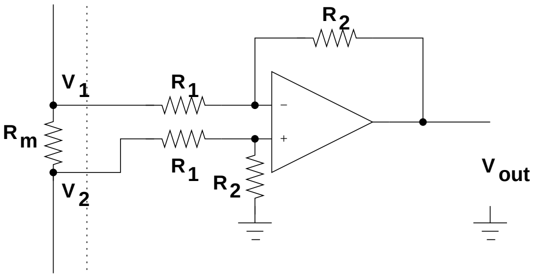


**Figure 10.11:** The differential amplifier

To use the full ADC range of 2.5 V this signal must be amplified by a factor of 50. This is done with a differential amplifier as shown in figure 10.11

This circuit has the following voltage transfer characteristics [Sedra and Smith, 1998] (page 88):

$$V_{out} = \frac{R_2}{R_1}(V_2 - V_1)$$

Since these current measurements are low frequent the OP181 amplifier is used and $R_1$ is chosen to 10 $k\Omega$ and $R_2$ is 500 $k\Omega$. For measurement I5, which is connected directly to the ADuC812, the above circuit is implemented directly with the addition of the input RC-filter as described for the temperature measurements.

In the cases of measurements I1-I4 the $V_1$ measurement is connected through the multiplexer of figure 10.8 and the $V_2$ is connected to the ground terminal that is shared by all resistors. This is depicted in figure 10.12.

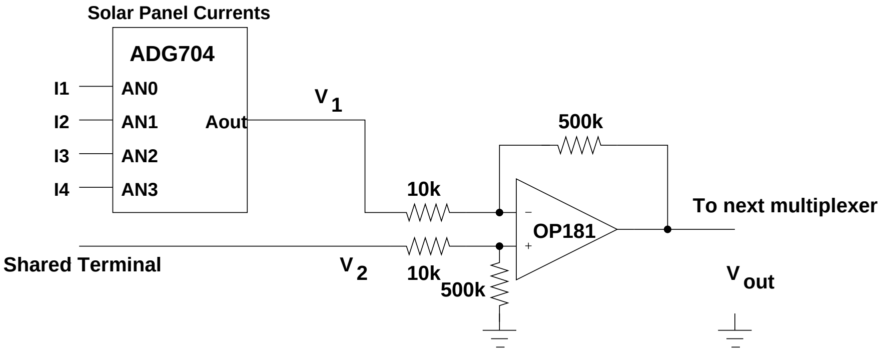


**Figure 10.12:** The differential amplifier and multiplexer

### Housekeeping Measurements of Load Currents

The currents supplied to each load must be measured (I12-I16). All these currents pass through individual protection circuits as described in chapter 8 on page 71, which has a PMOS-transistor connected in the current path. This transistor exhibits a on resistance of about 10 $m\Omega$ depending on temperature. Assuming these transistors to be at the same temperature it is possible to use this resistance to measure relative currents of each load. Now since the total load return current is measured for control (I11) this measurement can be used to normalize the individual load currents. To make this solution work it is a requirement that the transistors are thermally coupled on the circuit board.

The circuitry that is to perform these measurements is similar to that of figure 10.12 except that the input signals now are I12-I15 and the amplification is changed to accommodate the larger measurement range of 0-3A which gives an amplification of 83 times this is implemented with $R_1 = 10k\Omega$ and $R_2 = 833k\Omega$. Because of the large amplification, the OP191 Op-Amp is used, since the OP181 does not have a large enough gain to adequately suppress the Op-Amp offset voltage at this gain.

Measurement of I16 which is directly coupled to the ADuC812 is performed similarly to measurement I6 except the change of Op-Amp and changes of resistor values.

### Housekeeping Measurement of Solar Panel Voltage

Measurement V6 is the measurement of solar panel voltage and has a physical range of 0-6V. To measure this voltage the signal must be down-scaled. A voltage division of 6V/2.5V=2.4 is needed. To implement this a voltage divider with resistors of 200 $k\Omega$ and 480 $k\Omega$ are used. The output is connected to a OP181 Op-Amp configured as a voltage follower and the standard input-filter as described under temperature measurements is used to filter transients from the ADC multiplexer.

**Current Measurements for Converter Control**

Measurement points I7, I10 and I11 are all current measurements in the range 0-3A and are measured by inserting a 10 $m\Omega$ resistor in series with the current path. The amplification circuit that is used for each measurement is the differential amplifier as for the load current measurements, but the gain of the amplifier is adjusted such that a current flow of 1A corresponds to a digital value of 1024. This is to make the values sampled directly usable in the calculations on the MCU. The number format used by the software is described in section 11.7.2 on page 141. Since $U_{lsb} = 610\mu V$ then the 10 mV measured over the resistor should be amplified to $1024 \cdot u_{lsb}$ this gives an amplification of:

$$A = \frac{1024 \cdot 610\mu V}{10mV} \simeq 62.5$$

This gain is achieved by using resistors with the values $R_1 = 10k\Omega$ and $R_2 = 625k\Omega$

The corner frequency of the input filter of measurement I10 and I11 are specified in table 9.6 on page 98 and is 4.5 kHz. This gives the following capacitor size with a resistor of 51$\Omega$:

$$C = \frac{1}{2 \cdot \pi \cdot R \cdot F_c} = \frac{1}{2 \cdot \pi \cdot 51\Omega \cdot 4.5kHz} = 0.69uF$$

The corner frequency of the input filter of measurement I7 is specified in table 9.1 on page 92 and is 500 Hz. This gives a capacitor of 6.2 uF.

**Voltage Measurements for Converter Control**

The last two measurement points to be designed are V8 and V9 which measures voltages for converter control. These measurements will be performed similar to the housekeeping measurement of solar panel voltage.

The physical range of V8 is 0-10V and the filter requirement is 500 Hz (see table 9.1 on page 92). Again the signal must be scaled to suit the internal number representation scheme in the MCU, which means that 1 V should correspond to $1024 \cdot U_{lsb}$ on the ADC input. A voltage divider with a gain of:

$$A = \frac{1024 \cdot 610\mu V}{1V} \simeq 0.63$$

To achieve this gain, resistors of 400 $k\Omega$ and 250 $k\Omega$ are used.

The Op-Amp for the input filter is the OP181 and it is configured as a voltage follower and the input filter consists of a resistor of 51$\Omega$ and a capacitor of 6.2uF.

The physical range of V9 is 0-7V and the filter requirement is 400 Hz (see section 9.6 on page 98). The voltage divider will again consist of resistors of 400 $k\Omega$ and 250 $k\Omega$. The Op-Amp is the OP191, due to the high frequency requirement, and it is configured as a voltage follower and the input filter consists of a resistor of 51$\Omega$ and a capacitor of 7.8 uF.

### 10.3.5 Addressing the Multiplexers

In order to specify which signal the external multiplexers is to route through to the analog input of the MCU the multiplexers must be supplied with control signals that select which signal they route from input to output.

The ADG708 has 3 such signals and each of the two ADG704's have 2 control signals. Since only one of the ADG704 can be active at a time (i.e. routed through the ADG708) the two chips are able to share two control lines. The total amount of control lines needed then are 5 and these are taken from the latch described in subsection 10.2.3.

### 10.3.6 Common and Calibration

It should be noted that in all measurement configurations the Op-Amp common connections are connected to system common which is located at the common terminal of the solar panel strings (including resistors for measurement). The absolute lowest potential of the PSU circuitry is located at this point and it therefore gives the Op-Amps the best onditions available for amplification of the small signals with voltages close to system common.

The common terminal connected to resistor $R_2$ in the differential amplifier design (see figure 10.12 is connected to the common potential of the MCU. This means that the output of the Op-Amp is the differential signal on the Op-Amp inputs amplified with regard to the MCU common potential and it is thus possible to measures the amplitudes on the MCU.

Normally with earth based designs the Op-Amps in circuitry like what is designed here will be calibrated with potentiometers or multi-turn resistors in order to minimize the error due to the Op-Amp offset voltage. This is not possible in space since it is not desirable to bring mechanical components like potentiometers into space. Therefore, in order to calibrate the circuits, measurements on earth should clarify deviations from actual values. This deviation can then in software be used to adjust the measurements. To enhance accuracy even more different deviation constants should be used for different temperatures of the measurement circuitry.

## 10.4 Pulse Width Modulators

In this chapter the development of the PWM will be described. Since the circuits are alike for both the MPPTC and the PCC, they will be described together and the component values will be calculated for both PW-modulators.

The PWM is to be controlled by the MCU. In order to do this the two analog ports are used. The analog voltage from the MCU is proportional with the the desired width of the pulses.

The two PW-modulator circuits each consist of two sub-circuits; an oscillator and a PW-modulator. The oscillator generates the modulation frequency and the PW-modulator then modulates the width of the pulses, based on the analog voltage from the MCU.

Both the oscillators and the pulse width modulator are build around the 555 timer, in order to save PCB space, power and to get the ability to supply power with an incoming voltage of only 3.6 V, two TS3V556 ICs are used.

### 10.4.1 Pulse Width Modulators

The basic circuit of a 555 timer can be seen in figure 10.13-A. Using the timer as a PW-modulator requires a resistor and a capacitance. The diagram for the the two PW modulators can be seen in figure 10.13-B.

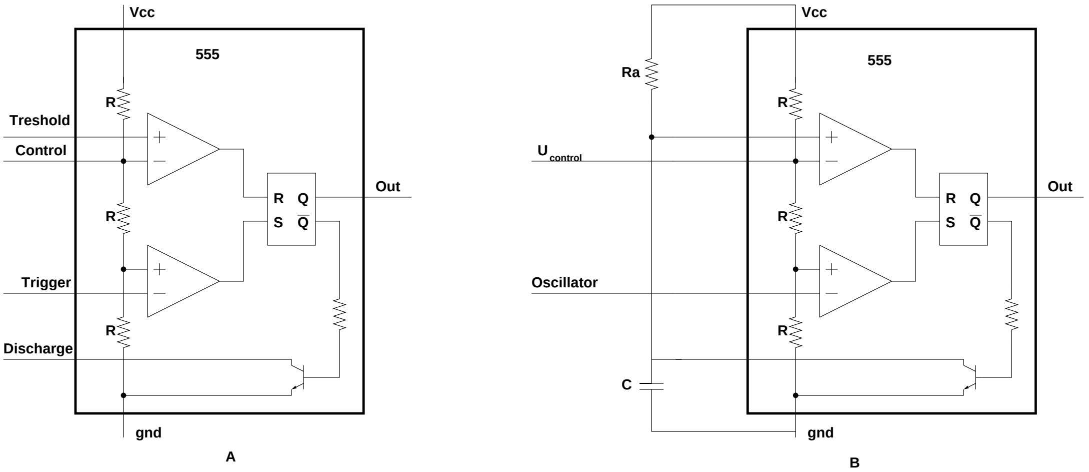


**Figure 10.13:** The oscillator consisting of a 555 timer, a capacitance and a resistor. Figure A show the basic diagram of a single 555 timer that will be used to calculate component values. Figure B show the pulse width modulator.

The analog voltage that is controlling the length of the pulses has to be higher then 0 V and lower than 2/3 $U_{cc}$. Looking at the diagram in figure 10.13-B makes it easier to calculate these two values. As a point of start the value of C is chosen to be $0.001\mu$F, this low value is chosen based on the fact that lesser value[3] equals lesser volume and lesser weight.

The oscillator input is a negative peak with a duration near 0 %. This duration is used in order to give the PW-modulator more time to operate. If the duration of the oscillator was 50 % the pulse width could only variate between 0 - 50 %. The oscillator voltage could as a minimum be either $U_{oscillator-high} < \frac{1}{2}U_{control}$ or $U_{oscillator-low} > \frac{1}{2}U_{control}$.

[3] Measured in Farad [F]

When the PW-modulator receives the negative peak it toggles the S input of the flip-flop twice, making the output go high. After discharging the capacitance will start charging and after a while reach the $U_{control}$. The voltage level on $U_{control}$ control the time it takes the capacitance to reach it. If the control voltage is high the capacitance will need more time to charge to this level. The resistor and capacitance must be proportioned so that a charge from 0 to $\frac{2}{3}U_{cc}$ will last for one switching period. This is mainly because the charging of the capacitance is non linear and therefore is to be kept ed from 0 to $\frac{2}{3}U_{cc}$ in order to use the almost linear part of the charging, when charging from 0 to $U_{cc}$

In order to calculate $R_a$ and $C$ the value of $C$ is chosen to 0.001$\mu$F. The value of $R_a$ is calculated by the use of formula 10.5.

$$\begin{aligned}
U_{cap} &= U_{cc} \cdot \left(1 - e^{\frac{-T}{R_a \cdot C}}\right) \Leftrightarrow \\
1 - \frac{U_{cap}}{U} &= e^{\frac{-T}{R_a \cdot C}} \Leftrightarrow \\
ln\left(1 - \frac{U_{cap}}{U_{cc}}\right) &= \frac{-T}{R_a \cdot C} \Leftrightarrow \\
R_a &= \frac{-T}{ln\left(1 - \frac{U_{cap}}{U}\right) \cdot C}
\end{aligned} \tag{10.5}$$

Where:
$U_{cap}$= The voltage over the capacitance [V]
$U_{cc}$= The supply voltage [V]
$R_a$= The resistor that limits the current to the capacitance [$\Omega$]

The components for the two PW-modulators are listed in table 10.2

| | Frequency | $R_a$ | C |
|---|---|---|---|
| PW-modulator 1 | 50 kHz | 18 k$\Omega$ | 0.001$\mu$F |
| PW-modulator 2 | 21 kHz | 44 k$\Omega$ | 0.001$\mu$F |

**Table 10.2:** The component values for the two PW-modulators

### 10.4.2 Oscillators

In order to supply the PW-modulator with a clock consisting of a stable frequency and a duty cycle of almost 100% a 555 timer circuit is used in order to build an oscillator. The circuit making up the oscillator can be seen in figure 10.14. The circuit only differs from the one used as the PW-modulator in one major way; the control voltage, $U_{control}$, is not used here and instead a bypasse capacitore is used.

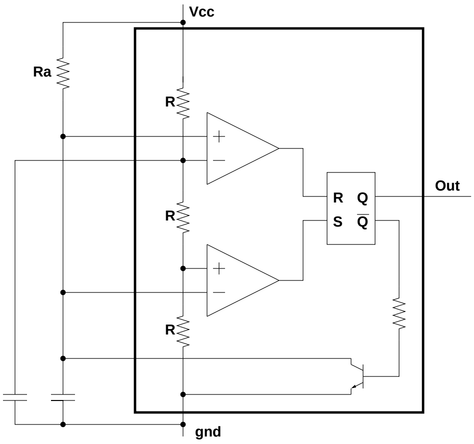


**Figure 10.14:** The 555 timer as a oscillator. The output will be a frequency with a duty cycle close to 100%

In order to calculate the values of $R_a$ and C the same formula as the one used for the PW-modulator is used. If formula 10.5 is used to calculate the resistance in a PW-modulator $U_{cap}$ must be replaced with $\frac{2}{3}U_{cc}$. As a point of start the capacitance is chosen to be 0.001 $\mu$F, due to the fact that the value of 0.001 $\mu$F is the recommended value at a free running frequency of 20 kHz and a total resistance ($R_a + R_b$) of 100 k$\Omega$ in the astable mode of operation[4] [LM555, 2000].

Since the same value is chosen for all four[5] capacitors and due to that the charging time needed is the same for the oscillators and PW-modulators in the two pairs the value of the resistors will also be in pairs - that is the two resistors for the MPPTC PW-modulator are the same.

### 10.4.3 Simulation

Simulating the circuit in figure 10.15 will produce the output seen in figure 10.16. The circuit is designed to switch at a frequency of 21.2 kHz, therefore the resistor are 47 k$\Omega$ according to formula 10.5. The voltage source, V5, simulates the analog voltage from the MCU and the voltage is also visible in figure 10.16. This voltage will during the simulation sweep from 0 to 3.6 V so that a PW from 0 to 100% is visible.

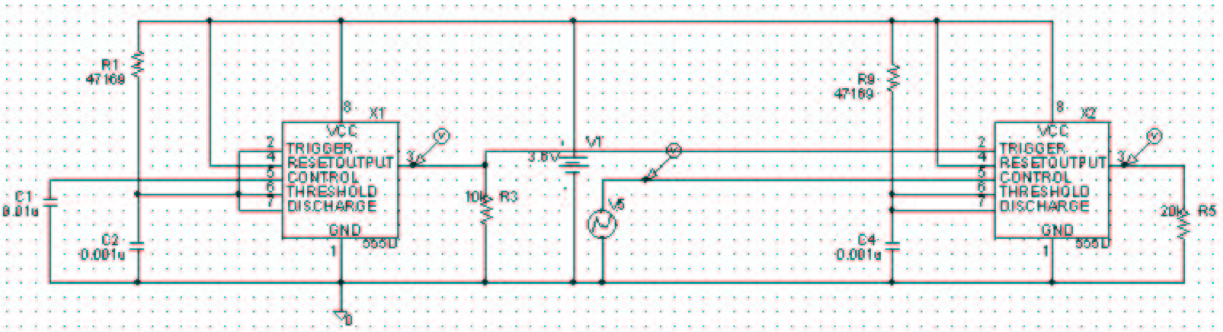


**Figure 10.15:** Diagram of one PW-modulator with its own oscillator. The voltage source V5 simulates the analog voltage from the MCU and is also visible in figure 10.16

In figure 10.16 the output of the PW-modulator and oscillator system is visible. As can be seen from the figure a voltage sweep from 0 to 3.6 V will produce an output PW of 0 to 100%. The top graph show the PWM output, this is the signal that is to control the converter. The middle graph show the control voltage, in this case a voltage that rises from 0 to 3.6 V. The bottom graph show the output of the oscillator, the duty cycle is close to 100%.

| | Frequency | $R_a$ | C |
|---|---|---|---|
| PW oscillator 1 | 50 kHz | 18214 $\Omega$ | 0.001$\mu$F |
| PW oscillator 2 | 21 kHz | 43715 $\Omega$ | 0.001$\mu$F |

**Table 10.3:** The component values

## 10.5 DAC Interface

Before the modulation input on the timers a hold capacitor is used. Then the PWM will still run because of the hold capacitor. This interface between the MCU and timer circuit will be presented here.

An electric switch (N-MOS transistor) after the DAC will be used to cut the connection between the hold capacitor and the MCU. In this way the capacitor will not discharge through MCU and then maybe destroy it! After the hold capacitor an operational amplifier (Op Amp) is used as a buffer between the capacitor and timer circuit. The output from the timer will go to the gate-driver.

To control the transistor a voltage-doubler (V-D) circuit is used directly on the transistor gate. The transistor will be open when the gate is logic high and be off when low. If the MCU shuts down, the ports will go low. To be sure that the P3.1 is low, a pull-down resistor is used. The hold capacitor (C) is chosen from the point of view that the MCU's DAC may maximall be loaded by 100 pF and to be on the safe side a 47 pF capacitor is used. The interface between the MCU and timer circuit can be seen in figure 10.17.

[4] Please note that the astable mode in the application note is not identical to the astable mode used in this project

[5] Two oscillators and two PW-modulators

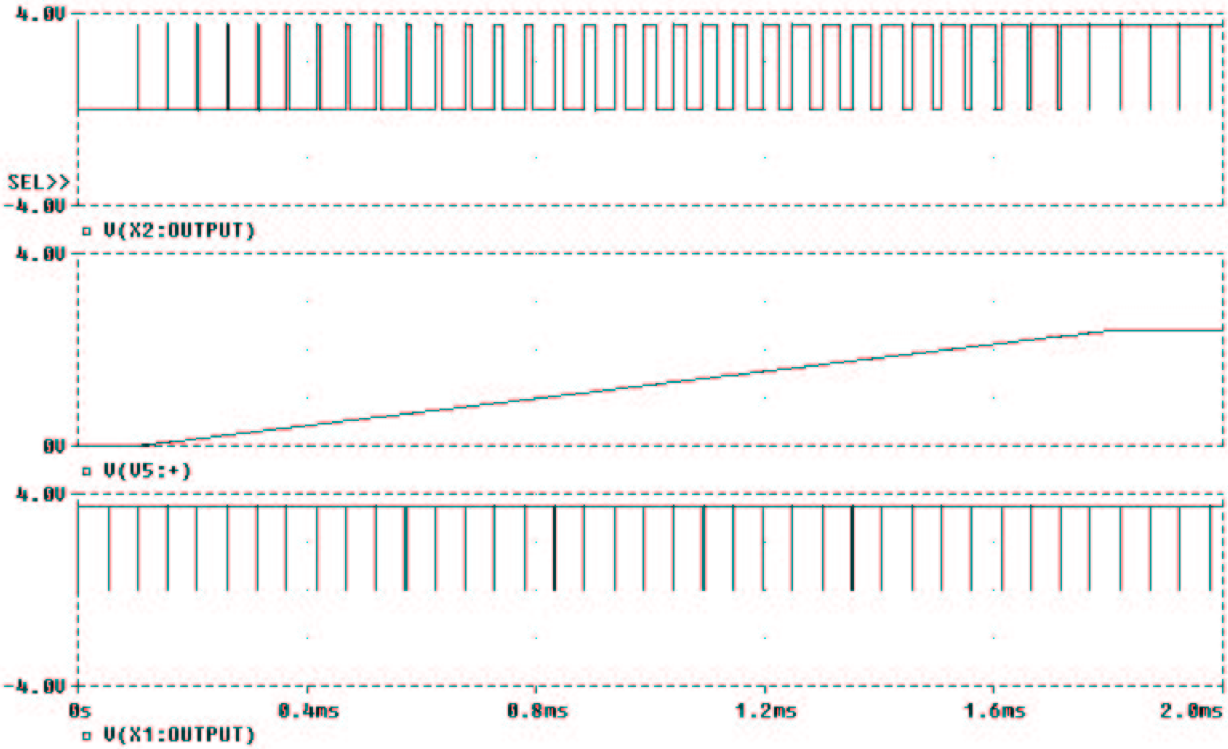


**Figure 10.16:** Simulating a PW-modulator and an oscillator. The top graph show the PWM output, this is the signal that is to control the converter. The middle graph show the control voltage, in this case a voltage that raises from 0 to 3.6 V. The bottom graph show the output of the oscillator, the duty cycle is close to 100%

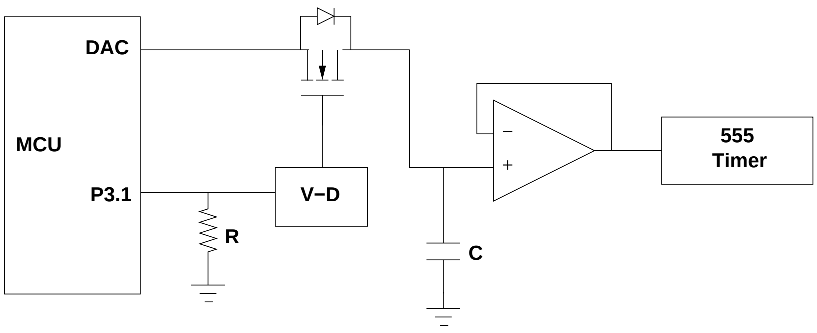


**Figure 10.17:** The DAC/Timer Interface

This circuit above, is the same circuit used for generating both the PWM-signals. The pin (P3.1) used to control the V-D will be used for both the circuits. The voltage doubler that is chosen is the type ICL7660ESA. The N-MOS transistor that will be used is the type HUF76145P3. The Op Amp that will be used is OP281GP.

## 10.6 Interface with Peripheral Devices

The MCU controls the over current protection circuit (OCPC) by 8 I/O signals. There are 4 input and 4 output signals, which are connected to the microcontrollers port 0 via two latches (latch 1 & 2), this is illustrated in figure 10.18.

Furthermore, the MCU is controlling the multiplexers via latch 1. The electric switch's (SW A & B) previously described in section 10.5 are controlled by P3.1 and the boot-selector port for OBC is P3.0. To activate the two latches, port 3's $\overline{WR}$ and $\overline{RD}$ are used.

To configure port 3 as input or output; Port 3 pins that have 1's written to them are pulled high by the internal pull-up's and in this state they can be used as inputs.

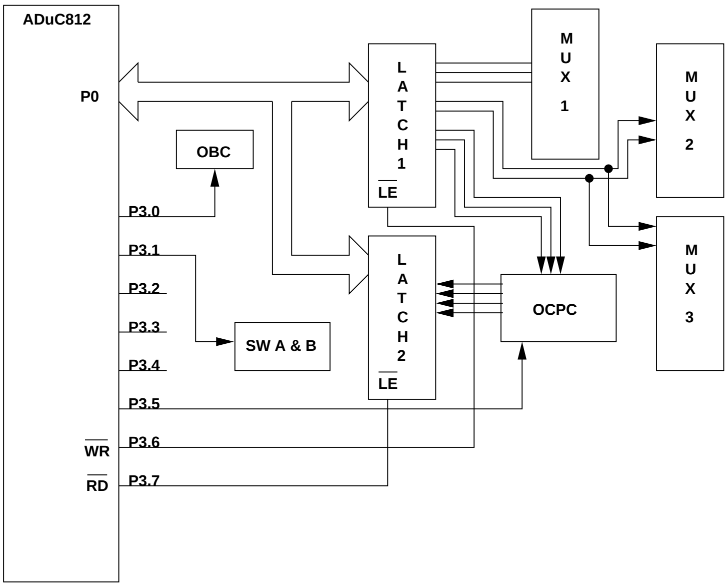


**Figure 10.18:** Interface with Peripheral Devices

| Port/Pin | Interface |
|---|---|
| P0.0 - P0.7 | Data for latches |
| P3.0 | OBC boot select |
| P3.1 | PWM Switches A& B |
| P3.5 | OCPC |
| P3.6 | External Latch 1 |
| P3.7 | External Latch 2 |
| Pin 9 | DAC0 for MPPTC |
| Pin 10 | DAC1 for PCC |
| Pin 26 | SCLOCK for $I^2C$ |
| Pin 27 | SDATA for $I^2C$ |

**Table 10.4:** Interfaces from ADuC812

## 10.7 Implementation

In this section the implementation will be described. The software that have been used to develop the schematic and PCB[6], is Protel 99. Furthermore, the total current consumption of the devices used will be estimated.

In this project the time has been short and then not been possible to implement all the hardware that is designed. The multiplexers and the external memory are not implemented. The multiplexers have not been implemented because it is not needed to measure the temperatures, this is only housekeeping information and have nothing to do with the main PSU. The external memory was not implemented according to that the external memory is only needed in space to protect the data, see section 10.2.3.

All in all there are four schematics and PCB's that have been developed, the schematics they can be found in appendix L on page 227. This is done to minimize the risk of fails in implementation. The reason for developing four PCB instead of only one, was that if there should be a mistake, it would be quick to make a new PCB.

The PCB's that have been developed are the following:

- MCU
- RS232
- PWM
- Converter/Measurement

[6] Printed Circuit Board

The MCU board is only consisting of the MCU, some bypass capacitors, an crystal, a reset circuit and some pin-headers for interfacing peripherals. Furthermore, this board includes the start up circuit.

The RS232 is the board used for downloading the program code. On the RS232 board the MAX232 circuit converts TTL voltages into RS232 voltages, thus data can be sent over the serial connection. For downloading the program code into the MCU's flash memory, the $\overline{PSEN}$ pin is pulled low. When this is done the MCU will go into serial-download mode and the program code can be transfered.

The PWM board is the one, that the two DAC on the MCU board must interface with. The timers on the board are the ones that generated the PWM signals, these are the outputs from this board.

The Converter/Measurement board is where the two converters are placed. Furthermore, on this board the components for measurement points are placed. This is the board that must interface with the DAC/Timer board, as well the MCU board.

**The total current consumption**

The currents used by the digital hardware components are listed below in table 10.5.

| Devices | Current consumption |
|---|---|
| ADuC812 | 15.8 mA |
| Reset | 45 $\mu$A |
| 2 x Timers | 180 $\mu$A |
| 3 x OpAmps (Op291) | 1200 $\mu$A |
| OpAmps (Op281) | 3 $\mu$A |
| Voltage Doubler | 80 $\mu$A |
| Total | 17.3 mA |

**Table 10.5:** The total current consumption used by the digital hardware

In the table the current used by the microcontroller is estimated as followed; the microcontrollers consumption (in normal mode) is dependent on the master clock frequency (MCLKIN) as it can be seen in the equation below [ADuC812, 2001]:

$$I_{DD} = (0.8 \ \ nA \cdot MCLKIN) + 3 \ \ mA \Rightarrow I_{DD} = (0.8 \ \ nA \cdot 16 \ \ MHz) + 3 \ \ mA = 15.8 \ \ mA$$

The current consumption of the start up circuit is now determined. In the table 10.7 above the total current used be the digital hardware components is 17.3 mA. Two Schottky diodes are used in the start up circuit, see section 10.2.1. With a forward voltage of 0.3 V and the current of 17.3 mA used by the components, this will give a loss of 10.38 mW in the two diodes. The Zener diode with a zener voltage of 3.9 V will give a loss of 67.5 mW. The total power used be digital hardware is determined to be:

$$10.38 \ \ mW + 67.5 \ \ mW + (17.3 \ \ mA \cdot 3.6 \ \ V) = 140.5 \ \ mW$$

## 10.8 Test Results

The module tests are made according to the test specification J.4 on page 218. The systems that have been tested are the listed below:

- Test of power supply select circuitry
- Test of power on reset circuitry
- Test of core MCU implementation
- Test of PW-modulators
- Test of measurement points

### Test of Power Supply Select Circuitry

The test was done by applying solar cell voltage of 4 V and it was controlled that the load voltage was 3.6 V. Then the battery voltage of 6 V was applied and it showed that the load voltage still was 3.6 V while the current now was drawn from the battery. Last the 5 V bus was applied and the current was drawn from there. Therefore, the on-board circuitry is capable of using its power from different power sources. This means that the Power Supply Select Circuitry is working.

### Test of Power on Reset Circuitry

The test was performed in order to test that the POR works. The test was performed and the result is in figure 10.19.

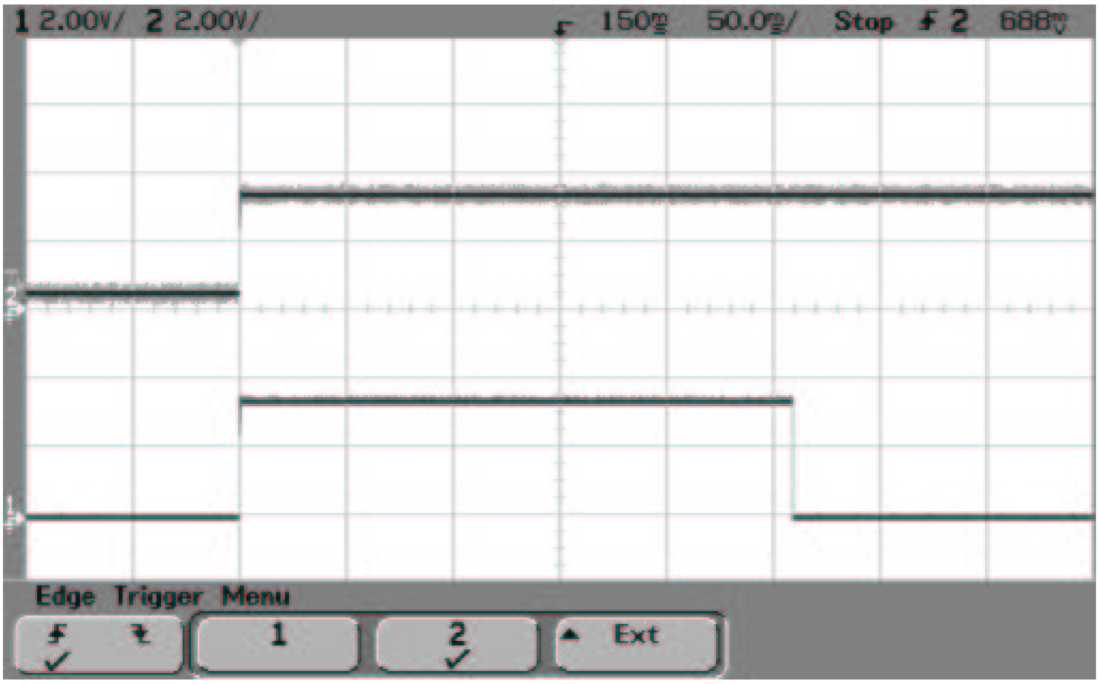


**Figure 10.19:** Test of the POR. The top (channel 2) showing the $V_{cc}$ is the trigger and show when the power is added. The bottom (channel 1) show the reset pulse

### Test of Core MCU Implementation

In order to test the MCU core and to ensure the development tools and code downloader is working, a test program was made that counts up on port 0. Using a oscilloscope is was clear that the MCU performed the task without problems and this showed that the MCU, development tool and the code downloader worked.

### Test of PW-Modulators

The first test is done to ensure that the duty cycle can be varied from 0 to 100%. The test program made a ramp on the DAC output from 0 to 2.5 V. The resulting pulse width and DAC voltage is shown in figure 10.20. Figure 10.21 show the same signal, but there is zoomed in on respectively on the maximum and minimum duty cycle.

The second test was done to ensure that the input correspond to the output and the input voltage and output dutycycle was measured. The result are shown in table 10.6. From the results it can be concluded that the

| Input [v] | Output [%] |
|---|---|
| 0.48 | 0 |
| 1.12 | 36 |
| 1.44 | 48 |
| 1.76 | 64 |
| 2.4 | 97 |

**Table 10.6:** Test results for PW-modulaters, input vs. output

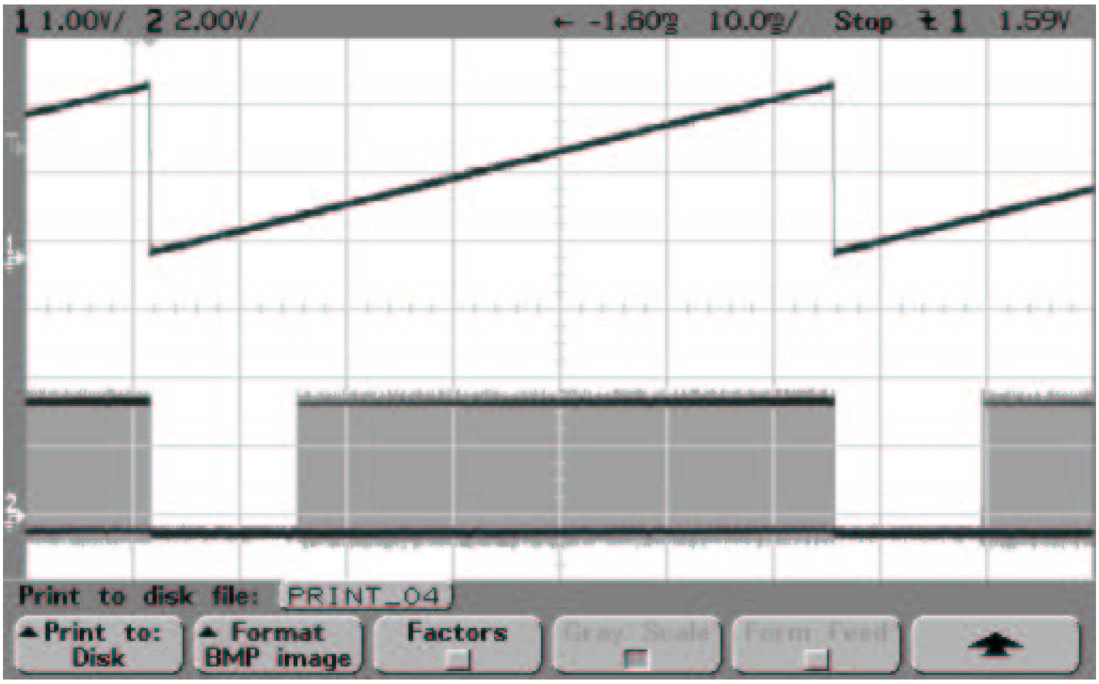


**Figure 10.20:** PW-modulator test. Top showing the input voltage on the pulse width on the bottom

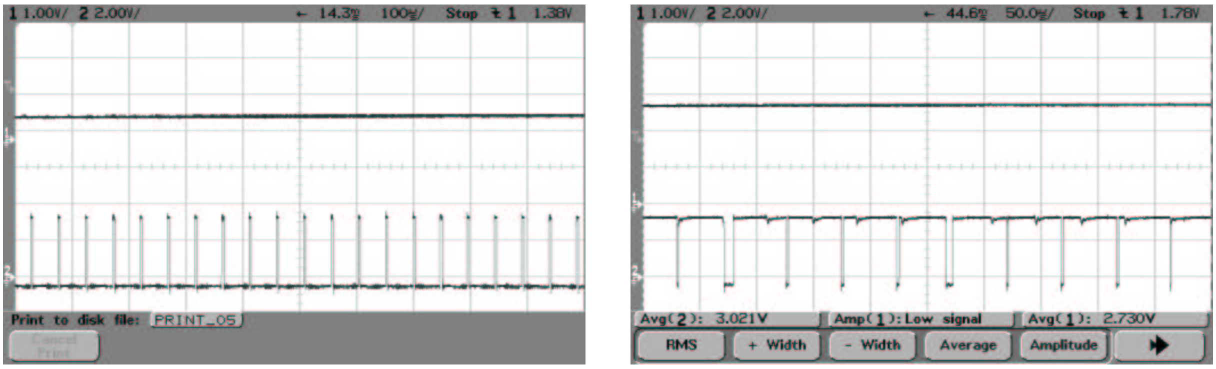


**Figure 10.21:** Zoom on the the high and low end duty cycle. Top showing the input voltage on the pulse width on the bottom the picture is the dutycycle

PW-modulators are working satisfactory.

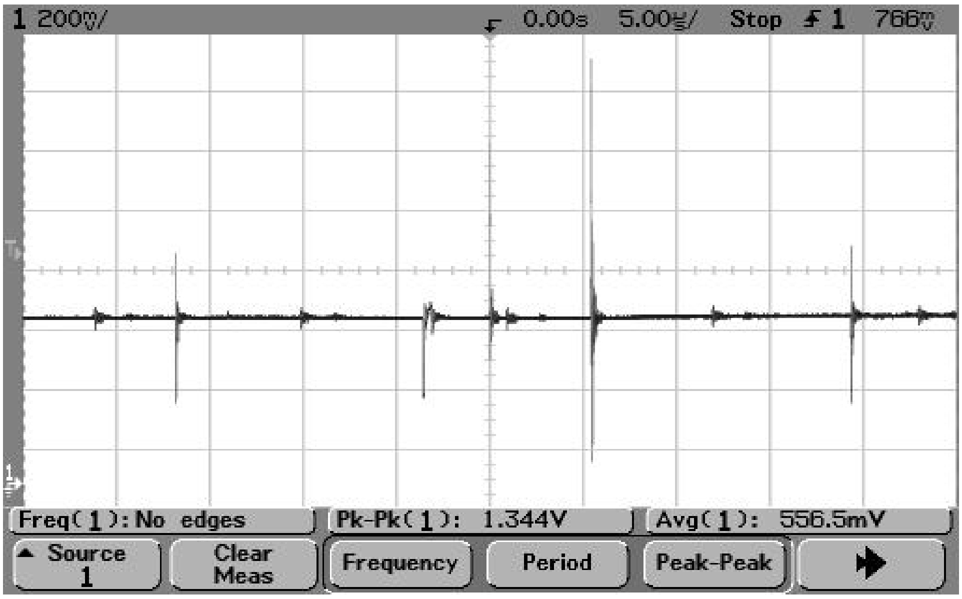
The third test was to find out how long the hold circuits was capable of maintaining the dutycycle. The test however was abandoned because the holdcircuit showed that it was not capable of maintaining the dutycycle for longer than a few ms. Because of timepressure it was decided not to spend any time on the circuits because of their minor importance for the PSU.

### Test of Measurement Points

Only the five measurement points used for converter control (I7, V8, V9, I10, I11) have been implemented and they are therefore the only ones to be tested. Each of these points have been tested by measuring the voltages at the output terminals, as well as the value to be measured. All points gave outputs that are in correspondence with the design when measuring the average output signal. Maximum deviations on the output signals was about 10 mV corresponding to $16 \cdot U_{lsb}$. This occurred when the output of the Op-Amps were close to zero and it is on basis of this assumed to be caused by the not ideal rail-to-rail capabilities of the Op-Amp.

Noise from the switching transistors in the converters was, however, visible on all the output signal as shown for the V9 measurement on figure 10.22

As can be seen from the figure the noise is of quite a large amplitude, but as the area of the noise-spikes are quite small then the chance of getting a sample corrupted due to the switching noise is small. To enhance the quality of the signal more low-pass filters could be applied or the software that samples the points could filter



1 200mV/ 0.00s 5.00us/ Stop 1 766mV

Freq(1): No edges

Pk-Pk(1): 1.344V

Avg(1): 556.5mV

**Figure 10.22:** The visible converter switching noise on measurement point V9

away large changes from one sample to another by applying a discrete moving average filter.

Since the current measured by I7, i.e. the boost-converter output current, is smaller of magnitude than measurement points I10 and I11 (see table 10.1), it was chosen to change the amplification of I7 by a factor of 16 to 1000 in order to minimize the effects of the above described noise on this measurement point.

# Part III

# Software Design and Implementation

# Chapter 11
# Software Platform and Real-time Analysis

## 11.1 Introduction to the Software Part

The main purpose of the software that runs on the MCU in the PSU is to sample data from the physical system, process this data and apply the result back to the physical system in order to ensure that it operates within the specified limits.

If the software fails in this process or if it is not fast enough to keep up with the physical system then there is no guarantee that the output of the PSU is within the specified limits and in effect users may suffer damage due to e.g. a too high output voltage.

It is the purpose of this part of the report to develop a software solution that can meet the real-time requirements from the physical system and which is both reliable and able to handle faults gracefully, i.e. it must have guaranteed safe failure modes which do not endanger neither the PSU or the users of the PSU.

The software that is going to be developed is divided in two major parts: the software-platform and the applications. The platform consist of the runtime services that are offered to the application programs by the underlying hardware, the Run Time Support System (RTSS) of the programming language and the operating system. Examples of such services are: Exception handling or mutual exclusion. Application programs are the pieces of code that do the actual work that the programmer wants the system to perform.

## 11.2 Hardware Platform Analysis

This chapter will provide the platform and the overall design of the software solution for the PSU. The platform being the layer of software (RTSS, language and other routines) that lies between hardware and application programs (see figure 11.1)

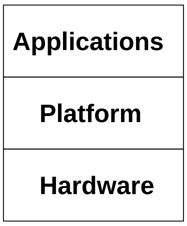


**Figure 11.1:** Levels of software

First the hardware environment will be analyzed in order to investigate how it effects the software platform. This analysis would not have been important if the applications were to be implemented on a system with large amount of processor and memory resources, but in this case the limited resources of the MCU is a major constraint and a good understanding of the hardware platform will help to keep the final software within these constraints.

Secondly a programming language is chosen and it will be analyzed how the RTSS of this language can help in development of the applications. Hereafter attention is brought to the problem domain where a real-time analysis is conducted in order to identify the applications and their real-time requirements.

Having found these requirements an overall system design which guarantees that the deadlines of the applications can be meet is developed and controlled for scheduability. Finally having analyzed the platform functionality provided by hardware and RTSS and the requirements of the application the platform can be extended in order to ease the development of the applications.

In the following the available resources of the hardware on which the application programs are to be implemented will be analyzed. Further, the hardware programming model will be described which is an overview of the interface between hardware and software.

### 11.2.1 Architecture Overview

The ADuC812 is build around a core implementation of the well known 8051 architecture developed by Intel. This architecture is a very typical 8 bit architecture with fixed length opcodes (8 bit) followed by one byte for data operands or two bytes for addressing operands. Further it uses the internal accumulator register as implicit operand for all algebraic and logical operations.

The memory is organized as a standard Harvard architecture having separate address-spaces for program and data memory. In addition the data-memory is further divided into two address spaces which are the internal and external data memory respectively. However there is no external RAM connected to the MCU in the PSU hardware.

The internal address space is further divided into three distinct regions with different addressing characteristics (see figure 11.2). The lower 128 bytes are addressable both by direct and indirect operations and the upper 128 bytes is divided in two section depending on addressing method. If indirect addressing is used then the addresses will work as general purpose RAM and if direct addressing is used then the address will provide access to the Special Function Registers (SFR) of the MCU. The SFR are registers that control the peripherals of the MCU, such as for example an internal timer.

To make things even more complicated the lower 128 byte of the internal RAM is divided into four register banks of 8 bytes each, a bit addressable area of 16 byte (128 bit) and a general purpose area of 80 byte. Clever use of the register banks can help boost performance of Interrupt Service Routines (ISR) functions since a dedicated register bank for an ISR minimizes the amount of stack-saving that must be performed when entering the ISR. The bit addressable area allow efficient storage of binary values. While the lower 128 byte memory is organized as just described it can also be addressed as ordinary general purpose memory.

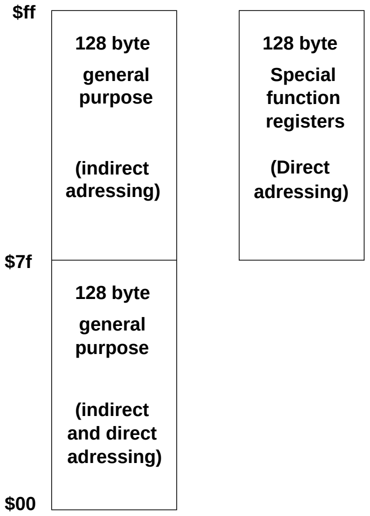


**Figure 11.2:** Memory map of internal memory

The 8051 core supports both interrupts triggered from peripherals implemented in the MCU and interrupts from external sources. Each machine cycle the SFR registers of the interrupt sources are polled in order to determine if interrupt processing should commence during the next machine cycle.

### 11.2.2 Processing Power

The MCU is runs at 16 MHz and since the fastest type of instruction the MCU can perform takes 12 oscillator periods the processor can provide a maximum of 16 MHz/12=1.33 Million Instruction Per Second (MIPS). If it is assumed that the total sample rate of the system is 50kHz then this gives 1.33 MIPS/50 kHz=26.67 instructions per sample . Remembering that the MCU is a 8-bit architecture and therefore only can manipulate 8 bits at a time it is clear that the processing power of the system is very limited. From here on 12 oscillator cycles will be called a machine cycle.

Looking at the instruction execution time table of the processor it can be seen that generally all instructions that performs algebraic or logical operations in internal registers executes in 1 machine cycle and all instruction that perform branching or data movement generally uses two machine cycles. Therefore, it is evident that in order to squeeze as many instructions out of the processor as possible the code should be well optimized with regard to the number of memory accesses. Specifically since indirect memory access under all circumstances will take at least 2 machine cycles, a stack mechanism for parameter passing will slow down the system

### 11.2.3 Memory

In addition to the memory already described in section 11.2.1 which is standard for the 8051 architecture, the ADuC812 has 640 bytes of flash-RAM that can be programmed from software, but since this memory is addressed through the SFR registers it is very tedious and slow to work with and it is only specified to withstand 10,000 write cycles, so when the satellite is specified to survive a year then it is only possible to write new values into the flash-RAM 27 times per day. Because of this it is not possible to use the flash-RAM as real RAM.

This leaves the total amount of available memory at 256 bytes, which is not a staggering amount. It is because of this constraint very important that the memory is utilized thoughtfully by choosing the smallest possible data types to describe data and by structuring the code to allow overlaying of the RAM, which means that the same area of RAM can be used to hold temporary variables of different function at interleaved time periods.

Another memory resource is the 32 kb of external ROM (as described in section 10.2.3 on page 109). This memory can be used to hold tables of data that can help both to speed up calculations and help to save memory for temporary variables that would normally have been used during the calculation.

### 11.2.4 Peripherals

The ADuC812 has a set of peripherals that will come in handy for the PSU software. A short description of each of these peripherals and a description of why they are relevant is given in the following.

**Analog-to-Digital-Converter (ADC)**

The ADC of the ADuC812 is a very capable 12-bit ADC that is able to multiplex between 9 analog inputs and sample a signal in less than 5 $\mu s$ with an accuracy of 0.5 LSB[1]. The ADC can be controlled by either polling or interrupts, or by working completely independently with the use of DMA[2]. However since the latter option requires external data memory it cannot be used in this implementation.

**Digital-to-Analog-Converter (DAC)**

Two 12 bit DACs are implemented in the MCU and they will be used to output an analog reference voltage to the two external pulse width modulator circuits. The DACs can be configured to output either a voltage between zero and the MCU supply voltage or a voltage between zero and an internal reference voltage of 2.5 V. In this project we will make use of the latter option because it provides the best accuracy of the output signal.

**I$^2$C Interface**

The MCU contains a software assisted implementation of the I2C protocol. This means that it is able to communicate as an I2C device, but part of the protocol, which is normally implemented in hardware, is in this MCU not implemented but can be emulated by software. There are also some of the features of the I2C bus which are not implemented in hardware and which cannot be software emulated.

**Timers**

Two general purpose 16 bit timers are available on the MCU. One of these can for example be used in conjunction with performance of the OBC boot algorithm as described in section 3.3.3.1 on page 27.

**Watch-Dog-Timer (WDT)**

The WDT is a device that will reset the MCU after a specified amount of time if the software has not set ones to two SFR-bits in a specified order. This is used to detect if the software is working properly, since the WDT will assume that the system is not working if the bits are not set.

## 11.3 Software Platform Analysis

In the following the programming language to be used will be chosen and it will be analyzed in order to find out what special features of that language and its RTSS can be used to help develop the applications and what features it lack that must be implemented as part of the platform.

### 11.3.1 Programming Languages

Languages such as Java or Ada are both good languages for real-time applications and have very extensive RTSS' that are able to handle a lot of issues that need to be addressed in most real-time software solutions. The following will provide a few examples of such issues.

If two applications or an application and an ISR need to access data concurrently i.e. pseudo-simultaneously then care must be taken to ensure that only one of the accesses is allowed to perform a read-and-modify operation at a given time. This access control is usually managed by semaphore data-types or in the Java language objects can be marked with the `synchronized` attribute which tells the RTSS to control concurrent access to the object.

When controlling a physical system by software all variables that reflect the physical system will have a well defined range of values and it can be assumed that the system is not working properly if this range is exceeded. In a language such as ADA the programmer can specify the range of each such variable and the ADA-RTSS will then control if the variable is within this range every time it is used and if it is not then it will raise an exception which will propagate up through the hierarchy of the program until an appropriate handler is found.

Having praised the advanced features of the mentioned languages it must be said that it is not feasible to use the languages with the ADuC812, because there is not processing power and memory resources enough to support these languages. This leaves two viable options assembler or the C-language.

While well written assembler code will execute faster than any compiled code it requires much of the application programmer, since the responsibility of all memory management rests with him or her and since it

[1] LSB: Least Significant Bit

[2] DMA: Direct Memory Access

is difficult to express abstract functionality as low level code. Therefore coding in assembler will be very error prone, specially if more developers develop the code concurrently. However it may be beneficial to use assembler in specific cases to boost performance of key algorithms.

Compared to assembler the C-language relieves the programmer by taking care of memory management and it allows the programmer to concentrate on the program at a more abstract level than what is possible in assembler. If however one compares C to languages such as Java or ADA, C does not boost a comprehensive RTSS that implements the previously discussed features of e.g. access control and range checking. This means that if such features are needed then they must be implemented by the developers as part of the software-platform.

**Choice of Language**

Based on the above discussion it has been chosen to use the C-language as the software development platform for this project. The lacking features of this language will need to be implemented and compiled with the application software. This functionality together with what functionalities are in the language itself is what we call the software-platform.

As C-compiler the Small-Device-C-Compiler (SDCC) has been chosen. It is an open source compiler package[3] that includes debugger and simulator. In addition to being (nearly) ANSI-C compliant the compiler includes a set of features specifically suited for development of MCU based applications. The following subsections will describe some of the features and limitations of this compiler with regard to the performance of the software.

### 11.3.2 Mutual Exclusion

If for example an application program needs to read and modify data that can also be accessed by an ISR then this situation may lead to a race-condition i.e. is that the total outcome of the interactions depend on the ordering of the interleaved accesses to the data. This is not desirable if one wants to have a predictable system. Therefore such data needs to be protected such that there is only one possible outcome of the interactions.

One possible method for doing this is to use semaphores or monitors, which is structures that controls access to shared resources, to restrict access to the data while they are being manipulated by one entity. Another solution is to disable interrupts while an application manipulates critical data. This makes sure that only the application can manipulate the data and if an interrupt is triggered in this interval then it will be pending until interrupts are turned on again and then executed at this time.

The SDCC compiler has a mechanism that implements the latter access control method. If a function is declared as **critical** then the compiler will generate code to disable interrupts upon entry to the function and enable them again upon exit. This feature is used as illustrated by the following code example:

```
short int danger() critical
{  /*
   Critical code here
   */
}
```

### 11.3.3 Reentrancy

It is a requirement of the ANSI-C standard that all generated code that does not contain any static declarations shall be reentrant. This means that all parameters and local variable must be allocated on the stack each time the function is called. For a small 8051 this however is a major performance constraint since this will involve a lot of manipulation of the stack, which is only needed for functions that are called recursively or is called from two concurrently executing functions.

Therefore the SDCC compiler by default breaks from the ANSI-C standard and compiles all code as non-reentrant. If however reentrant code is needed then the keyword **reentrant** should be used in the same manner as **critical** in the above code-example.

### 11.3.4 Interrupt Service Routines

Functions that are to be used as ISR functions must be declared with a special syntax:

[3] Available from: **http//:sdcc.sourceforge.net/**

```
short int ISR_adc() interrupt 5 using 1
{  /*
   ISR code here
   */
}
```

The ”interrupt” attribute tells the compiler to place the address of the function on the corresponding interrupt vector in ROM in this case vector number 5. The ”using” attribute informs the compiler what register bank the compiler should assign variables to when executing this function. This allows ISR routines to execute faster, since the memory in the register banks can be addressed using direct addressing and since no entry code in the ISR needs to handle allocation of local variables. If no register bank is specified then the ISR function will allocate local variables as if it was a normal function.

### 11.3.5 Storage Classes

Because of the complex memory mapping of the 8051 core the SDCC compiler includes a set of non ANSI-C storage classes that helps to utilize the memory of the MCU elaborately. The following two storage classes are of special interest for the PSU hardware configuration.

The storage class **idata** tells the compiler to store the declared variable in the higher 128 of the internal RAM. This mean that the variable will be placed in the area of RAM that is only addressable by indirect addressing. It will therefore be beneficial for performance to put all variable that is only seldomly addressed in this area and thus leave room in the direct addressable memory area for variables that are often used. A **data** storage class is also defined which tells the compiler explicit to store the declared variable in direct addressable RAM. This is the default behavior.

The third important storage class is the **bit** storage class that uses the special bit addressable area of RAM to store boolean values. This storage class helps to preserve memory space.

The final storage class to be considered is **code** which tell the compiler to locate the data in ROM. This allows for example tables of static data to be placed in ROM. Examples of declarations that use these storage classes are given in the following code example:

```
idata char data;  //Stores 8 bit in indirect addressable RAM
data char data;   //Stores 8 bit in direct addressable RAM
bit flag;         //Stores 1 bit in bit addressable RAM
code char data;   //Stores 8 bit in ROM
```

### 11.3.6 Calculations with Floats

The SDCC compiler is able to operate with 32 bit floating point numbers (IEEE single precision), but doing this will take up a lot of processing power and memory. Therefore in order to conserve as many resources as possible it will be beneficial to find ways to describe non integer variables with integer storage classes as small as possible. This involves accepting the variables to be described by fixed decimal numbers, but if the provided resolution is adequate this will pose no problem.

Combining this approach with the fact that in the binary number system it is very efficient to do calculations with powers of two, very calculation economic integer representations of non integer physical variables can be designed. If for example we wish to express 25.34 in a representation system as just described we can multiply it by a power of two (called the scale-factor); Using $2^8 = 256$ as an example scale-factor we obtain:

$$25.34 \cdot 256 = 6487.04$$

Now, accepting the round off error we end up with 6487 which can be described by 2 bytes rather than a floating point variable of 4 bytes.

Numbers described in the same system can be added and subtracted without problems, but if two numbers are multiplied it is necessary to down-scale the result with the original scale-factor, because no information on where the decimal point is, is available for the multiplication process. An example:

$$\begin{aligned} Decimal : 25.34^2 &= 642.12 \\ Fixedpoint : \frac{6487^2}{256} &= 642.12 \cdot 256 \simeq 164380 \end{aligned}$$

In the same manner numbers which are divided must be scaled by the scale-factor after division.

Using this method it is generally possible to outperform floating point computations if the truncation error is not critical and if all variables that takes part in the calculations can be described well enough using the same presentation.

## 11.4 Fault Tolerance Analysis

If the software designed here fails during execution when launched in space it may be responsible for the failure of the complete Cubesat mission. It is therefore very important that the software is designed such that software failures are either prevented or detected, so that the fault condition can be remedied. Three types of faults will be discussed (Quoted from [Burns and Wellings, 2000]):

**Transient faults** A transient fault starts at a particular time, remains in the system for some period and then disappears. Examples of such faults occur in hardware components which have an adverse reaction to some external interference, such as electrical fields or radioactivity. After the disturbance disappears so does the fault (although not necessarily the induced error).

**Permanent faults** Permanent faults start at a particular time and remain in the system until they are repaired; for example, a broken wire or a software design error.

**Intermittent faults** These are transient faults that occur from time to time. An example is a hardware component that is heat sensitive, it works for a time, stops working, cools down and starts to work again.

The following will discuss each of these failure types and what actions that can be taken to prevent and detect these faults. Special emphasis will be put on faults due to Single-Event-Upsets (SEU) as described in chapter 2 on page 21.

### 11.4.1 Transient Faults

Transient faults may occur from two sources; Due to faults in the software design or due to external events, a SEU for example. The software design faults should be prevented by complete white-box testing of the software during the development process. This means that all possible execution paths in the software design should be analyses using a simulator or an in-circuit-emulator.

If for example a SEU alters the value of a memory location that holds the value of a sample taken by the ADC sampler and this values is used by a control task then the control task will produce a faulty output for as long as the value from the sample has an effect on the output of the task. During this interval the system will exhibit a performance degradation in which the PSU-output may deviate from specified limits until the transient has expired.

For each task that is to be designed an analysis must be performed that examines the worst case behavior of the PSU due to a transient fault in that specific task. If this analysis shows that the worst case transient fault may jeopardize the safety of the system, e.g. batteries may be over-charged, then the task should be redesigned.

### 11.4.2 Permanent Faults

Again this fault condition can occur as a result of a software design error and this must be prevented by the white-box test already described.

In the case of a SEU what distinguishes a permanent fault from a transient fault is that the fault condition (the value in error) remains permanently in the system. A possible scenario for this type of fault could be a variable affected by a SEU that holds information on the current software state, e.g. a variable that specifies if an I2C data-transfer has been initiated or not. This fault condition may lead to severe errors, such as entering an infinite loop.

Two counter measures will be taken against this fault condition. The MCU watch-dog-timer (WDT) peripheral will be used to reset the MCU if the software fails to reset the WDT before the deadline. During the time when the MCU resets and reinitializes the users may experience a performance degradation because the input signal to the external pulse width modulators are not updated. However a performance degradation is better than a total system failure and must therefore be tolerated in case of a WDT-reset.

The second measure against this fault condition is to protect critical variables with error correcting codes, such as for example Hamming codes. This of course leads to a higher overhead due to the extra time it takes to handle these data. Another approach is not to correct the corrupted data, but simply detect the error and reset the MCU. This can for example be done by saving the same variable in two locations and then check that they are equal before use and if they are not then reset the MCU. The latter approach will be used here because of the limited processing power.

### 11.4.3 Intermittent Faults

As described in the definition of this fault condition, it results in transient faults and will therefore be handled accordingly. However in order to prevent theses fault conditions thorough environmental testing should be performed in order to find any sources to Intermittent Faults.

## 11.5 Real-Time Analysis

In this section attention will be turned to the problem domain in order to analyze how the application software of the MCU can be split into a number of software entities, which each has unique functionality and resource requirements.

This analysis is performed partly on basis of the COBRA method (As described in [Gomaa, 1999]) and partly on general experience from other programming projects. In the first step an environmental model is developed in which inputs and outputs from the system is specified. Then the internal software-structure of the system is designed and this structure is then simplified into a set of scheduable tasks, and resource and scheduling requirements for these tasks are derived.

In order to perform such an analysis one must know what functionality the system must provide. The following list specifies what functionality the PSU must provide. If nothing else is stated these requirements are derived from the ”Requirements Specification” (see chapter 3 on page 23).

1. Implement the designed control algorithms for the converter (MPPT, PCC inner loop & PCC outer loop). Derived from the ”Controller Design” chapter ( 9 on page 81)

2. Collect housekeeping information.

3. Receive commands and communicate data through the I2C bus.

4. Handle the load protection circuits. Derived from ”Over Current Protection” (see chapter 8 on page 71).

5. Implement the OBC boot selector port.

6. Implement the Watch-Dog-Timer (WDT)

### 11.5.1 Environmental Model

Looking at the software for the PSU as one whole entity a system context diagram can be derived. This diagram, which for the PSU is depicted in figure 11.3, identifies all interfaces of the system by showing all data flows from and to the system and assigning terminators to those data flows. Terminators are hardware devices or other software systems.

From the figure five interfaces of the software can be identified. Shortly described these interfaces are:

**Physical System** From the physical system, mainly the converters and the distribution circuitry, a number of values are measured, which is used both for control of the two converters and for housekeeping information.

**PWM** The pulse width modulators which controls the duty cycles of the two converters receives the reference duty-cycle from the DACs in the MCU which in turn is calculated in the software.

**Protection Circuits** The protection circuits have two data-ports for reading and writing respectively. In order to change state of the circuits the software must write to the protection circuit write port and in order to see if a user has been turned on or off due to a fault-condition the read port must be polled.

**I2C-Bus** The I2C-bus is the main communication interface to other subsystems of the satellite and most important to the data handling software on the OBC. Through the I2C bus the PSU will receive commands to turn on or off users and the PSU sends its housekeeping data through this interface.

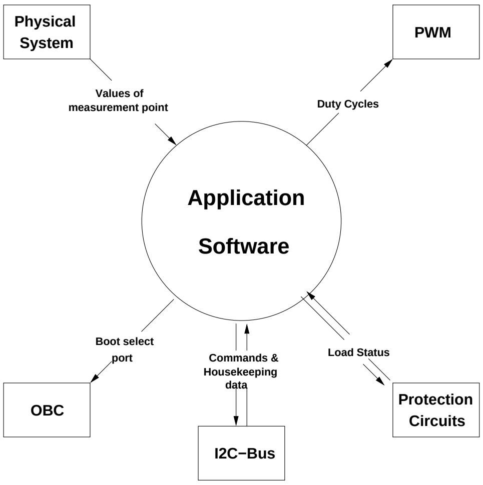


**Figure 11.3:** System Context Diagram of the Software for PSU

**OBC** The interface to the OBC is to control the boot-mode of the OBC, such that the OBC has a reliable way to determine if it should boot from PROM or EEPROM.

Having clarified the interfaces between the system and the environment it is now possible to begin to decompose the system (the circle in the diagram) into a number of smaller subsystems that each implements a well defined part of the required functionality of the software.

### 11.5.2 Structural Decomposition

At this point it is the goal of the RT-analysis to decompose the system into a number of smaller tasks. In this context a task will be defined as a piece of software that either needs to be scheduled or activated by asynchronous signals in order to operate.

Two criteria are used for this decomposition; the first criteria is composition with regard to functionality and the second criteria is that two subsystems must not access the same external interface. The latter criteria is to ensure that there will be no problems with mutual exclusion at the hardware level.

The decomposition can be seen in figure 11.4 which is an extension of the system context diagram. Outside the circle the information items from the system context diagram is shown and the boundary of the circle is populated by hardware driver tasks which are responsible for the interfacing between the exterior world and the software. Inside the circle the tasks that provide the software functionality, are shown. Finally all communication flow between the different tasks is shown, as well as the shared object ”PCC Current Reference” which is used by both the inner and outer loop of the PCC control algorithms.

#### Simplification

Having decomposed the system into a number of entities it is now desirable to simplify the design as much as possible in order to allow for an efficient implementation of the design. At the circle boundary all hardware driver tasks are depicted. Some of these tasks may not necessarily need to be implemented as a self contained piece of software. Both port-drivers are only used by one other task and it will therefore not be a problem to include this functionality into the ”Command and Control” task and thereby simplify the design.

The same goes for the DAC-task even though it is shared by more tasks. This is because when the DAC-peripheral first is initiated, it is controlled solely by two registers in which the pulse width is written or read.

The ADC tasks cannot however in an easy way be combined with one or more tasks, since strong mutual exclusion is important due to the nature of the ADC hardware. One ADC conversion cycle starts by specifying the states of the multiplexers, then the ADC is programmed and finally when the conversion has been completed a result is returned. If one task was to have exclusive access to the ADC through an entire conversion cycle it would spend a lot of time doing busy waiting on the conversion, i.e. it would block the CPU for useful usage while the conversion is performed. Therefore the ADC-task is needed in order to drive the ADC and multiplexers independently from the users of the data from the ADC.

The communication task handles the communication through the I2C bus in conjunction with the I2C hardware driver task. The nature of these tasks are that they execute aperiodically i.e. it cannot be predicted when

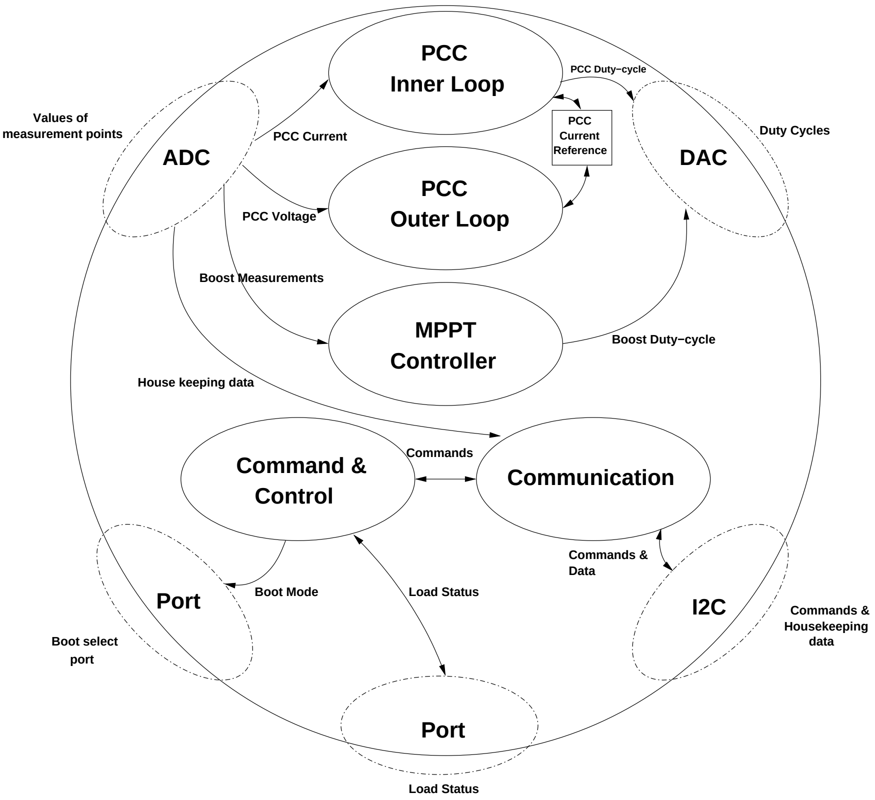


**Figure 11.4:** Diagram of the system after structural decomposition. Full lined ellipses are tasks, dotted eclipses are hardware drivers and squares are data objects

these tasks are active, but their actions are initiated on signals from interrupts or other tasks. This kind of activity is not desirable in a system that must be guaranteed to meet specifications in real-time, because of the difficulty of proving that these specifications can be meet under all circumstances. Therefore in order to make the system as predictable (and thereby scheduleable) as possible it is chosen to implement the I2C driver task as a very simple ISR which is only responsible for the transmission or reception of one byte at a time. The data-flow is then, through a FIFO-buffer, handled by the communication task, which will be scheduled periodically. Assuming overhead due to I2C interrupts to be small it is possible to predict system behavior in this implementation.

Now since the Command and Control task is very tightly coupled with the communication task due to the fact that all activities of the communication task is either initiated by the Command and Control task or the target of the data handled by the Communication task, it has been chosen to implement this functionality in one task.

### 11.5.3 Task Functionality and Requirements

The result of the above simplification is depicted in figure 11.5. In this figure the simplified hardware drivers are shown as terminators (following CODARTS notation) since they are reduced to sinks or sources of data-flow. The ADC-task is now treated as an ordinary task and the I2C task is depicted with a dotted line to indicate that it is a ISR and not a real scheduable task.

The following paragraphs will sum up the functionality and scheduling requirements of each of the tasks in the simplified diagram.

#### ADC Sampler

This task is responsible for sampling, including control of multiplexers, and storage of data from the physical system. This data is then accessed by the other tasks in the system and this access should be synchronized in such a way that the ADC task and the reading task do not encounter problems with mutual exclusion.

In table 10.1 on page 112 in the ”Digital Hardware” chapter the values that the ADC task must measure along with the frequency of each measurement are given. These numbers are the requirements that the ADC task must meet.

In order to minimize busy-waiting the ADC sampler should be implemented as an ISR that each time the

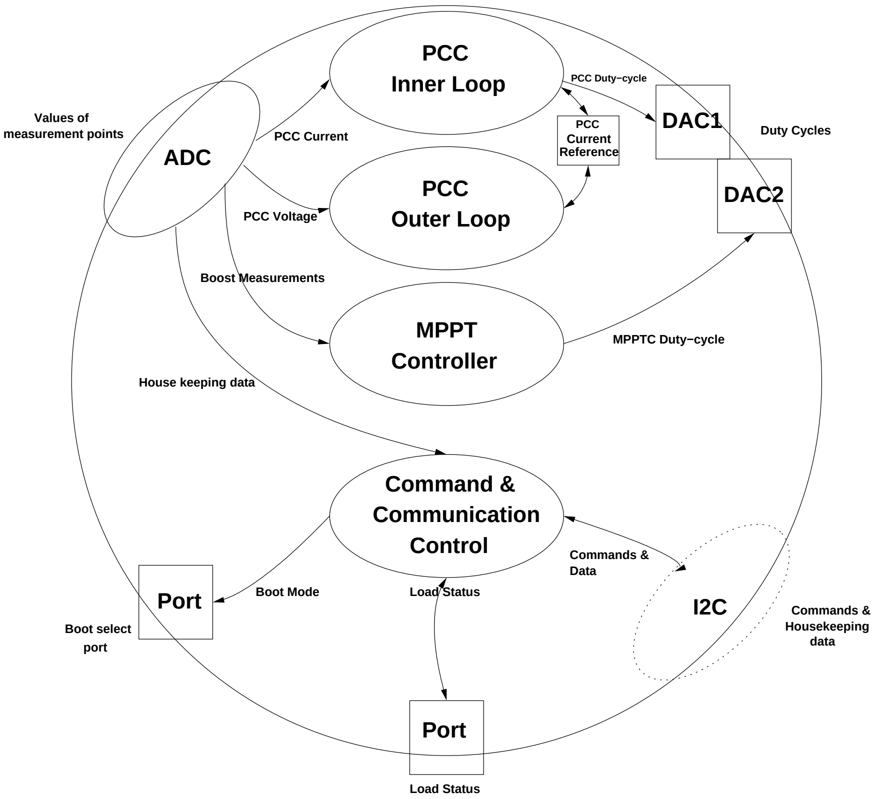


**Figure 11.5:** The Software system after simplification

ADC has finished a conversion cycle sets up multiplexers and ADC for the next conversion process. In effect the ADC sampler can be seen as an independent process sampling in accordance to some schedule.

### PCC Inner Loop

The PCC inner loop controls the current in the buck converter on basis of current measurements that it receives from the ADC task and the current duty-cycle of the converter. This task is periodic with a rate of 18 kHz (see section 9.5 on page 89) and has deadlines equal to the period time.

### PCC Outer Loop

The PCC outer loop controls the voltage of the buck converter output on basis of voltages measurements received from the ADC task, as well as the current duty-cycle. This task is periodic with a rate of 3 kHz (see section 9.5 on page 89) and has deadlines equal to the period time.

### MPPT Controller

The MPPT controller is responsible for finding the maximum power point of the solar cells. In order to do this the tasks uses measurements of the output current of the boost-converter. This task is periodic with a rate of 100 Hz (see section 9.5 on page 89) and has deadlines equal to the period time. If the battery voltage rises above a certain threshold the MPPT controller should limit the duty cycle such that the power supplied by the MPPT converter is close to what is consumed by the users on the main powerbus.

### Command, Control and Communication

This task (from here on abbreviated 3C) has the following functionality:

1. Administrate system information:
   - Collect Housekeeping information from ADC task
   - Poll load-status-port periodically
   - Hold status of boot-port
2. React to commands from OBC:

  - Turn loads on and off
  - Set value of boot-select-port
  - Send housekeeping information to OBC

3. Monitor the PSU:
   - Check that critical system variables are within sane limits
   - Determine if other task are working properly
   - If needed then reset the software

4. Implement the OBC external watch-dog

As can be seen from the list of functionality the 3C task is mainly responsible for system monitoring and control. The faster this is performed the better, but in order to have execution time enough for the control algorithms the rate at which the 3C task is scheduled should be limited. However the data-flow from the I2C ISR sets a minimum limit.

The I2C communication operated at 100 kbit/s and when each frame is 10 bits then the maximum frequency at which the 3C task must handle data is 10 kHz. The I2C protocol however is able to perform what is called clock stretching which allows the slave (The PSU in this case) not to respond in real time. Therefore the rate for the 3C task will be set to 1kHz, and in order to speed up the I2C communication such that the bus will not be occupied because of the PSU for long periods of time a send buffer of 10 bytes will be implemented between the 3C task and the I2C ISR. The choice of a send buffer rather than a receive buffer is that the peak I2C load with regard to the PSU communication is when housekeeping data is sent from PSU to DHS.

**I2C Interrupt Service Routine**

The I2C ISR works together with the 3C task in order to perform the I2C communication. This ISR is responsible for the transfer of data on the byte level. It transmits data-bytes from the above mentioned send buffer and it receives data from the I2C bus and forwards it to the 3C task for interpretation. This task is due to its ISR nature aperiodic and cannot be scheduled.

## 11.6 Overall Design Approach

Until now the tasks that must run on the system have been identified and requirements have been assigned each of the tasks. In the following an overall design approach that can implement these tasks are given.

### 11.6.1 Scheduling

All the tasks that have been identified share a set of characteristics; They all use data acquired by the ADC and they all have fixed periods and deadlines, i.e. they do not vary during runtime. Because of the latter similarity it is clear that a runtime scheduler, which schedules the tasks dynamically, is not needed and tasks should therefore be executed in order of a fixed schedule. Such a schedule is called a Fixed-Priority Schedule (FPS) and it can be derived on basis of task deadlines, periods and worst case computation time. Since the worst case computation time of each task is not known until they have been implemented this schedule will first be derived following the implementation chapters. Therefore, in the following discussion, it will simply be assumed that the schedule exists.

Since all processes uses data sampled by the ADC the ADC need to acquire data according to the same schedule as the schedule of the taskset. Further it must be assured that the data for a certain task is available before the tasks is executed. How this can be done is depicted in figure 11.6. In the figure the schedule consists of three tasks only for simplicity.

Both the ADC and the other tasks are scheduled by the same schedule, but they are displaced in time such that the ADC always samples the data prior to execution of the task that needs data. A problem is however that the execution times of each sampling and the corresponding task is not generally the same and therefore the ADC sampling process may either fall behind the tasks or it may run ahead of the tasks. If the ADC process falls behind the tasks it is a major problem because then the tasks would process data that is one sampling period old. In order to avoid this, synchronization is needed.

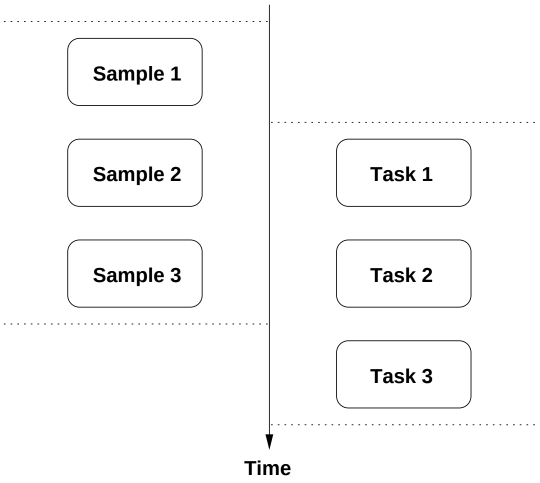


**Figure 11.6:** Overall scheduling approach

### 11.6.2 Synchronization

Synchronization will be assured by assigning a variable to both the ADC sampling process and the taskset. This variable hold the position in the schedule. Using this variable it is possible to ensure proper synchronization by executing the following test each time a new task is executed:

```
while ((ADC_POSITION <= TASK_POSITION)<1)
  /*do nothing*/ ;
```

This makes sure that the sampling process does not fall behind the execution of the taskset. Synchronization is also needed to ensure that the sampling process does not run to far ahead of the execution of the taskset, because if it was allowed to run freely it may overrun the taskset execution. This is done with a similar test that is performed in the ADC ISR code each time the sampling process restarts its schedule:

```
if ((ADC_POSITION - TASK_POSITION)<MAX_AHEAD)
  /* Repeat last Sample */;
```

By repeating the last sample the ADC ISR continues to be triggered and the test is therefore performed periodically without busy waiting. Theoretically the `MAX_AHEAD` variable can be as large as the numbers of entries in the schedule minus one. But this will lead to a very bad response time of the system because the data that is worked upon is at most one sampling period old, whereas if the two processes were completely synchronized the data would be as new as possible.

The method of synchronization described here is in literature [Burns and Wellings, 2000] said to utilize **Rendevouz**-points, because the two processes run freely most of the time, but at certain times (at the Rendevouz-point) they are re-synchronized.

### 11.6.3 Inter-Task Communication

Inter task communication will be performed with the fastest possible solution, which is shared memory. In order for this to succeed it is necessary that access to these memory locations is synchronized. For variables shared by a task and the ADC sampler, synchronization is already ensured as already described, and for variables shared by two ordinary tasks synchronization is also ensured because of the scheduling approach that does not allow preemption.

Finally for communication between the I2C interrupt and the 3C task synchronization must be ensured manually by using the **critical** keyword when shared data is accessed from the 3C task.

### 11.6.4 Summary

To sum up the results that has been established until now a top-down diagram of the software structure is given in figure 11.7. The items assembled under ”Platform” will be designed in the nest section, while those items collected under ”ISR” and ”Tasks” will be designed in the subsequent two chapters.

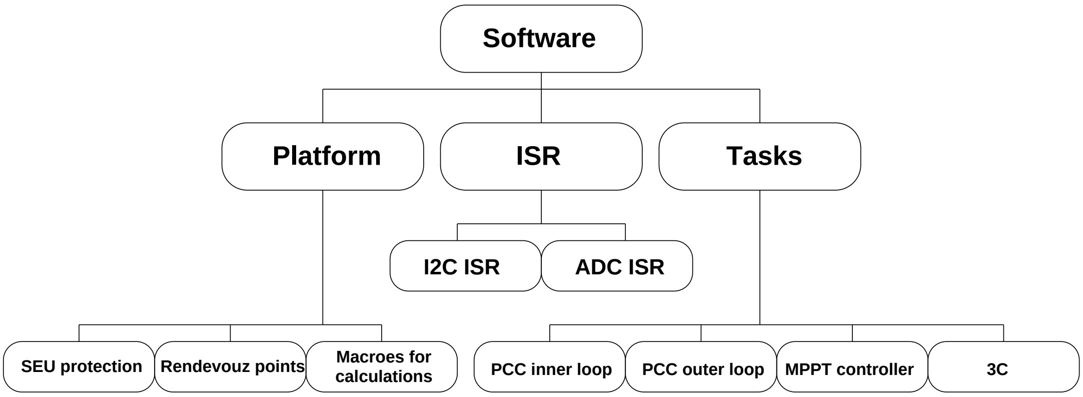


**Figure 11.7:** Top-down diagram of the software structure

## 11.7 Extending the Platform

Until now the analyses and design section has put forward the following issues that must be implemented to enhance the RTSS of the SDCC compiler:

- A fixed decimal point number scheme should be chosen

- Mechanisms to handle critical data must be made available

- Code for handling the rendevouz point shall be implemented

In general the above mechanisms will be implemented ad macros rather than functions. This is because the macros expands to code in ROM every time it is called and is thus faster to execute than if called as an ordinary function. As mentioned in subsection 11.2.3, ROM is the only resource that is plentiful on this system.

### 11.7.1 Rendevouz Points

In order to handle the synchronization between the ADC sampling process and the execution of the taskset, the solution proposed in 11.6.2 has been implemented as two macros. The first macro `TASK_SYNCRO` does busy waiting until the ADC is ahead of the taskset and it should be called prior to execution of every task.

The other macro `ADC_SYNCRO` evaluates if the ADC is to far ahead of the taskset, such that the ADC ISR can check if it should restart the schedule or repeat the last sample.

### 11.7.2 Number Format and Calculations

As described in subsection 11.3.6 a fixed decimal number scheme is needed for the calculations that control the converters. The physical range of the values sampled for these calculations are in the range 0-10V and 0-3A. Using a 16 bit word and letting the upper 6 bit describe integers and the lower 10 bits fractions, it is possible to describe numbers in the range 0 - 64.999 in steps of $\frac{1}{1024}$. An example of such a number is given in figure 11.8

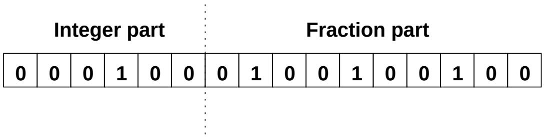


**Figure 11.8:** The chosen number representation scheme

AS explained in 11.3.6 the use of this representation scheme gives some problems with multiplication and division. In order not to put the burden of manually shifting the results of multiplications or divisions up or down respectively, four macros have been implemented to support these calculations.

The following code piece shows the declaration of the macro to multiply two numbers:

```
#define MULTIPLY(source1, source2, result)
        result=(source1*source2)>>10;
```

The macro takes three arguments, but it is possible to use the same argument as both source and result. The macro multiplies the two values and makes the appropriate bit-shifts. A similar DIVIDE macro has been defined for division.

### 11.7.3 SEU-protection Mechanisms

In order to protect critical data from SEUs two macros have been defined which operates on two special structures that is used to hold critical data. The declarations are given below:

```
/* Macro to write safe data */
#define SAVEPROTECTED(s_struct, value)
        s_struct.safeA=value;
        s_struct.safeB=value;

/* Macro to read safe data */
#define READPROTECTED(s_struct,result)
        if (s_struct.safeA!=s_struct.safeB) RESET;
        result=s_struct.safeA;

volatile struct Safe_char{
  char safeA;
  char safeB;
};

volatile struct Safe_int{
  short int safeA;
  short int safeB;
};
```

Depending on the size of the data that is needed to be protected one of the above structures are declared in a variable. This variable can then be accessed through the two macroes SAVEPROTECTED() and READPROTECTED() by specifyfying the structure as the first parameter and the value or result variable in the second parameter.

When data are read a comparison between the safeA variable is made with the redundant safeB variable. If they are in disagreement then the RESET macro is called and the MCU is restarted. The reason for declaring Safe_char and Safe_int as **volatile** is that otherwise the optimizer of the compiler may in some uses of the macro conclude that the programmer is outright stupid, because it does not give sense to compare two numbers which, by the structure of the code, can be proven always to be equal. So by declaring the structures as **volatile** it is certain that the optimizer does not eliminate the code that can detect a SEU.

# Chapter 12
# Design of Software Tasks

In this chapter the individual software tasks that are to run on the MCU is designed, except for the 3C task which is described in the next chapter together with the $I^2C$-communication.

For the tasks to be designed here the general approach will be to first describe functionality of the task then design the code and finally test the code and analyse worst-case execution time and consequences of permanent and transient faults. All code designed here is available from the enclosed CD-ROM[1]

## 12.1 ADC Sampler

It is the job of the ADC-sampler to sample data for the controller tasks and 3C task in the software. The ADC-sampler is designed to be implemented as an ISR that executes each time the ADC hardware has completed a conversion. Because of this the ADC-sampler is not a scheduable tasks as the others, but it can be seen upon as an autonomous process that executes in parallel with the normal tasks.

### 12.1.1 Design of ISR

The ADC-sampler is implemented as a ISR rutines that has execlusive acces to register bank 1 (as described in subsection 11.3.4 on page 132) in order to speed up the execution of the rutine.

The ISR is executed each time the ADC hardware triggers an interrupt due to a finished conversion cycle. When executed the ISR first check if it is still synchronized with the execution of the taskset. This is done using the `ADC_SYNCRO` macro as explained in subsection 11.6.2 on page 140 and if the ADC find it is synchronized then the last conversion cycle is repeated. Otherwise the ISR continues by first saving the converted value in memory and then by programming the internal and external multiplexers for the next sampling. Finally the conversion is started and a "return from interrupt is performed".

In order to keep track of which samples to take, a lookup-table, located in ROM, is used in conjunction with an index pointer which is the `ADC_POSITION` variable that is also used for synchronization between the ADC-sampler and execution of the taskset (described in subsection 11.6.2 on page 140).

The saved samples are saved in variables, which are accessible from other tasks, with names corresponding to the designator of the measurement point as described in table 10.1 on page 112.

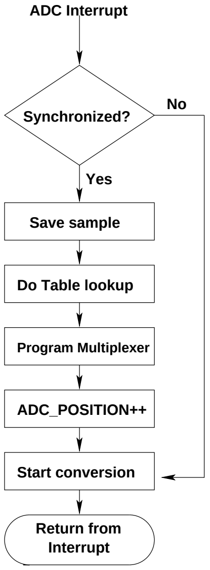


**Figure 12.1:** Flowchart of ADC ISR

### 12.1.2 Code Analysis

Using the assembler-code output from the SDCC compiler, the Worst Case Execution Time (WCET) has been determined by counting how many clock cycles is used by the instructions in the longest possible execution path of the code. The WCET for the ADC-sampler is 63 Machine Cycles.

The fault analysis showed that no SEU can bring the code to a permanent fault condition. The only variable that holds "state-information" is the `ADC_POSITON` variable, but if this is altered then either the synchronization test in the ADC-code itself or the synchronization macro performed by the taskset will bring the system back to ordinary operation and thus this only leads to a transient fault.

The above described possible transient fault together with the possibility of a SEU to affect any of the variables that hold sampled data make up the set of possible transient fault conditions, but all of these data will either be re-synchronized or re-sampled during one cycle of the schedule that holds the points to be measured. Therefore transient faults in the ADC-code is not judged to be critical, and no further measures are taken.

[1] /CDROM/sourcecode

## 12.2 PCC Outer Loop

The controller for the PCC outer loop was designed in section 9.6 on page 98 and it is given by the following function in the Z-domain:

$$D_o(z) = 0.05\frac{1-0.73z^{-1}}{1-z^{-1}} \tag{12.1}$$

This controller is to be executed at a frequency of 1.6 kHz.

### 12.2.1 Design

In order to implement this Z-transform on the MCU, it is rewritten in recursive form:

$$\begin{aligned} y_o[n]-y_o[n-1] &= 0.05x_o[n]-0.037x_o[n-1] \Leftrightarrow \\ y_o[n] &= 0.05x_o[n]-0.037x_o[n-1]+y_o[n-1] \end{aligned}$$

In order to make this equation consistent with the number format on the MCU it must be transformed to that number system (described in section 11.7.2 on page 141). This is done by first multiplying the constant coefficients with 1024 and then truncate the result. This gives:

$$y_o[n] = 51x_o[n]-37x_o[n-1]+1024y_o[n-1] \tag{12.2}$$

All variables ($y_o[n]$ and $x_o[n]$) are allready described by the correct numberformat in the MCU and are therefore not scaled in the transformation.

According to figure 9.6 on page 85 the input X to the system is the difference $V_e$ between the measured voltage and the constant reference voltage. The output is the reference current to the inner loop called $I_{ref}$. Equation 12.2 now yields:

$$I_{ref}[n] = 51\cdot(V_{ref}-V_o[n])-37\cdot(V_{ref}-V_o[n-1])+1024I_{ref}[n-1]$$

This equation can be implemented with the following code:

```
#define Vref 5*1024                    //Constant reference
short int Iref;                        //shared variable

void outerloop()
{ data static short int oldV9;    //Declare (n-1) variables
  data static short int oldIref;
  data short int Ve,Ve2;          //Intermediate variables

  Ve=Vref-V9;
  MULTIPLY(51,Ve,Ve);
  Ve2=Vref-oldV9;
  MULTIPLY(37,Ve2,Ve2);
  Iref=Ve-Ve2+oldIref;

  oldV9=V9;                       //Save (n-1) variables
  oldIref=Iref;
}
```

However this code is slow to execute when compiled to machine code. Therefore optimizations have been made to speed up the code by a factor of about two. The result of this can be seen below:

```
#define Vref 5*1024                    //Constant reference
short int Iref;                        //shared variable
data short int oldV9;                  //Declare (n-1) variables
data short int oldIref;

void outerloop()
```

```
{ Iref=(((Vref-V9)<<5)-(((Vref-oldV9)<<4)>>10+oldIref;

  oldV9=V9;                      //Save (n-1) variables
  oldIref=Iref;
}
```

The difference between the two pieces of code is that all multiplications have been replaced by bit shift operations which executes faster, but slightly less accurate. Additionally all intermediate variables have been omitted in order to reduce the number of memory accesses and in order to force the compiler to utilize registers more aggresively.

### 12.2.2 Code Analysis

The code analysis of the assembler output from the compiler showed that the WCET of this function is 111 machine cycles. Fault analysis showed that no variables hold state-information and the only fault condition is therefore transient faults. Since the routine uses values sampled at the previous execution the maximum period in which a transient fault is in effect in the software is two task periods. This equals $2 \cdot (3\ kHz)^{-1} = 667\ \mu s$.

The effect of this transient fault on the physical system can be seen as an impulse signal superimposed on the controller input signal, which in effect leads to an impulse reponse from the physical system with a settling time of <10ms (see section 9.5.2 on page 93).

## 12.3 PCC Inner Loop

The controller for the PCC inner loop was designed in chapter 9 on page 81 and it is given by the following function in the Z-domain:

$$D_i(z) = 0.6\frac{1 - 0.80z^{-1}}{1 - z^{-1}} \tag{12.3}$$

This controller is to be executed at a frequency of 18 kHz.

### 12.3.1 Design

As for the design of the outer PCC loop the Z-transform is rewritten in recursive from:

$$\begin{aligned} y_i[n] - y_i[n-1] &= 0.6x_i[n] - 0.48x_i[n-1] \Leftrightarrow \\ y_i[n] &= 0.6x_i[n] - 0.48x_i[n-1] + y_i[n-1] \end{aligned} \tag{12.4}$$

As for the outer loop the constants on the input in the transfer function for the outer loop has to be scaled with 1024. Extension of equation 12.4 by 1024 gives:

$$y_i[n] = 614x_i[n] - 491x_i[n-1] + 1024y_i[n-1] \tag{12.5}$$

According to figure 9.6 on page 85 the input to the system is the output from the summation-point that adds the inductor current with the negative output current and the output is the dutycycle. Equation 12.5 can now be rewritten as

$$d[n] = 614 \cdot (I_{ref}[n] + I_o[n] - I_L[n]) - 491 \cdot (I_{ref}[n-1] + I_o[n-1] - I_L[n-1]) + 1024d[n-1]$$

This equation has been implemented with code similar to that of the PCC outer loop.

### 12.3.2 Code Analysis

The code analysis of the assembler output from the compiler showed that the WCET of this function is 68 machine cycles. Fault analysis showed that no variables hold state-information and the only fault condition is transient faults. Since the routine uses values sampled at the previous execution the maximum period in which a transient fault is in effect is two task periods. This equals $2 \cdot (18\ kHz)^{-1} = 111\ \mu s$. This fault will have the same kind of effect on the physical system as described in subsection 12.2.2.

## 12.4 MPPT Algorithm

The MPPT algorithm will be designed according to the flowcharts in figure 9.10 on page 91 and 9.11 on page 92.

### 12.4.1 Design

The MPPT-algorithm is split in two parts:

- The P&O algorithm
- The over-charge algorithm

Both algorithms are to be carried out at a frequency of 100 Hz.

**P&O Algorithm**

The P&0 algorithm adjusts the dutycycle to reach the MPP. For this purpose the return current is measured and used as a measurement of the power since the output voltage is assumed constant during the short periods of time, that the MPPT lasts. This return current is the measurementpoint I7.

The algorithm was designed from figure 9.10 on page 91, and the implementation gave the following code:

```
DeltaD=DutyCycle-Dp;   //Calculate differences
DeltaI=I7-Ip;
Dp=DutyCycle;          //Save (n-1) values
Ip=I7;
if (DeltaI>0)
{ if (DeltaD>0) DutyCycle=DutyCycle+1;
   else DutyCycle=DutyCycle-1;
}
else
{ if (DeltaD>0) DutyCycle=DutyCycle-1;
   else  DutyCycle=D+1;
}
```

**Over-Charge Algorithm**

This algorithm has to make sure the battery voltage does not rise above 8.4 V, since a higher voltage may damage the batteries. Therefore the voltage across the batteries is measured, and this is the measurepoint V8.

This algorithm was designed according to figure 9.11 on page 92, which gave the following code:

```
if (V8<Vref)
{ if (OCP==1)
   { OCP=0;
     DutyCycle=409;
   }
}
else
{ OCP=1;
  if ((V8-Vp)<0) MaxDuty=MaxDuty*1024;
  else MaxDuty=MaxDuty*512;

  if (DutyCycle>MaxDuty) DutyCycle=MaxDuty;
}

Vp=V8;           //Remember last voltage (V8)
```

### 12.4.2 Code Analysis

Code analysis showed that the WCET of the above described two algorithms is 150 machine cycles. The code does hold one state variable `OCP`, but a SEU changing this variable does not lead to a permanent fault, because the algorithm given any initial state will perform correct power tracking from that state. Therefore only transient faults are possible.

Recovery from a transient fault will take maximmally 1024 runs of the algorithm, since this is the longest time it will take the algorithm to adjust the duty-cycle from one extreme to the other. This corresponds to about 11 seconds in which power-tracking does not work optimally. However since the battery voltage is checked each time the algorithm is run then a transient fault cannot damage the battery, but will only lead to a performance degradition, i.e. less obtained power for a period of maximally 11 seconds.

# Chapter 13
# Command, Control and Communication Task

## 13.1 Overview

In this chapter the command, control and communication (3C) task is described, first with a general introduction to the required functionality of the task. Then the interfaces of the 3C task are discussed. The functionality of the 3C task are defined in section 11.5.3 on page 137 as the following:

1. Administrate system information
2. React to commands from OBC
3. Monitor PSU
4. Function as OBC watch-dog timer

To do these jobs the 3C task is divided into two parts: a task and an ISR. There will not be distinguished specificly between 3C parts in the following because of the strong cohesiveness between that makes it difficult to divide them up and as explained in chapter 11 on page 129. They will be implemented as one task.

**Interfaces of the 3C-Task**

First the interfaces to the 3C task defined in section 11.5.3 on page 137 are discussed.

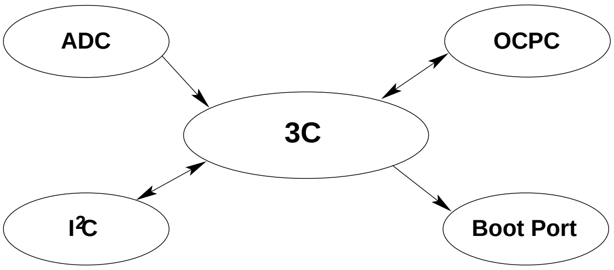


**Figure 13.1:** The four interfaces of the 3C-task

**ADC** This interface is not a directly interface to the ADC, but is really to the ADC task which handles the ADC. The ADC-task places the data in memory and the 3C task must read it there.

**OCPC** This interface is a bidirectional port on which the 3C task must read the status of the loads and turn loads on and off.

**I$^2$C** This interface will be serviced by the I$^2$C driver which will take care of the low level operation of the I$^2$C-bus. All data on the bus to the PSU are fetched by the driver and interpreted by the 3C task. Data sent from the 3C task are buffered and then sent by the driver to the OBC.

**Boot Port** This interface is the most simple of the four consisting only of a 2-wire port that can be toggled by commands received over the I$^2$C bus.

## 13.2 The I$^2$C ISR

As mentioned the I$^2$C ISR will fetch data from the bus and forward it to the 3C task as well as deliver data from the 3C task on to the I$^2$C bus. The specific communication protocol are defined in chapter G on page 201, and this states that 6 different requests from the OBC must be handled by the PSU and therefore by the I$^2$C ISR. Before the ISR is designed an introduction to the I$^2$C Bus on the Cubesat is given.

### 13.2.1 The I$^2$C Bus on the satellite

The I$^2$C is a serial communication bus developed by Phillips where masters and slaves are connected through two bidirectional wires, a serial data wire (SDA) and a serial clock wire (SCL). The I$^2$C features integrated filtering for spikes on the bus, data transfer speeds of up to 3.4 Mbit/s and collision detection and arbitration. The bus is multi-master which means that it is possible to have more than one device controlling the same bus. However this function will not be used in the Cubesat, where only the OBC will function as master and all other devices, including the PSU, are slaves.

The transfer sequence begins with a START condition, which is when SDA goes high while SCL is high. The START condition indicates begin of data transfer and is always generated by the bus-master.

After a data transfer sequence is initiated by a START condition the master sends one byte on the SDA consisting of the address of the slave it wants to communicate with and a R/$\overline{W}$ bit (see figure 13.2). The R/$\overline{W}$ bit determines whether the master writes data to slave or expects to read data from slave. As an extension a to the standard I$^2$C protocol a header byte and a checksum byte are always sent after the address byte.

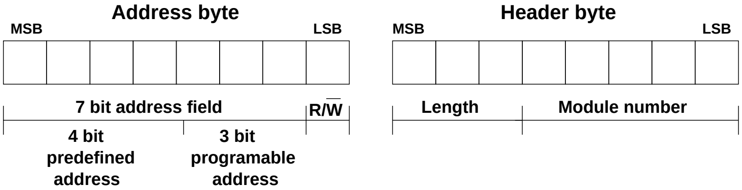


**Figure 13.2:** To the left is the address byte which is the first in all communication on the I$^2$C bus. To the right is the header byte which defines the amount of data send and the command as the module-number

All bytes transmitted on the bus are followed by a acknowledge signal (A) generated by the receiver to signal that the byte has been read. The transfer sequence are terminated by the master with a STOP condition which is a low to high transition on the SDA while SCL is high.

First in the header byte there are a 3-bit data-length field which defines how many 8 bit data packages the header will be followed by. 000 is equal to one package, 001 to two and so on to 111 which is equal to 8 packages. The rest of the header is a 5 bit module field which is used either to define a module in the slave which the master wish to read or simply to define a command.

The first data package after the header is always a checksum which is calculated by adding the header and all data packaged together, the checksum is then the 8 lowest bits of this number. When data is received the checksum is calculated again and is subtracted form the received checksum and if the result is zero the the data is valid. If the result yields non-zero the data is invalid and a retransmission must be initiated by the OBC.

There are three basic communication formats that can be used on the I$^2$C bus: master write to slave, master reads from slave or a combination, i.e. master both writes to and reads from slave. The first two formats are depicted at figure 13.3 and they are the ones used in the protocol on the Cubesat.

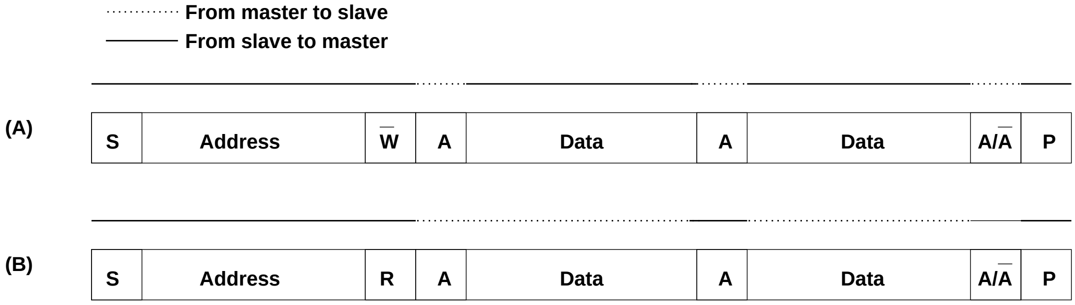


**Figure 13.3:** Two basic I$^2$C communication formats: (A) Master write to slave and (B) Master read from slave

The third combined format requires the dataflow to be altered between a START and a STOP condition with what is known as a repeated START condition, which is not supported by the ADuC812 and will therefore not be used in the Cubesat. However the other two formats will be used in connection with each other to implement the bidirectional functionality. There are two specific communication formats (combinations of the two above described formats) used on the Cubesat between the OBC and the PSU:

**Housekeeping request** In this format the OBC request housekeeping data, i.e. it begins by using format A, sending a header and a checksum, thereby requesting the PSU to send housekeeping data. The OBC then uses format B to read the requested housekeeping. If the slave received valid data in the request it will send the requested housekeeping, but if the data was invalid an error message in form of module- number 20 in the header will be send.

**Command request** In this format the OBC request a command to be executed, e.g. a subsystem turned off. It begins by using format A to issue the command to the PSU, which will test the checksum for errors. The OBC will the use format B to receive a VALID (module- number 19) or INVALID (module- number 20) from the PSU.

As it can be seen from the above both formats consists of first a format A communication and then a format B - the only difference is the contents of the transmissions.

It is in this connection important to remember that only the OBC can initiate communication both back and forth. Therefore the OBC must always ask the slave whether the transmitted data was valid and if not initiate retransmission. If OBC receives invalid data itself it will not inform PSU, but only initiate retransmission.

### 13.2.2 The I$^2$C on the ADuC812

The I$^2$C implementation on the ADuC812 consists of three SFR's:

**I2CADD** In this register the address of the MCU are defined when operating in slave mode.

**I2CDAT** This is the send and receive register.

**I2CCON** This register holds the configuration of the I$^2$C controller, indicated by 8 bits.

Of the 8 configuration-bits only 4 apply to the slave implementation of the PSU which are:

| | |
|---|---|
| I2CM | This bit enables slave mode when cleared |
| I2CRS | Setting this bit will reset the I$^2$C interface |
| I2CTX | This bit mirrors the R/$\overline{W}$ of the ongoing communication |
| I2CI | This bit is interrupt flag for the I$^2$C port, it is set after a byte is transmitted or received. |

The I2CI bit must be cleared in software after each read or write, which will release the SCL. As long as the I2CI bit is set the SCL are hold low and thereby ensuring that the master does not transmit again until slave is ready.

### 13.2.3 Design of I$^2$C ISR

When designing the I$^2$C ISR three datatypes are created, a 2 byte receive buffer (RBUF), a 6 byte send buffer (SBUF) and a receive flag (RFLAG).

When an interrupt caused by the I$^2$C interfaces occurs, the I$^2$C ISR is run and it must then determine which communication format that has begun. This is done by examining the R/$\overline{W}$ bit in the received address byte. If it is 0 a format A is initiated and the ISR clears I2CI to indicate that the OBC can send the next byte and waits. When the next byte arrives it is moved into the receive buffer (RBUF[0]) and the receive flag (RFLAG) are set to inform the 3C task that data has arrived. The I2CI bit is then cleared and the ISR waits for the checksum to arrive. When that happens the checksum is moved to RBUF[1] and the this time the I2CI bit is not cleared and it is the up to the 3C task to clear it when the data is interpreted and e.g. the housekeeping data is transfered to the send-buffer.

If the R/$\overline{W}$ bit is set then data must be transfered from the send-buffer and to the I$^2$C bus one byte at a time. The flow in the I$^2$C ISR is shown on figure 13.4.

For the ISR however it is impossible, when the interrupt occurs, to know at which point it is waiting for interrupt and therefore a ISR status register (ISR_STAT) are created in which the current entry point of the ISR are placed.

### 13.2.4 Code Analysis of the I$^2$C ISR

In the code implemented for the ISR routine a number of variables that hold state-information have been identified and protected by the protection mechanisms designed in subsection 11.7.3 on page 142.

The bus-data itself is not protected from SEUs by the I$^2$C ISR, since protection of these data will be handled by receiver and transmitter tasks which, inteprets the data including checksumming.

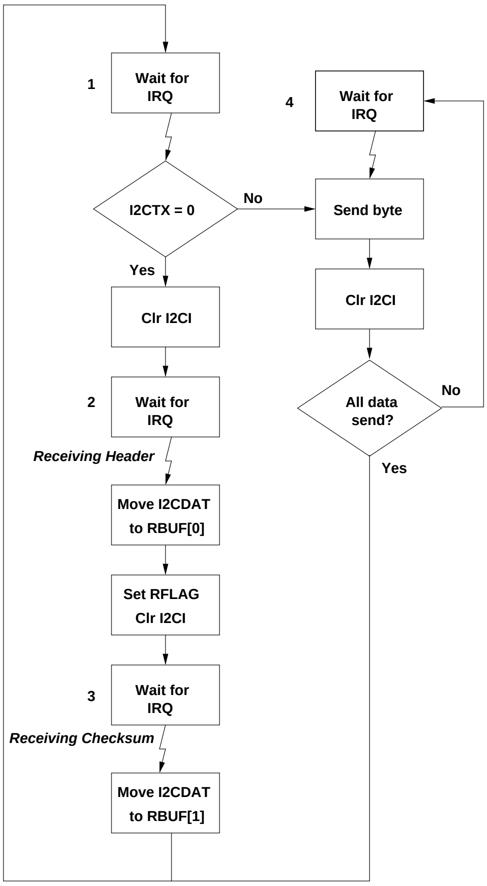


**Figure 13.4:** The flow in the I$^2$C ISR

## 13.3 The 3C Task

As mentioned above the 3C task must take care of all control and command jobs on the PSU, as well as interpret and react to data from the I$^2$C ISR. The 3C task is scheduled at 1kHz as described in section 11.5.3 on page 137.

When called, the 3C task first resets the internal watchdog timer and then it examines RFLAG to determine if new data has arrived. If this is the case then RFLAG is cleared again and the contents of RBUF is read and the checksum is calculated. Then the calculated checksum is compared with the received data to determine if the data is valid. If this is the case then actions are taken according to the module number in the received data. Thereafter is moved to SBUF VALID, with the exception of a housekeeping transmission where the correct housekeeping will be accepted by the OBC as VALID. Then the new checksum is calculated and placed in the send buffer and the I2CI is cleared, thereby signaling that PSU has data ready.

If the received data is not valid the only thing returned to OBC is a header with the module number corresponding to INVALID. In figure 13.5 the flow of the 3C task can be seen.

### 13.3.1 Actions Corresponding to Module Numbers

As described above the 3C task must perform different assignments according to the module number that is received from the OBC. These are described in appendix G on page 201 and will be summarized here with more implementation specific details.

**Boot Port**

To set boot port to PROM pin P3.0 is set low and to set it to EEPROM pin P3.0 is set high.

**External Watchdog Timer**

When a specific command or any other signal arrives the external watchdog timer must be reset. If no signal has arrived from the OBC for 10 seconds the OBC must be power cycled. The external watchdog timer is implemented from the assumption that the speed of the 3C task found in section 11.5 on page 135 holds. It states that the 3C task is run at 1 kHz, i.e. 1000 times per second. Each time the 3C task is run it increments a 32 bit register and every time this value reaches 10000, 10 seconds have elapsed and the OBC is shut down

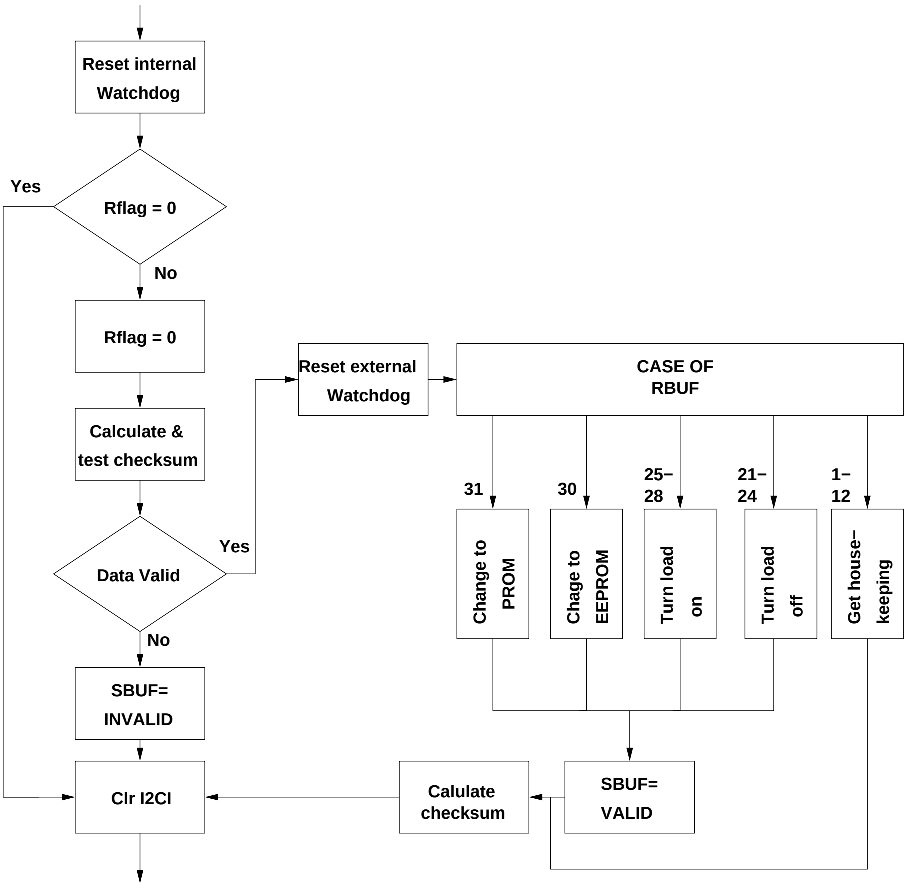


**Figure 13.5:** The flow in the 3C task

using the OCPC. The register is then set to 300000 and used to count down in for the 5 minute delay before the OBC is turned on again. This is also done if the OBC is shut down because of a overcurrent.

When a signal arrives from the OBC the watchdog register is cleared, thereby resetting the external watchdog timer.

### Turn Loads On and Off

When loads must be turn either on or off the OCPC port are used which is addressed as external memory as described in chapter 10 on page 107. This means that the status of the subsystems can be read on the lowest 4 bits, while they can be turned on and off using the 3 lowest bits for ACS, TRD and CAM and pin P3.0 for OBC. A 1 indicates on and 0 off.

### Get Housekeeping Data

When getting housekeeping the only thing that is done is that the requested data is moved from the memory, where the ADC-sampler has placed it, to the send buffer.

## 13.3.2 Code Analysis of 3C Task

Again, as with the I$^2$C ISR, state-variables are protected by the designed protection mechanism. Further the checksumming mechanism used on received and transmitted data helps to make the implementation robust, because it is possible to detect faults incurred by e.g. a SEU. No variables in this tasks is subject to transient faults.

# Chapter 14
# Scheduling and Implementation of Software

This chapter will describe implementation of the designed software including scheduling of tasks and ADC samplings. Further test and evaluation of the software will be described.

## 14.1 Scheduling

This section treats scheduling theory and applies this to the software designed for this project. The section is written primarily on basis of [Burns and Wellings, 2000]. A fixed schedule that schedules the execution of the taskset will be derived on basis of the task requirements and this schedule will further be used to derive the schedule for the ADC. In table 14.1 the notation that will be used through the section is shown. Each of the entries will be explained as they are used.

| **Identifier** | **Description** |
|---|---|
| C | Worst Case Computation Time |
| D | Deadline of the task |
| N | Number of tasks in the taskset |
| P | Priority of the task |
| R | Worst case response time of task |
| T | Task period |
| U | Utilization of the task |
| I | Interference of the tasks |

**Table 14.1:** Notation used in scheduling theory

In general time will be counted in machine cycles (at a 16MHz clock) rather than in seconds. This is done to avoid inconvenient fractional numbers. One machine cycle (MC), which is 12 clock periods, corresponds to:

$$MC = \left(\frac{16MHz}{12}\right)^{-1} = 0.75\mu s \quad [s]$$

### 14.1.1 Tasks Model and Requirements

In order to mathematically derive a schedule for the taskset, it is necessary to have a well defined model of what a task is. To that end we will let a task be an entity which requires to execute at most **C** machine cycles on the MCU in a period of **T** MC. Further all tasks will be assumed to be periodic with deadlines equal to periods, i.e. **D=T**. At each time interval **n**$\cdot$ **T** the process is said to be released, i.e. it becomes ready to execute.

Further, a process can be said to exhibit preemptive or non-preemptive behavior. Preemption is the possibility of a process of higher priority to interrupt a process at a lower priority, when the higher priority process is released. In practise preemption is a result of an interrupt that triggers an ISR or in more comprehensive systems it is handled by the operating system[1].

In table 14.2 requirements on **C** and **T** are given as well as their frequency in Hz and whatever they exhibit preemptive behavior or not. The frequency requirement on the ADC ISR is condensed from table 10.1 on page 112. The I2C ISR frequency requirement is specified for the worst case situation, which is when data is being transmitted at the maximum rate of 100 kbit/s.

Already from the values stated in the table it is obvious that the real-time requirements cannot be met, since **C**$>$**T** for the ADC ISR. Actually, as can be seen from the table, the taskset is very far from being scheduable on the given processor. This is a fact that cannot be omitted, however the remaining part of this section will still need to derive some schedule to order the execution of the taskset. This schedule will be derived by assuming that the MCU is fast enough to cope with the tasks. Then, when the software is implemented, it will be possible to evaluate the severity of the lack of processor time.

[1] Which then in turn will use interrupts to handle preemption

| Taskname | C [MC] | Hz | T [MC] | Preemption |
|---|---|---|---|---|
| ADC ISR | 63 | 39203.6 | 34 | Preemptive |
| PCC inner loop | 53 | 18000 | 74 | Non-Preemptive |
| I2C ISR | 103 | 10000 | 133 | Preemptive |
| PCC outer loop | 86 | 1600 | 1200 | Non-Preemptive |
| 3C | 265 | 1000 | 1333 | Non-Preemptive |
| MPP-tracker | 131 | 100 | 13333 | Non-Preemptive |

**Table 14.2:** Tasks in the taskset and their requirements

### 14.1.2 MCU Utilization

Having established the fact that the taskset is not scheduable the next thing to do is to analyze the processor utilization in order to calculate a processor speed at which the taskset can be said to be scheduable. The utilization, meaning the percentage of time the task needs to execute instructions on the MCU in order to fulfill deadlines, of a task is given by:

$$U = \frac{C}{T} \tag{14.1}$$

Extending this to the taskset a figure for the total utilization of the MCU is obtained:

$$U_{MCU} = \sum_{i=1}^{N} \left( \frac{C_i}{T_i} \right) \tag{14.2}$$

In order to prove that a taskset is schedulable with a fixed schedule, as will be used here, the following sufficient test must evaluate to true [Burns and Wellings, 2000] page 471:

$$\sum_{i=1}^{N} \left( \frac{C_i}{T_i} \right) \leq N \cdot (2^{1/n} - 1) \tag{14.3}$$

Stating that the test is sufficient rather than necessary means that the taskset may be schedulable even though its utilization is higher than what is evaluated at the right-hand side of equation 14.3, but in general it cannot be proven.

In the case with **N**=6 the maximum utilization according to equation 14.3 is $U_{mcu} = 73.5\ \%$. The utilization of each task and the total utilization can be seen in the second column of table 14.3. As can be seen $U_{mcu} = 362\ \%$ which is way beyond the theoretical limit. Therefore, in order to make the taskset scheduable a new MCU-speed will be calculated:

$$Clock\ frequency = \frac{362\ \%}{65\ \%} \cdot 16\ MHz = 89\ MHz$$

The percentage of 65 % rather than 73.5 % is chosen to account for overhead and process synchronization which have not been included in the process model. New values for **T** and **U** have been calculated at this new frequency and can be seen in table 14.3.

| Taskname | U (old)[%] | C [MC] | T [MC] | U [%] |
|---|---|---|---|---|
| ADC ISR | 182 | 63 | 190 | 33 |
| PCC inner loop | 72 | 53 | 414 | 13 |
| I2C ISR | 77 | 103 | 745 | 14 |
| PCC outer loop | 10 | 86 | 6720 | 2 |
| 3C | 20 | 265 | 7465 | 4 |
| MPP-tracker | 1 | 131 | 74665 | $\simeq 0$ |
| In total | 362 | - | - | 65 |

**Table 14.3:** Tasks in the taskset and their requirements

### 14.1.3 Deriving the Schedule

At this point the actual schedule which must be abided to by the software is to be derived. To do this, one remaining problem must be solved. The taskset consists of both preemptive tasks and non-preemptive tasks, and due to the fact that the preemptive tasks executes as ISR routines, and therefore effectively schedule themselves, it is necessary to consider what effect this has on the scheduability of the non-preemptive tasks.

A simple approach to solve this problem is to eliminate the preemptive tasks by considering them as ”stealing” a percentage of the processor time which in effect adds this percentage to the execution time of each process. This assumes that the ISR-code execution is evenly distributed in time. Taking this view allows a schedule for the non-preemptive tasks to be derived.

The schedule is derived using rate monotonic priority assignment. Meaning that the priority is inversely proportional to the task period. This gives the following priority assignment: $P_{inner-loop} = 4$, $P_{outer-loop} = 3$, $P_{3C} = 2$ and $P_{MPP} = 1$. The schedule is derived by assuming that all tasks are released simultaneously in what is called a ”critical instant” then the highest priority process is chosen to execute and when finished then the next runnable task with the highest priority is chosen and so on. This scheduling algorithm has been implemented as a C++ program[2] that outputs the schedule for the software designed here.

The output of the program is a schedule consisting of 202 entries that schedules all tasks. However the program also reveals that not all deadlines are met. This is due to the interference phenomena that will be described shortly. In spite of this the schedule will still be used for the software, since the taskset was not scheduable in the first place when the MCU is running at 16 MHz.

**Interference and Response-time**

The above approach has two problems concerning what is called task interference, and it is these problems that are responsible for the problems described above:

- The assumption that the ISR code is evenly distributed is not entirely valid
- The lack of preemption invalidates the utilization test

Regarding both points the problem, the utilization test given in equation 14.3, is only valid when a release of a higher priority task immediately results in a context switch, i.e. the higher priority task preempts the lower priority task. In the case presented here this will not happen due to the lack of preemption.

Both the lack of preemption and the execution of ISR code will result in interference of the high priority tasks that cannot start execution immediately after release. This interference may cause the higher priority task to miss its deadline even though the utilization test (equation 14.3) holds for the taskset.

The fact that the utilization test alone is not adequate to evaluate if the taskset is scheduable and not is already evident from table 14.3 where it can be seen that $C_{3c} > T_{inner-loop}$. This makes it impossible to schedule the inner-loop in a non-preemtible environment.

In general, in order to see if a non-preemtible taskset is scheduable a second scheduability test should be conducted, which analyses if task response-times are within the deadlines. In a non preemtible execution environment the worst case response time ($R_i$) of a given task is calculated by [Burns and Wellings, 2000] page 495:

$$\omega_i^{n+1} = B_{max} + \sum_{j \in hp(i)} \left\lceil \frac{\omega_i^n}{T_j} \right\rceil \cdot C_j \tag{14.4}$$

where:
$B_{max}$ : is the maximum blocking time of a task
$j \in hp(i)$ : is the set of processes with higher priority than task $i$

This recursive equation yields $R_i$ if it converges. If it does not then the tasks is not scheduable at all. When using equation 14.4 to evaluate reponse times for the 4 non-preemtible tasks then interference from the ISR routines should be accounted for as well by including them in the taskset and assigning them higher priorities than the non-preemtible tasks. $B_{max}$ equals the worst case WECT of the taskset.

This second scheduability test will however not be applied here, since it is already evident that it will evaluate the taskset as not scheduable both with the given CPU and the CPU calculated to run at 89 MHz.

[2] Path:...

## 14.2 Software Implementation

The designed software has been integrated and tested using a simulator program. Some specific parts of the implementation will be covered in the following.

All un-used addresses in ROM have been programmed to hold the **NOP** instruction (No OPeration). This ensures that if a SEU makes the program jump to some unspecified address then the MCU will execute **NOP** instructions until the program pointer wraps to zero.

### Initialization

When the MCU is powered up the following peripherals are initialized by writing to the SFR-registers of each peripherals.The peripherals are initialized in the following order:

- Watch-dog-timer
- ADC
- DAC
- I$^2$C

The WDT is the first peripheral to be initialized in order to make sure that if a SEU disturbs the program during initialization then the WDT will reset the MCU.

Further, since the SDCC-compiler does not allow the initial state of a structure to be defined from the declaration then all the structures used by the SEU protection mechanisms are initialized manually.

### Schedule Implementation

The schedule has been implemented as a loop in the main function. The idea of this approach can be seen in the following code:

```
loop:
 TASK_POSITION=0;

 TASK_SYNCRO;
 innerloop();
 PCCDUTY;

 TASK_SYNCRO;
 innerloop();
 PCCDUTY;

 TASK_SYNCRO;
 outerloop();

 TASK_SYNCRO;
 MPPT();
 MPPTDUTY;

 ... Rest of Schedule ...
 goto loop;
```

The loop executes the functions according to the schedule and ensures syncronization by calling the `TASK_SYNCRO` before each task-execution. Further every time a task has updated a duty-cycle then the new duty-cycle is programmed in the DACs by the two macros `MPPTDUTY` and `PCCDUTY`.

The above code could have been implemented by more elaborate mechanisms, such as for example a switch statement that reads what actions to take from a table, but the chosen implementation however is the fastest possible.

The schedule for the ADC ISR is derived directly from the schedule of the taskset. Since this schedule does not deal with the housekeeping measurements then each time the 3C task is in the ADC schedule a new housekeeping measurement points is selected and measured. Therefore housekeeping data is sampled at a higher rate than required.

**External Pulse-Width-Modulators**

The analog PWM-modulators that are programmed from the DACs of the MCU will only operate properly with an input voltage of between 0.48V to 2.4V (See section 10.7 on page 122). Therefore, since the DACs outputs 0-2.5V, two tables have been implemented which maps the duty-cycle, internally represented as a number between 0-1024, to two bytes which respectively represents the corresponding low- and high-byte of the value to be programmed to the DAC.

This means that every time a DAC is programmed the calculated duty-cycle is used as index for both tables:

```
DAC0L=LOWTABLE[PCC_duty];   //Program DAC0 low-byte
DAC0H=HIGHTABLE[PCC_duty];  //Program DAC0 high-byte
```

**RAM, ROM and Stack Usage**

In order to see that both RAM usage and stack usage of the complete software is within acceptable limits, i.e. not all RAM is used and the stack does not move into memory occupied by other variables, code analysis has been performed by interpreting information files generated by the compiler and by running the software in a simulator.

The analysis showed that 102 bytes of RAM is used out of 207 bytes available in both the direct and indirect addressable RAM. The worst case stack size is 10 bytes and it showed up that the stack potentially could overwrite other variables in the RAM. Therefore the compiler was forced to locate the stack after the data segment in direct addressable RAM. The total ROM code-size is 5189 bytes meaning, whereoff about 2 kb is used for storing the duty-cycle tables and about 1.5 kb is used hold the function calls in the schedule.

## 14.3 Test and Evaluation

As described in the test specification in section J.5 on page 220 the module test of the software and controllers is performed with a number of limitations. This is partly caused by the lack of the necessary test equipment, but also because of the lack of processing power with makes it impossible to perform the tasks at an appropriate speed.

The test was carried out on the final day before the deadline of the project and therefore problems encountered during the test was solved as best possible on the spot. It should also be noted that the MCU was clocked at 11 MHz during the whole test, because the software, which due to the test conditions was changed often, must be downloaded to the device at this frequency.

When beginning the test the software was only able to run for short periods of time whereafter the system would stop. This error was tracked to the synchronization of the tasks which caused a deadlock in the system. The error was judged to be too comprehensive to be solved on the spot and therefore the synchronization was removed.

This however caused the ADC ISR to be executed free-running and thereby taking up much more processor power than designed to. This was partly solved by increasing the acquisition time of the ADC, but the ADC ISR used much processor power. It was therefore to be expected that the tasks in the taskset would execute at lower frequencies than designed for.

**Frequency of the Controllers**

The first test was to determine the frequency of the controllers by toggling a pin each time the inner loop was executed. The frequency can be seen at figure 14.1 and the frequency of the signal can be read to about 60 Hz. Because the pin is toggled each time the inner loop is run the frequency of the inner loop is about 120 Hz rather than the 2.2 kHz that could be expected with the processor running at 11 MHz. This is considered to be due to the free running ADC.

**Performance with all task running**

Because of the results in the former test it was juged that a test with all task running was not relevant to perform. Therefore the tests where done with only the task(s) under test running. As descriped in the test specification (section J.5 on page 220) the test on the I$^2$C ISR and 3C task was done as a whitebox with a debugger.

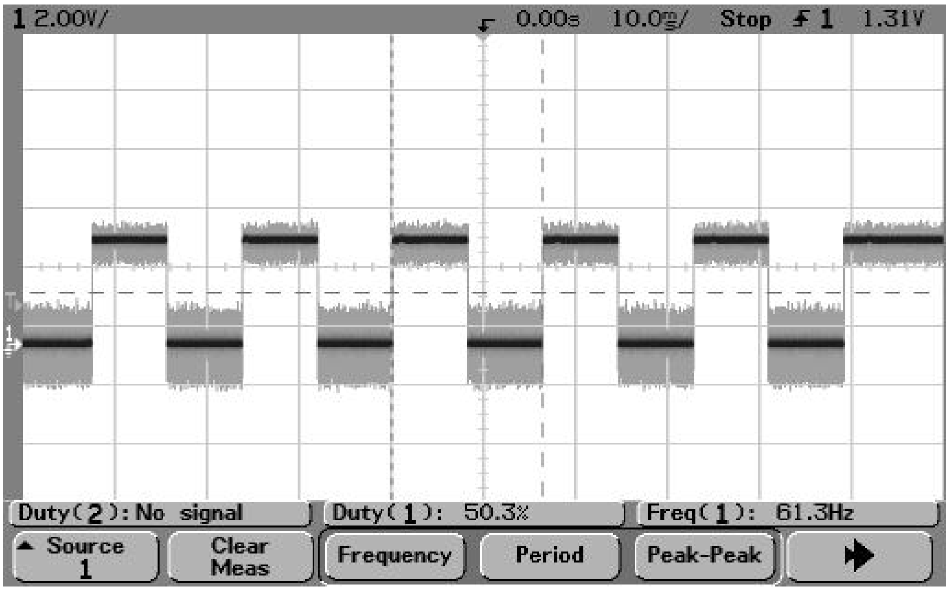


**Figure 14.1:** The frequency of the inner loop execution

### MPPT

For this test it was not possible to obtain a suitable solar array or programmable power supply and therefore an alternative test was made. Two standard alkaline batteries with a voltage of 1.5 V each where connected in series and used as a power source instead of the solar cells. The task of the MPPT was then to draw maximum power from the batteries. From figure 14.2 the result of the tracking can be seen and it is clear that the MPPT had found a power point to track and it is belived that this is the MPP of the batteries.

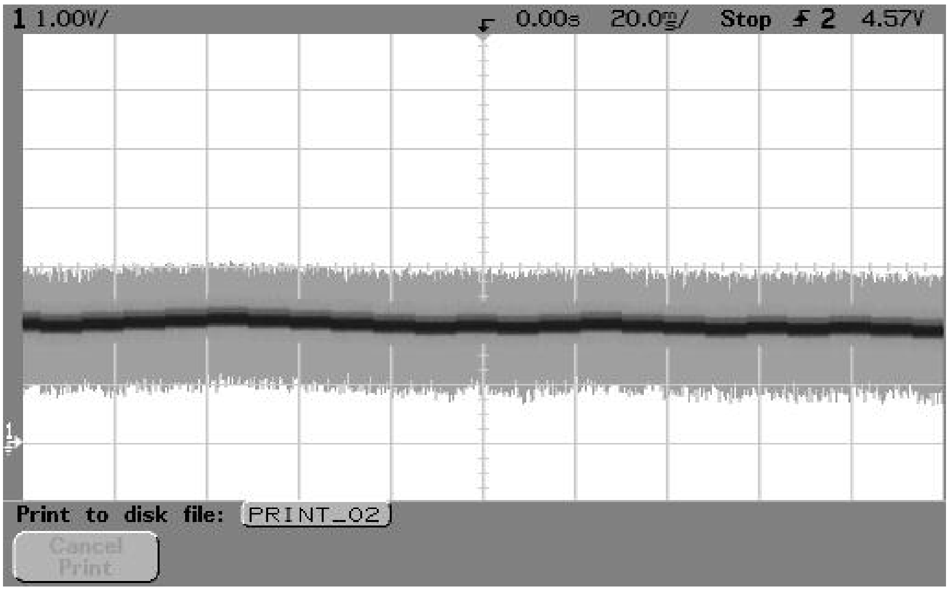


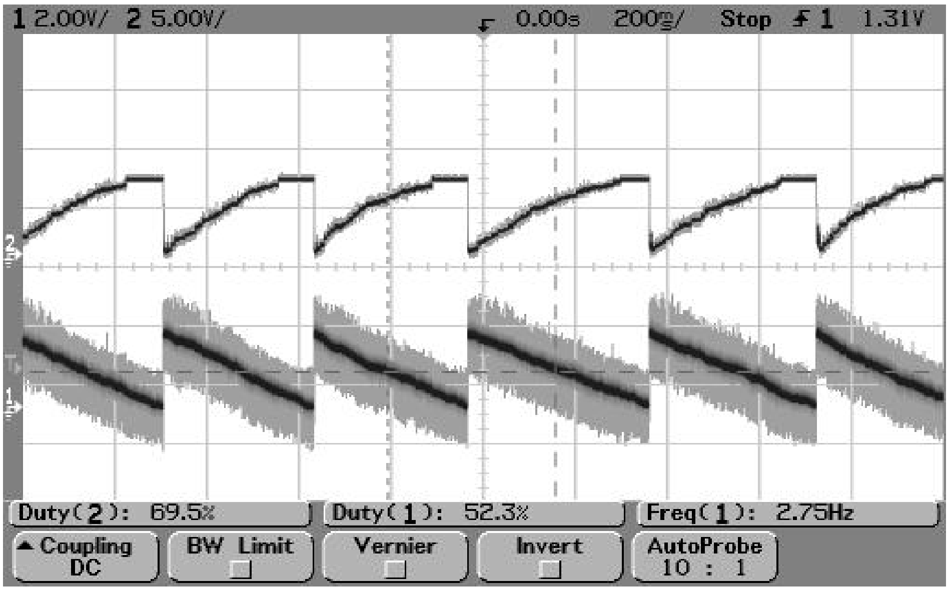
**Figure 14.2:** The testresult of the MPPT

### PCC controller

When testing the PCC controllers, both inner and outer loop, it became clear that even with no other tasks running, the speed of the controller could still not reach a satisfactoring level. Therefore an unstable bus voltage was expected, but the large oscillation that was seen on the bus (see figure 14.3) cannot be explained by this. The explanation to the ripple on the output voltage was found in the last minute with no time left to solve the problem. When the controller tries to regulate the bus voltage it will lower the DAC value which will raise the duty-cycle. This is because the gate-driver is inverting. The problem arises when the DAC value reaches 0 the contoller will continue to lower the value which will cause a overflow in the DAC register which will wrap from 0 to 1023.

This phenomena can clearly be seen on figure 14.3 where the controller lowers the DAC value, thereby stepping up the duty-cycle to raise the bus voltage. The resulting rise in voltage that can be seen is identical to the response of the controller found in section 9.5 on page 89, but with a large overshot. This is probably caused by the very slow controller frequency. A slow frequency causes a decrese in phasemargin and thereby making the system more underdamped which again gives a larger overshot. When the duty-cycle drops below the 0 it wraps to high and the voltage drops suddently.

The problem could be solved by including boundary checks in the controller tasks so that the duty-cycle never moves beyond limits.



1 2.00V/ 2 5.00V/ 0.00s 200ms/ Stop 1 1.31V

Duty(2): 69.5% Duty(1): 52.3% Freq(1): 2.75Hz

Coupling DC

BW Limit □

Vernier □

Invert □

AutoProbe 10 : 1

Figure 14.3: The bus voltage at top and dutycycle at the bottom

### Conclusion on the Test

The conditions under which this test has been performed have been very unsatisfactory, because of the time pressure during the last part of the project. The software tested here has not prior to this test been tested at all on the hardware and no real results were expected.

However, in spite of all the shortcomings, the testing showed that the software was able to measure physical values in the system, do calculations on these values and apply the result back to the system.

The general design approach, apart for the synchronization rutines, did work and the taskset was executed and behaved as expected. A problem with the calculations were found as well as their probable solution.

It is the strong belief of the project group that a few days more of testing and integration could have solved the remaining problems and brought the implemented parts of the PSU to operate with the best possible performance with the processor available.

# Part IV

# Deployment, System Test and Results

# Chapter 15
# Implementation and Evaluation

## 15.1 Prototype Implementation

Considering the limited time available during one semester, compared to the task of designing and implementing a product that is actually suited to operate in space, it was chosen not to try and implement the complete design. Instead a prototype which can be used to test and evaluate the key functionality of the PSU was designed and build. The key functionality is considered to be: Power conversion and power point tracking. In order to build a prototype that can test this functionality, the following designs need to be implemented:

1. Converters

2. Measurement points for converter controls

3. MCU including PWM-modulators, but excluding external ROM.

4. All designed software

The following will describe how this prototype has been implemented. The prototype has been built on a number of Printed Circuit Boards (PCB) in order make it easier to test individual circuits. In all, four PCBs have been manufactured and the schematic diagrams of these can be found in appendix L on page 227. The four PCBs are:

- MCU Core Implementation

- PWM-modulators and gate drivers

- Converters and measurement circuitry

- Serial Connection (RS232) interface print

On all PCBs all connections that is to be connected with other PCBs are assembled in connector ports of 8 signals. This allows easy connection and disconnection of individual PCBs The contents of each PCB will be described briefly in the following. The PCBs can be found on the enclosed CDROM[1].

### MCU Core Implementation

The MCU PCB consists of the minimum components needed to operate the MCU. These are; the ADuC812 micro-controller, power on reset circuit, 11 MHz crystal and the necessary by-pass capacitors. The 11MHz crystal is chosen for the prototype rather than the specified crystal oscillating at 16MHz. This is because software can be downloaded easily to the internal flash-ROM of the MCU at this frequency.

### Pulse Width Modulators and Gate Drivers

The PW-modulators designed in chapter 10 on page 107 are implemented on this PCB board as well as the interface circuits to the DACs of the MCU. Also the designed gate-driver circuitry for the two converters are implemented on this board.

### Converters and Measurements

The Converters and Measurements PCB consist of the designed boost- and buck-converter and the circuitry to measure all the currents and voltages needed for control of these two converters.

### Serial Connection (RS232) Interface Print

This PCB is not part of the design, but it implements the circuitry that is needed to download software from a PC to the flash ROM of the MCU. This consists mainly of a MAX232 circuit that is used to convert voltage levels between the RS-232 connection and the MCU.

[1] CDROM/PCB/

### 15.1.1 External Parts of the PSU

The following will describe what has been used to simulate the external parts of the PSU, i.e. solar-cells, batteries and users. Further it should be noted that not all the components used, such as capacitors, fulfill the requirements of a space application.

**Solar Array**

At the time the PSU prototype was implemented the solar cells that is supposed to be used on the Cubesat were not available as they were not yet shipped from the manufacturer. Instead a battery of 3V has been used to simulate the input from the solar cells during the tests that was performed on the assembled prototype.

**Batteries**

The batteries from Danionics were available for the prototype testing, but because only a limited number of batteries are available, a Ni-Cd battery with nominal voltage of 7.2 V have been used for all initial testing.

**Users**

The users of the PSU have been simulated by a variable resistor that variates the resistance in steps, with step-sizes available in more decades.

## 15.2 Evaluation

In this section the product described above will be evaluated hereunder discussing: What has been accomplished, what has not been accomplished and why this has not been accomplished.

### 15.2.1 Accomplished / Not Accomplished

In this project a complete hardware prototype for the PSU unit has successfully been designed and produced, here under a MCU, PWMs, DC-DC converters and measuring circuits. These have all been brought to working condition which has been verified both through module- and integration-tests. The only exception is the hold circuits for the PWMs which do not fulfill their tasks.

The DC-DC converters do not fulfill all of their performance requirements. The PCC converter is not capable of maintaining an adequate efficiency throughout its complete operating range and the MPPTC cannot fulfill the efficiency requirement at all. The PCC does not create a ripple on the power bus larger than allowed while the MPPTC does create a ripple a little larger than allowed. This however is of no great concern since no other circuits are effected by this. Both the MCU and the PWMs performed satisfactorily during all tests and are therefore considered fully functional.

Controllers have been designed for the converters and have proven their functionality through simulations. The MPPT has shown good functionality throughout the tests, but has yet to be tested under proper test conditions. The PCC controller has yet to show functionality outside the simulator.

A functional software-platform has been developed which however is not capable of offering mutual exclusion. The I$^2$C communication protocol has been designed and successfully tested in a simulator.

Neither the design or the completion of a space worthy PSU has been accomplished. The hardware does not fulfill neither the acceleration, the vibration, the radiation, the vacuum or the size requirements. All components however fulfill the temperature requirements. It has also not been accomplished to implement all the designs for the PSU done throughout the project, under here Housekeeping measurements, OCPCs, external memory for the MCU and hardware interface for the I$^2$C interface.

# Chapter 16
# Conclusion

In this conclusion first a summary of the project is given, here under describing the four phases of development in the project: Analysis, Design, Implementation and Test. This will also include a evaluation of the parts and their correspondingly contents.

This is followed by a short discussion of and conclusion on the Cubesat and the interaction between Cubesat and the project described in this report. Finally a conclusion on the project as a whole is presented.

### Analysis

First the surroundings around the satellite and materials in the power supply was analysed, which was done in order to find out how the electronics behave in space, thereby placing requirements on the power supply. This lead to the requirement specification which, based on the previous analysis and documents from the Cubesat project, specified the requirements of this project. Then a system analysis was made, under which different topologies for the power supply was discussed and a solution was chosen thereby determining the overall design.

### Design

In the design part first a introduction to the PSU was given and the design was divided into seven parts: Maximum Power Point Tracking Converter, Power Conditioning Converter, Output Filtering, Over-current Protection Circuits, Controller Design, Digital Hardware and finally the software.

Control algorithms were then designed to control the converters. This ended in a MPPT algorithm to control the MPPTC, and a cascaded loop controller for the PCC. Different software tasks and their real-time requirements were identified and the tasks were designed and implemented on a platform consisting of the C-language and a number of support routines.

### Implementation

The full design was not implemented, but only a subset that represents the core PSU functionality was implemented in a prototype. The prototype consisted of the two converters, measurement points for converter control and the digital hardware in a minimised configuration.

The complete software design was implemented in code, but due to lack of test equipment it has been impossible to test parts of the software design in reality.

### Tests

Module tests were carried out for each module of both hard- and software, as well as simulations of the controller designs. The parts that are part of the prototype implementations was brought together and brought to a functional state, but due time shortages it was not possible to test the prototype against the level of the system acceptance test. Instead a final integration test was performed which demonstrates the capability of the prototype as it worked at the end of the project period.

### The Cubesat Project

Even though it does not show very clearly in the final product, a very large amount of resources have been used to communicate with other groups within the Cubesat project, e.g. in order to obtain specific requirements. This has resulted in a number of changes of the PSU design throughout the project period. This has been a valuable, but challenging experience for the project group.

### Summary

All in all, much needs to be done yet if the PSU designed in this project is to become fully operational for the Cubesat project, but with or without further design changes, all the resource demanding preliminary work has been done and the PSU can with some effort be brought to a fully operational state.

# Bibliography

[01gr732, 2001] 01gr732. *OBC Bootsequence*. AAU, 2001.

[01gr930, 2001] 01gr930. *Mission Analysis and Design*. AAU, 1.2 edition, 2001.

[1N5823, 2001] 1N5823. *5 amp Schottky Rectifier 1N5823*. Microsemi, 2001.

[930 and 931, 2001] Gr 930 and Gr 931. *Missionanalysis*. AAU, 2001.

[ADG704, 1999] ADG704. *CMOS, 4 ohm Low Voltage 4-channel multiplexer*. Analog Devices, 1999.

[ADG708, 2000] ADG708. *CMOS, 3 ohm Low Voltage 4/8 channel multiplexers*. Analog Devices, 2000.

[ADuC812, 2001] ADuC812. *MircoConverter, Multichannel 12-Bit ADC with Embedded FLASH MCU*. Analog Devices, 2001.

[ANALOG, 2001] ANALOG. *ADuC812 Errata Sheet, REV E2*. Analog Devices, 2001.

[AVX, 2001] AVX. *Surface Mount Tantalium Capacitors*. AVX, 2001.

[BAT85, 2000] BAT85. *BAT85*. Philips Semiconductors, 2000.

[Burns and Wellings, 2000] Alan Burns and Andy Wellings. *Real-Time Systems and Programming Languages*. Addison Wesley, 2000.

[Connolly, 2000] Ryan Connolly. *The P-POD Payload Planner's Guide*. California Polytechnic State University San Luis, 2000.

[DANIONICS, 2001] DANIONICS. *Lithium-Ion Polymer Batteries, DLP 4443573*. Danionics, 2001.

[Ebert, 1998a] Hans Ebert. *Grundlæggende Maxwellsk Feltteori*. Aalborg Universitet, 1998.

[Ebert, 1998b] Hans Ebert. *Quasistatiske Elektriske og Magnetiske Felter*. Aalborg Universitet, 1998.

[Emcore, 2001] Emcore. *InGaP/GaAs/Ge Dual-Junction Solar Cells*. Emcore, 2001.

[Erickson, 1999] Robert W. Erickson. *Fundamentals of Power Electronics*. Kluwer Academic Publishers, 1999.

[FAIRCHILD, 2000] FAIRCHILD. *HUF76145P3, UltraFet*. Fairchild Semiconductor, 2000.

[Gene F. Franklin and Emami-Naeini, 1994] J. David Powel Gene F. Franklin and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison Wesley, 1994.

[Gomaa, 1999] Hassan Gomaa. *Software Design Methods for Concurrent and Real-Time Systems*. Addison-Wesley, 1999.

[Hansen, 2001] Flemming Hansen. *Byg din egen satellit*. Dansk rumforskningsinstitut, 2001.

[Haugen, 1994] Finn Haugen. *Regulering Av Dynamiske Systemer*. Tapir Forlag, 1994.

[Heilmann, 1992] Thomas Heilmann. *Regulerings Teknik*. Akademisk Forlag, 1992.

[ICL7660, 1999] ICL7660. *MAX1044 Voltage Doubler*. MAXIM Integrated Products, 1999.

[IEEE, 1998a] IEEE. *IEEE Recommended Practice for Software Requirements Specifications*. IEEE, 1998. IEEE standard 830.

[IEEE, 1998b] IEEE. *IEEE Standard for Software Test Documentation*. IEEE, 1998. IEEE standard 829.

[Knopf, 1999] Hannes Knopf. *Analysis, simulation, and evaluation of maximum power point tracking (MPPT) methods for a solar*. Portland University, 1999.

[Lawetz, 1975] Henrik Lawetz. *Beregning af solindfald*. Danmarks Tekniske Hoejskole, 1975.

[LM555, 2000] LM555. *LM555 Timer*. National Semiconductor Corporation, 2000.

[MBR1100, 1999] MBR1100. *Axial Lead Rectifier, MBR1100*. Motorola Inc, 1999.

[MC33151, 1999] MC33151. *MC33151 Low-Side Gate driver*. Motorola inc, 1999.

[MCP101, 2001] MCP101. *Microcontroller Supervisory Circuit with Push-Pull Output*. Microchip Technology Inc., 2001.

[MK9-P22A, 2001] MK9-P22A. *Conceptual Structual Design*. AAU-Cubesat, 2001.

[Ned Mohan, 1989] William Robbins Ned Mohan, Tore Underland. *Power Electronics - second edition*. Wiley, 1989.

[OP181, 1996] OP181. *Ultralow Power, Rail-to-Rail Ouput Operational Amplifiers*. Analog Devices, 1996.

[OP191, 2000] OP191. *Micropower Single-supply Rail-to-Rail Input/Output Op Amps*. Analog Devices, 2000.

[Palz, 1978] Wolfgang Palz. *Solar Electricity*. Butterworths, 1978.

[PHILIPS, 2000] PHILIPS. *Datahandbook Soft Ferrits and Accessories*. Philips, 2000.

[Roesch, 2001] Will Roesch. *http://www.ssel.montana.edu/merope/subsystems/subsys_thermal.html*. SSEL, 2001.

[Sedra and Smith, 1998] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits*. Oxford University Press, 1998.

[Semiconductors, 2000] Philips Semiconductors. *The $I^2C$-BUS SPECIFICATION, VERSION 2.1*. Philips, 2000.

[Serway, 2000] Serway. *Physics*. Saunders College Publishing, 2000.

[SUB75P03, 2001] SUB75P03. *SUB75P03-07, Single P-Channel*. Vishay Siliconix, 2001.

[Yamaguchi, 2001] Nobumasa Yamaguchi. *horse.mes.titech.ac.jp/srtlssp/cubesat/index.html*. TIT, 2001.

# Appendix A
# Abbreviation List

This appendix contains all the abbreviations used throughout the report.

| Abbreviation | Description |
|---|---|
| ACS | Attitude Control System |
| BOL | Beginning Of Life |
| CCM | Continuous Conduction Mode |
| DCM | Discontinuous Conduction Mode |
| DHS | Data Handling Software |
| DHW | Digital Hardware |
| DMA | Direct Memory Access |
| EOL | End Of Life |
| FPS | Fixed Priority Schedule |
| ISR | Interrupt Service Routine |
| $I^2C$ | Inter Integrated Circuit |
| LEO | Low Earth Orbit |
| LSB | Least Significant Bit |
| MC | Machine Cycle |
| MIPS | Million Instructions Per Second |
| MPP | Maximum Power Point |
| MPPT | Maximum Power Point Tracking |
| MPPTC | Maximum Power Point Tracking Converter |
| MPZ | Matched Pole Zero |
| OBC | On Board Computer |
| OCPC | Over Current Protection Circuit |
| OTP | One Time Programmable |
| PCB | Printed Circuited Board |
| PCC | Power Conditioning Converter |
| POR | Power On Reset |
| PSU | Power Supply Unit |
| PWM | Pulse Width Modulation |
| RTOS | Real Time Operating System |
| RTSS | Run Time Support System |
| SDCC | Small Device C Compiler |
| SEL | Single Event Latchup |
| SEU | Single Event Upset |
| SFR | Special Function Registers |
| SPDT | Single-Pole-Double-Throw |
| SPST | Single-Pole-Single-Throw |
| TRD | Transmitter/Radio Device |
| UGBP | Unity Gain Bandwidth Product |
| WCET | Worst Case Execution Time |
| WDT | Watch Dog Timer |
| 3C | Command, Control and Communication |

# Appendix B
# Solar Cells

In this appendix solar cells are going to be analysed focusing shortly, first on the theory of solar cells in general and afterwards on the actual solar cells that are to be used on the Cubesat.

## B.1 Functionality of Solar Cells

The solar cells is the power source in the Cubesat. They deliver the input power to the power supply.

### B.1.1 How Solar Cells Work

The solar cells are photo-voltaic cells, which as the name implies convert energy from photons into electric energy. The cells are typically made of doped silicon, and energy from the absorbed sunlight knocks electrons loose creating an electric current. This current can be drawn out through metal contacts to external use for example in a power supply. The silicon is doped like an ordinary semiconductor, and thus there are two plates, one with N-type silicon and one with P-type silicon. Another type of solar cells is the GaAs solar cell. It consists of P-doped and N-doped GaAs on the top and bottom respectively. Each have metal contacts to draw the current. The sunlight is absorbed by the P-doped GaAs and a current can flow. The electric field between the plates generates a voltage and thereby power is obtained.

Should the cells however receive little or no illumination, they change characteristics, and start to act as diodes instead. This also happens if a too high voltage level is forced on the cell. These effects count for both silicium, GaAs and most other known cells.

### B.1.2 Voltage-Current Relationship

When drawing power from a solar cell, it is important to note that the power available is dependent on the manner in which it is drawn from the cell. The voltage and current drawn are unlinearly dependent on each other. To obtain the maximum efficiency of the power cells it is therefore important to keep the voltage level at a certain point, depending on the maximum power available. This is called maximum power point tracking (MPPT) and is done by using the I/U characteristics of the particular cell. A typical I/U graph for a solar cell, is shown in figure B.1

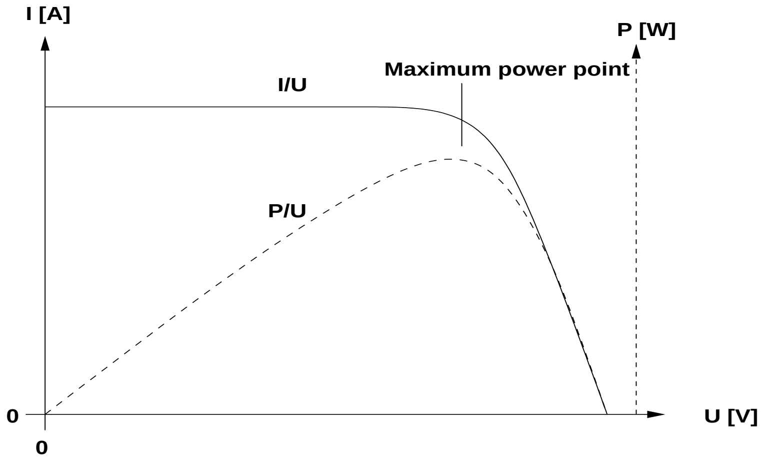


**Figure B.1:** I/V characteristics of a solar cell

As can be seen after a certain point a rise in the current will have a big effect on the voltage level of the cell. Similarly a rise in the voltage level above a certain point will result in a drastic loss of current. For comparison, a power curve is shown with a dotted line. When compared to this curve, the point on the I/U curve that gives the maximum power can be found. This is the maximum power point.

The I, U, and P characteristics, are also dependent on the temperature of the cell. Precisely how, may vary with individual types of cells, but in general, the current increases slightly and the voltage level decreases significantly with increasing temperature. An example of this is shown for a silicon cell in figure B.2 [Palz, 1978].

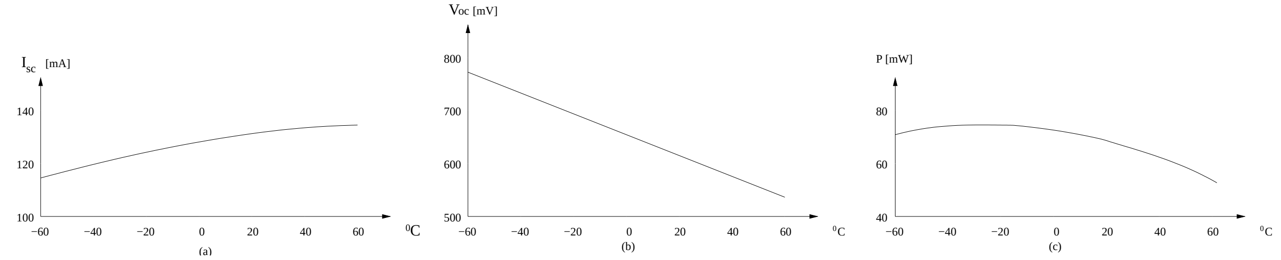


**Figure B.2:** Temperature effect on I (a), U (b), and P (c) characteristics of a solar cell

Here $I_{sc}$ is the short circuit current $V_{oc}$ is the open circuit voltage, and P is the maximum power. As can be seen, the current increases unlinearly, with a rising temperature (a), while the voltage level decreases linearly with rising temperature (b). These put together give the power curve seen in figure (c), where for ordinary use the effect decreases with a rising temperature. Beyond a certain point (here about -30 °C),the power of the cell will decrease with a fall in temperature. This is caused by the unlinear characteristic of the current/temperature curve. The varying temperature will also cause the point of maximum power to move, which must be taken into consideration if maximum power point tracking is used.

### B.1.3 Radiation Dependency

The following describes the dependency of the power on the radiation with regard to two parameters: radiation intensity and the amount of radiation.

**Intensity**

The maximum flow of electrons caused by the absorption of sunlight is the short circuit current and it is independent of the voltage. This short circuit current depends linearly on the intensity of the light [Palz, 1978]. The open circuit voltage varies a little with light intensity, and the short circuit current varies a lot. Therefore the power is dependent upon the light intensity [Palz, 1978].

**Amount**

Figure B.3 shows the power lost in a solar cell relative to the maximum power it can deliver. It is shown as a function of the amount of irradiation by 1 MeV electrons. It shows that the power lost in the solar cell grows as the irradiation grows.

## B.2 The Solar Cells on Cubesat

The solar cells on Cubesat will be dual-junction GaAs-solar cells from Emcore [Emcore, 2001]. The satellite will be carrying solar cells on five of its six sides, since the last side will be reserved for the payload of the satellite. The cells will be in the form of $76 \cdot 36$ mm panels. There will be two panels on each side. This gives

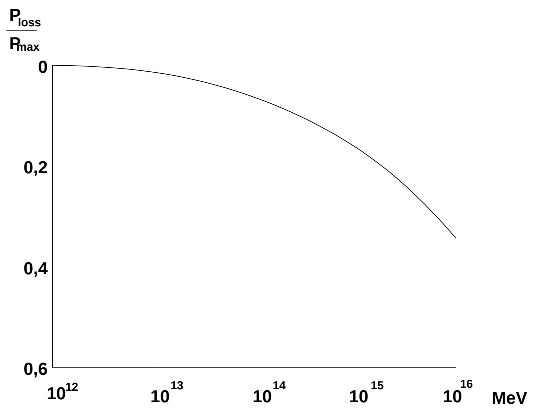


**Figure B.3:** $P_{loss}/P_{max}$ as a function of total irradiation by 1 MeV electrons

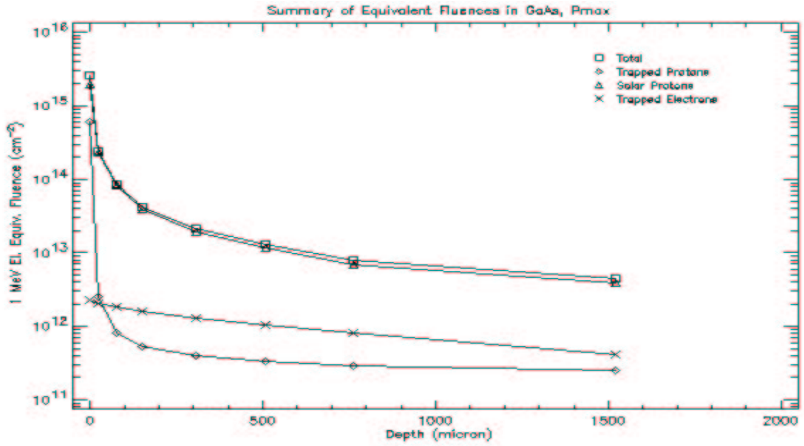


**Figure B.4:** 1 MeV equivalent radiation on GaAs vs penetration depth

a total of 10 panels, and a total area of $2 \cdot 5 \cdot 7.6\ cm \cdot 3.6\ cm = 273.6cm^2$. The mass of the solar cells is $2.4\ \frac{g}{panel} \cdot 10\ panels = 24\ g$. At beginning of life (BOL) the solar cells have an efficiency at 23.2 % of 28.0 °C. After receiving a radiation dose of $1 \cdot 10^{15} MeV$ the solar cells have an efficiency of 20.6 %. The radiation Cubesat will experience during a one year period, given in $1MeV$ equivalent for GaAs, can be seen on figure B.4.

As can be seen on figure B.4 the above mentioned radiation level $1 \cdot 10^{15} MeV$ will be reached in just under half a year. We can therefore expect a fall in the efficiency of the solar cells during the one year life expectancy of the cubesat.

The temperature dependency of the cells is very small, 0.03 % /°C . The temperatures expected are from -40 °C to 80 °C which is approximately ± 60 °C compared to the defined working temperature of 28 °C. This will give a change in the efficiency of the cells of:

$$\pm 60^0 C \cdot 0.03\%/^0 C \cdot 23\% = \pm 0.41\% \tag{B.1}$$

This means that the temperature variations will cause the cells efficiency to variate from approximately 22.8 % to 23.6 %. To minimise the reflection of sunlight on the surface material of the cells, they are coated with an anti reflective coating. This coating has an absorption of 0.9 or $90\%$. That means that the maximum amount of sunlight reflected on the surface, at any angle, will be $10\%$.

Because of lack of data on the solar cells, the maximum power points for different conditions are not known, but a qualified guess is that the maximum power point (MPP) will be at somewhere between 1.8 V and 2.2 V for a single panel.

### B.2.1 Connecting the solar panels

As known from [Emcore, 2001] the Cubesat will have a total of 10 solar panels. Since it is desired that the panels appear as one unit (4), they have to be connected in some fashion so that they only have one power output. The cells can either be connected in series or in parallel as shown on figure B.5

#### Series Connection

When connected in series, the current running through the cells will be the same for all the cells in the connection, whereas the voltage will vary from cell to cell, with the total voltage over the connection being the sum of the individual voltages of each cell. This will, compared to the parallel connection, give an overall greater voltage, and an overall smaller current. Figure B.6 shows the I/U characteristics of two panels in series, each illuminated with a different intensity.

Because of the series connection, the same current will run through both panels, and the voltage over each panel will be determined by this current, and the degree of illumination.

As can be seen on the figure B.6 [Knopf, 1999], if a large current is drawn $(I_1)$, the strongly illuminated panel $(P_1)$ will have a reasonably high voltage and the overall power from that panel will be high. The less illuminated panel $(P_2)$ on the other hand will experience a voltage drop in stead of gain due to the diode effect, which will result in loss of power. If a small current is drawn $(I_2)$, the less illuminated panel $(P_2)$ will give

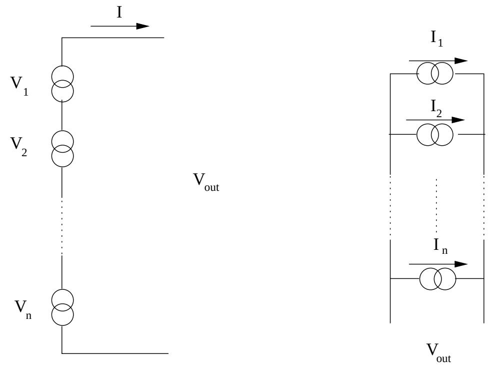


**Figure B.5:** Series (left) and parallel (right) connection of panels

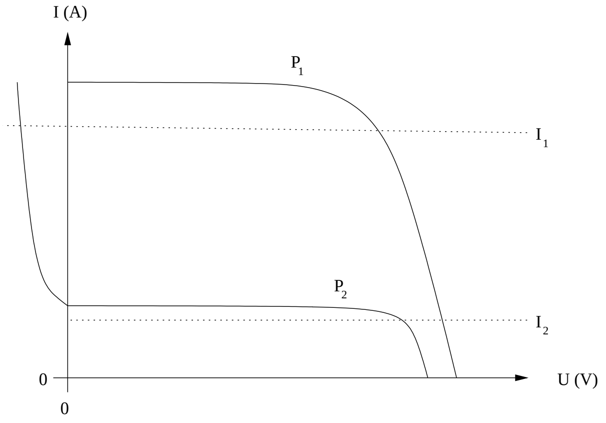


**Figure B.6:** I/U characteristics of panels with varying degree of illumination connected in series

a reasonable high voltage level, but because of the smaller current the overall power from this cell will not be very high, even though it may be close to optimal for this panel. The strongly illuminated panel $(P_1)$ will still have a high voltage, but again the low current will cause the power output of that cell to be far from optimal. The P/U characteristics of the series connection can be seen compared to the P/U characteristics of each panel in figure B.7

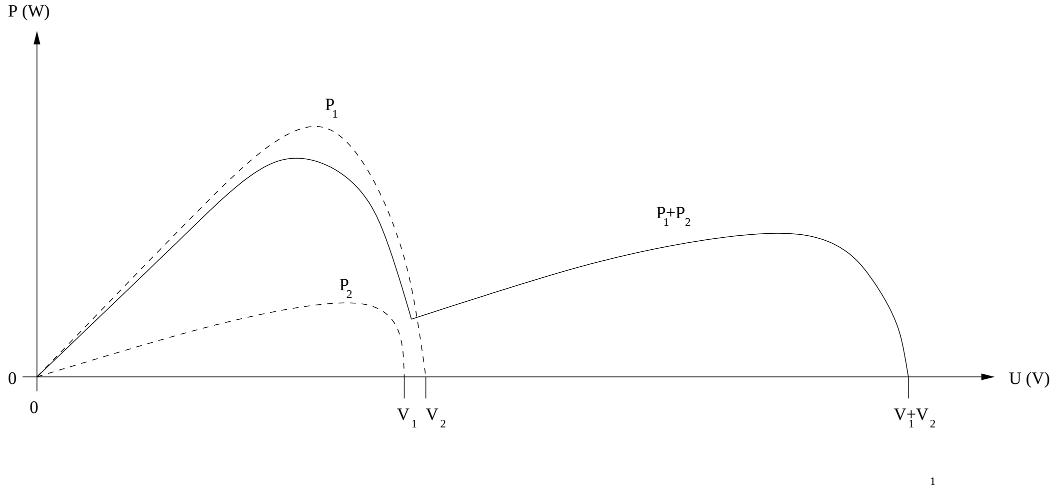


**Figure B.7:** P/V characteristics of panels, with varying degree of illumination, connected in series

It is here obvious that the maximum power of the series connection is less than the maximum power of the single most illuminated panel. This will however only be true, when big differences occur in the intensity of the illumination on the different panels, for example one panel in sun and the other in shadow.

All in all we can conclude, that when the different panels in a series connection are illuminated differently, it is not possible to get the maximum power output from all panels. Moreover, if the difference in illumination is big enough, the total power output will be less than the output would have been from the single best illuminated panel. This though can be averted with the use of diodes.

**Parallel Connection**

When connected in parallel, it is the voltage level that will be the same for all panels in the connection, whereas the current flow will be independently determined for each panel. This gives, compared to the series connection an overall greater current, and an overall smaller voltage. Figure B.8 shows the I/U characteristics of two panels in parallel, each illuminated with a different intensity.

In the parallel connection, the same voltage will occur over both panels, and the current running through each panel will be determined by this voltage, and the degree of illumination.

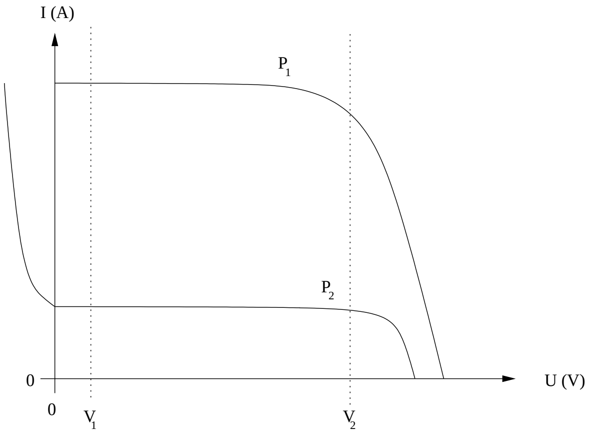


**Figure B.8:** I/U characteristics of panels, with varying degree of illumination, connected in parallel

As can be seen on the figure B.8[Knopf, 1999], the change of voltage level over the panels will effect both panels in a similar manner. A low voltage$(U_1)$ will have both panels working in a low power mode, with little power on the connection output. Operating at a higher voltage$(U_2)$ will cause both panels to operate in a higher power mode, resulting in a higher output on the power bus. Most interesting is that the point of maximum power of the two panels will be close to each other, and therefore the maximum power output on the power bus will be very close to the added maximum power of the two individual cells. This can be seen more directly in figure B.9.

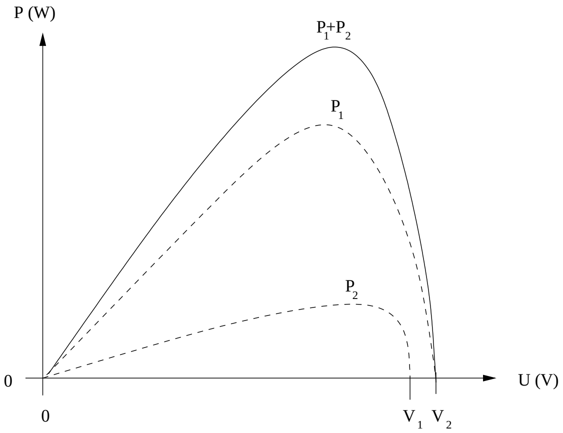


**Figure B.9:** P/U characteristics of panels, with varying degree of illumination, connected in parallel

**Safety Diodes**

Since all the panels on the Cubesat, will never be in the sun at the same time, some of them will be in shadow, and therefore function as diodes.

If the panels are connected in parallel, this would mean that the power produced by the illuminated cells would be dissipated through the unilluminated cells. To avoid this, diodes can be connected in series with the individual panels, so that no reverse current will run through the shadowed panels. Connected in this way, the diode will also prevent a defect panel shortening to ground from affecting the rest of the panels, and the following circuitry. This could for example happen by a micro meteoroid impact.

If the panels are connected in series, the affect of the shadowed panel would be to effectively shut down the current flow, and thereby the production of power, through the whole connection. The same affect could be seen by a damaged panel. To avoid a shadowed or damaged panel from disrupting the use of the rest of the panels, diodes can be connected in parallel to the individual panels, thereby allowing the current to circumvent the damaged panel in an array.

The disadvantage of using safety diodes, is the dissipation of power in the diode during ordinary running conditions. If the panels are connected in series, the diodes are connected to the individual panels in parallel, and the current through the panel array, will have to go through the diodes connected to the shadowed panels. With the panels connected in parallel, the diodes are connected in series with the individual panels, and the current from any illuminated panel will have to pass through only one diode. There is also a power loss in the reverse biased diodes, caused by the leakage current through these diodes, but this loss is usually negligible.

**Panel Connection**

The series connection is a bad idea when it comes to connecting panels with widely varying illumination. On the Cubesat there will most of the time be some panels in the sun, though with varying angle, and some in the shadow of the satellite. Therefore the panels on the different sides of the satellite will be connected in parallel. The cells on the same side, will have the same degree of illumination, and therefore do not present the same problem in a series connection. Since the batteries are to be charged at close to 8.4 V to minimise the power loss in the MPPT converter, the voltage level of the final solar array should be as close to that level as possible. Therefore the two panels on the same side are connected in series.

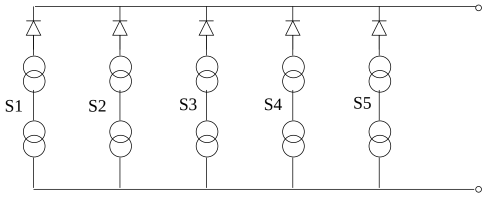


**Figure B.10:** The circuitry of the solar panels on the Cubesat

To avoid the problem with shadowed or damaged panels, each set of series connected panels, are to be connected with a series diode. The parallel diodes suggested in B.2.1 are not incorporated, for should one of the panels in the series connection become damaged, the voltage level over the other would be close to twice the optimal operating voltage, and the panel would therefore not be able to be used anyway. So in this connection the parallel diodes would not have any helpful effect. The final panel circuitry can be seen in figure B.10. The power lost in the series diode is proportional to the voltage drop over the diode, and diodes with as low a voltage drop should therefore be used. Since the effect of a current/voltage ripple on the solar panels is not known, this ripple is attempted minimised by a capacitor, as seen in figure B.10. The size of this capacitor is chosen to be 68 uF as this gives a good dampening of the ripple. The dampening affect was in a crude p-spice simulation shown to be a factor of several hundred, but since the affect of the ripple is not known in the first place, no further resources are allocated to prove or improve this simulation.

Finally it is worth mentioning that to obtain the maximum power from the solar panels, some kind of regulation is advised to keep the panels operating at their maximum power point.

# Appendix C
# Orbit Analysis

## C.1 Purpose of the Orbit Analysis

This appendix provides an analysis of the orbit of the satellite. The aims of this analysis are to:

- Derive formulas that describe the orbit of the satellite
- Analyze a set of case-orbits in order to estimate:
  - expected power input after deployment from launch vehicle while the satellite tumbles
  - expected worst-case and best-case power input after orbit has stabilized and camera has been directed towards Earth and no rotation around the camera axis is experienced

These numbers will be used as reference for the design of the power supply and rather large approximations are accepted. The following sections will first provide a model of the amount of radiation exposure of the photovoltaic cells and then this model will be utilized to calculate values for a set of cases.

## C.2 Model Derivation

The situation that is going to be analyzed is depicted in figure C.1. It can be seen that the Earth is in an elliptical orbit around the Sun. Therefore it is necessary to distinguish between the situation when Earth is in aphelion, which is when Earth is farthest from the Sun, and perihelion, which is when Earth is nearest to the Sun, i.e. it has to be figured out what radiation can be expected from the Sun at these positions.

When deriving the model it will be assumed that the satellite is in a horizontal and circular orbit around the Earth . This does not correspond to the near polar orbit that the Cubesat will be deployed in, but it will serve to find the maximum possible time of eclipse.

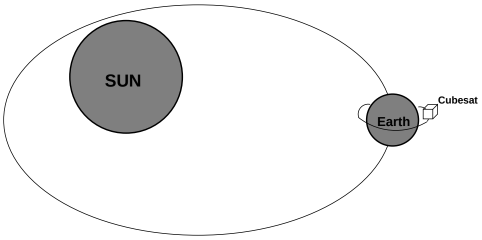


**Figure C.1:** The Earth in an elliptical orbit around Sun and the satellite in a circular horizontal orbit

The following subsections will derive simple formulas that can be used to calculate the required values that will be used for numerical simulations which are described in the next section.

### C.2.1 Intensity of Radiation at Location of Satellite

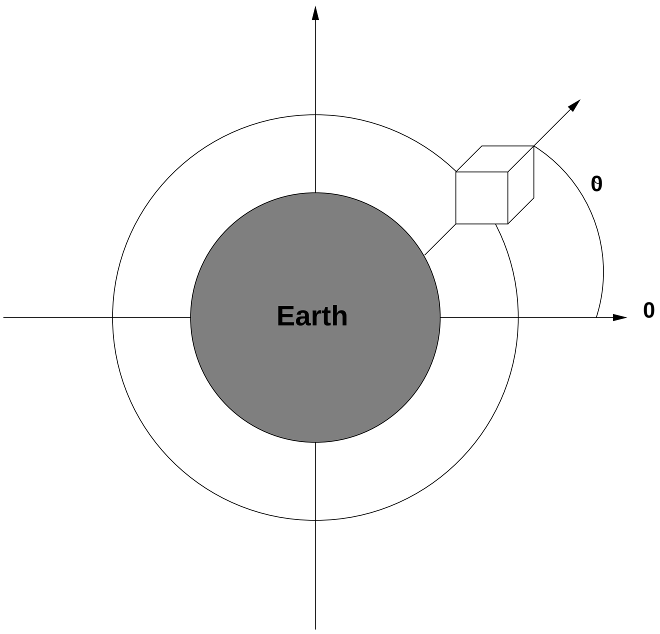
The intensity of radiation from the Sun decreases proportional to the square of the distance between the Sun and the point of measure. However since the eccentricity of the Earth-orbit is rather small it is an adequate approximation to let the intensity variate by [Lawetz, 1975]:

$$I = I_{avg}(1 - A \cdot \cos(\theta_{Earth})) \ \ [W/m^2] \tag{C.1}$$

where:
$I$: is the intensity at the point of measure [W/m$^2$]
$I_{avg}$: is the intensity at the mean Sun to Earth distance [W/m$^2$]
$A$: is the eccentricity of the orbit
$\theta_{Earth}$: Is the angular position within the orbit [rad]
If values are supplied the formula yields:

$$I = 1353\frac{W}{m^2}(1 - 0.033 \cdot \cos(\theta_{Earth})) \ \ [W/m^2] \tag{C.2}$$

This formula, which is derived for the Earth's orbit, is also valid for the satellite, but since the variation of the distance that are imposed due to the orbit of the satellite around the Earth is of no practical consequence.



htb

**Figure C.2:** Satellite orbit around Earth. $\theta$ is position angle

This is due to the fact that the distance from Sun to Earth is order of magnitudes greater that the distance from center of Earth to satellite.

In order to find minimum and maximum values it should be noted that:

$\theta_{Earth} = \pi$ at perihelion (mid winter)

$\theta_{Earth} = 0$ at aphelion (mid summer)

### C.2.2 Satellite Orbit and Shadow

Having established the intensity of sunlight that reaches the satellite, it is desirable to figure out when the satellite is in shadow and therefore not able to convert the sunlight to electricity. To this end it is needed to first establish a simple model of the satellites orbit around the Earth.

An expression for $\theta_{sat}$ as a function of time must be found. The velocity of the satellite around the Earth can be found from Newtons law of gravity [Serway, 2000]:

$$v = \sqrt{\frac{G \cdot M \cdot m}{r}} \ [m/s] \tag{C.3}$$

where:
$G$: is the universal gravitational constant $[6.670 \cdot 10^{-11} Nm^2/kg^2]$
$M$: is the mass of the Earth [kg]
$m$: is the mass of the satellite [kg]
$r$: is the distance from center of Earth to satellite [m]
The angular distance traveled can now be found by:

$$\theta_{sat} = \frac{v_{sat} \cdot t}{r_{orbit}} \ [rad] \tag{C.4}$$

where:
$t$: is time [s]
Thus the angular velocity is given by:

$$\omega_{sat} = \frac{d\theta_{sat}}{dt} = \frac{v_{sat}}{r_{orbit}} \ [rad/s] \tag{C.5}$$

And since the time of one orbit is given by:

$$t_{orbit} = \frac{2 \cdot \pi}{\omega_{sat}} \ [s] \tag{C.6}$$

Then $t$ is constrained within: $0 \leq t \leq t_{orbit}$ for one orbit.

These results can be used to develop a formula that can tell whether the satellite is in the shadow of the Earth or not. The situation is depicted in figure C.3. As can be seen from this figure three distinct situations exists:

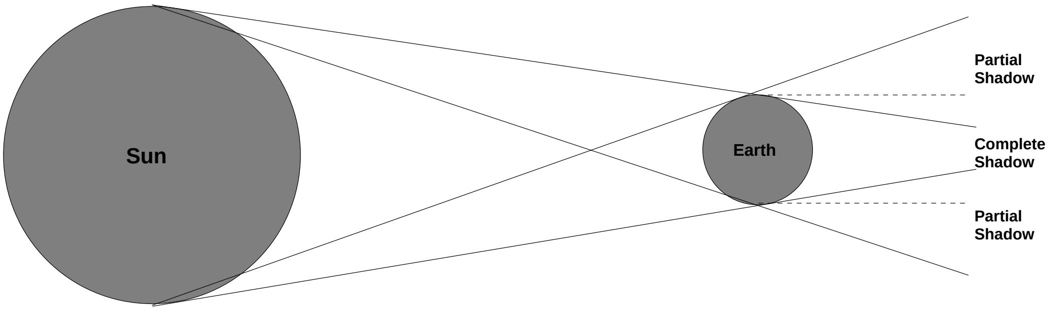


**Figure C.3:** The regions of shadow projected by Earth

- satellite partially in shadow
- satellite completely in sunlight
- satellite completely in shadow

However because the Sun is at a very great distance away from the Earth compared to the radius of Earth itself then the angular distance between the ”partial-eclipse-line” and the ”complete-eclipse-line” becomes very small, hence the distinction that the satellite is in sunlight when the numerical value of its y-component is greater than the radius of the Earth is feasible. This situation is depicted in the picture by the dashed line that protrudes from the radian of Earth.

This is can be formulated as:

$$S(t) = \begin{cases} 1 & \text{for } |r_{sat} \cdot \sin(\omega_{sat} \cdot t)| > r_{Earth} \wedge \cos(\omega_{sat} \cdot t) < 0 \\ 0 & \text{else} \end{cases} \tag{C.7}$$

This function yields one if the satellite is in the sunlight and zero if it is in the shadow.

### C.2.3 Angle Between Satellite Surfaces and Sun

Because the intensity is defined for a surface perpendicular to the Sun it is needed to derive an expression for the angle between a given surface and the Sun. To this end the orientation of the surface will be defined by the surface normal vector $\mathbf{n}$ and the orientation of the radiation from the Sun $\mathbf{s}$. The angle of declination can then be found using the definition of the scalar quantity:

$$\theta_{n-s} = \cos^{-1}\left(\frac{\mathbf{n} \cdot \mathbf{s}}{|\mathbf{n}||\mathbf{s}|}\right) \tag{C.8}$$

The power that reaches a surface can now be found by the formula [Lawetz, 1975]:

$$P = I \cdot A \cdot \cos(\theta_{n-s}) \;\; [W] \tag{C.9}$$

where:
$A$: is the area of the surface $[m^2]$

### C.2.4 Conclusion

So far the results of this analysis can be combined in the following formula that describes the amount of power that is converted to energy on the satellite:

$$P = S(t) \cdot I \cdot \eta \cdot \sum_{n=1}^{6} \underbrace{A_n \cdot \cos\left(\theta_n(t)\right)}_{\text{positive terms only}} \;\; [W] \tag{C.10}$$

where:
$S(t)$: is the derived ”shadow-function”
$I$: is a the intensity calculated for a given position of the Earth $[w/m^2]$
$\eta$: is the degree of effectiveness of the photovoltaic cells
$A_n$: is the area of the $n$th side of the satellite $[m^2]$
$\theta_n$: is the angle between the Sun and the normal-vector of the $n$th side [rad]
$t$: is the time of the orbit i.e. $0 \leq t \leq t_{orbit}$ [s]

This formula can now be used to simulate one orbit by providing the needed orbit-constants and angular velocities of the satellite. By integrating over one orbit the average input power can be found.

## C.3 Calculations for the AAU-Cubesat

In the following different orbit-cases will be analyzed to find the amount of power-input that can be expected on the cubesat. The following cases will be analyzed:

**Tumbling** After deployment from the launcher the satellite will tumble (i.e. rotate with undefined angular velocities) until the attitude control system has stabilized the spin of the satellite.

**Normal orbit worst & best case** After successful detumbling the satellite will travel with the camera side pointing towards Earth.

All simulations share the following satellite and orbit parameters:
The matlab programs can be found on the CD-ROM that is enclosed with the report[1].

| Parameter | Value |
|---|---|
| Orbit altitude | 96° |
| Orbit inclination | 600 $km$ |
| Efficiency of solar cells (see B on page 173) | 23% |
| Satellite mass | 1 kg |
| Solarcell area for each side (5 sides in all) [MK9-P22A, 2001] | $56.25\ cm^2$ |

**Table C.1:** Orbit parameters used for simulation

### C.3.1 Tumbling

The purpose of this simulation is to give an idea of the input power that can be expected when the satellite has been deployed from the launcher and until it has finished its tumbling phase. The simulation is carried out for aphelion and it assumes that the satellite is deployed in the worst possible initial conditions i.e. its orbit-plane-normal-vector is perpendicular to the plane of the Sun. This means that the satellite will experience its maximum time in eclipse during the simulated orbit. The graph from this simulation is shown in figure C.4.

The angular velocities of the rotation of the satellite used for this simulation are: $\omega_\theta = 0.007\ rad/s$ and $\omega_\phi = 0.003\ rad/s$. The average power during the simulation is 1.06 W and the average power for the period when the satellite is not in the shadow is 1.68 W. The satellite is exposed to shadow in 35 minutes out of a total orbit time of 96 minutes. This simulation has also been performed with other angular velocities and these simulations have given results that are quite similar.

### C.3.2 Normal Orbit - Worst case

After detumbling the camera will be pointing towards the Earth and for this simulation it will be assumed that the attitude of the satellite will only change (seen from the Sun) in order to keep the camera pointing to Earth. The worst case is when the orbit-plane-normal-vector is perpendicular to the sun-plane and the attitude of the satellite is such that only one side of the satellite is illuminated when the declination from the sun-plane to the satellite-plane is 0°. The result of the simulation is shown in figure C.5.

The angular velocities for this simulation are: $\omega_\theta = 0.001074\ rad/s$ (calculated from equation C.5)and $\omega_\phi = 0\ rad/s$. The average power during the simulation is 1.30 W and the average power for the period when

[1] CDROM/Matlab/orbit.m

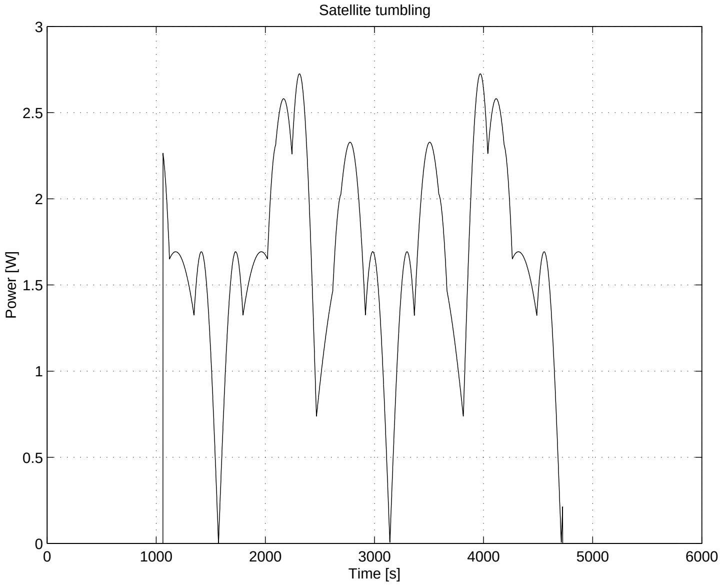


**Figure C.4:** Power input when the satellite tumbles

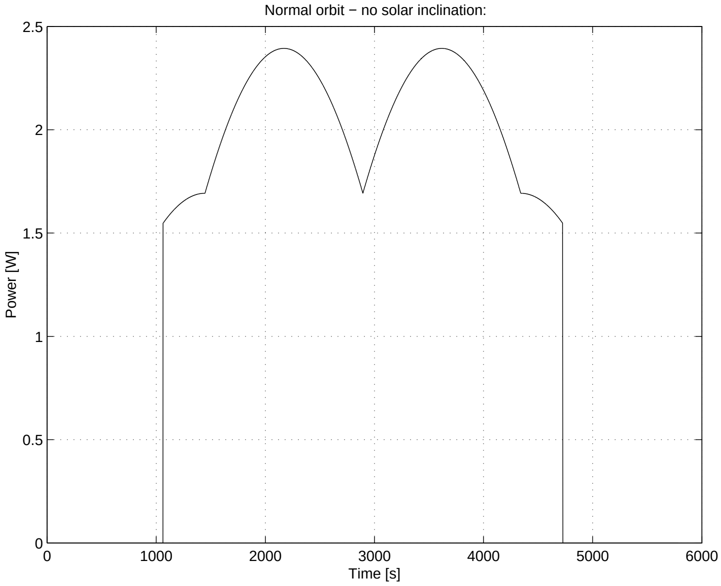


**Figure C.5:** Power input when the satellite points camera to Earth. Worst case

the satellite is not in the shadow is 2.05 W. It can be seen from the figure that maximum power is obtained when two sides of the satellite are illuminated simultaneously by the Sun.

This orbit is the absolute worst case[2]. Because the orbit of the cubesat is near sun-synchronous a power-input close to the worst case can be experienced for extended periods of time each year depending on how much the satellite orbit deviates from that of an sun-synchronous orbit.

### C.3.3 Normal Orbit - Best Case

The best case is when the orbit is such that the satellite is always illuminated by the Sun and its attitude is such that three sides are pointed towards the Sun during the complete orbit. The simulation is carried out for perihelion (winter) and the results are shown in figure C.6.

The average power during the simulation is 2.54 W. As in the case with the worst-case scenario extended periods with input close to the best-case can be expected if the exact orbit inclination is close to that of a sun-synchronous orbit.

[2] Solar eclipses are not accounted for in this appendix

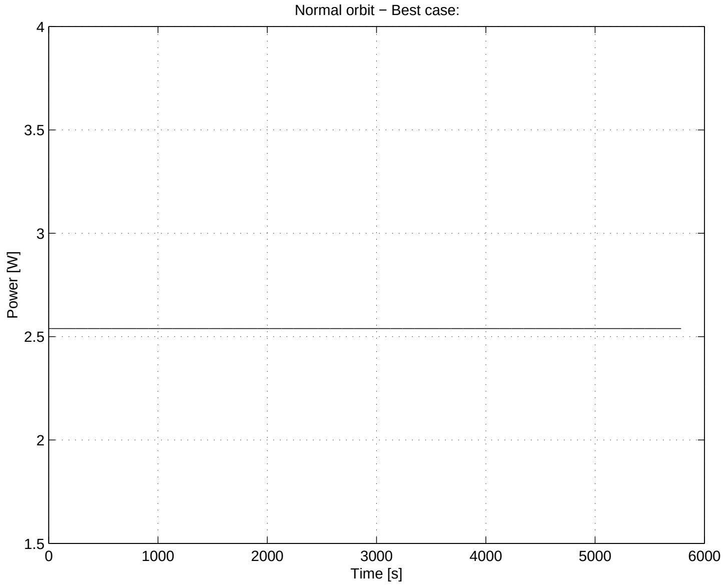


**Figure C.6:** Power input when satellite points camera to Earth. Best case

## C.4 Summary

The simulations on basis of the model has provided us with an idea of the average power that can be expected when the satellite is deployed in its orbit. The results are that one can expect about 1.06 W after deployment, when the satellite is still tumbling, about 1.3 W in a worst-case orbit with the camera pointing towards Earth and about 2.54 W in a best case orbit where the satellite is in the sun during the entire orbit.

There are however a number of influencing factors that has not been included in the model presented in this appendix. Some of these would give higher performance and other worse. Examples of these factors are:

**Better** The Earth reflects about 400 $W/m^2$ of power from the Sun in the visible spectra and 200 $W/m^2$ in the infrared spectra

**Better** Reflection from the moon has not been taken into account and eclipses of the Sun has not been considered

**Worse** Even though the solarcells have an anti-reflective coating their effectiveness are not completely independent from the angle of declination between the surface and the Sun.

Further it has been discussed that depending on the deviation of the orbit from that of a sun-synchronous there will be seasons with sunlight through the complete orbits and season with eclipses in each orbit of up to 35 minutes.

# Appendix D
# Battery Analysis and Power Budget

In this appendix an analysis of the demands on the battery of the satellite will be conducted, considering the factors influencing on the selection of the batteries. As part of this a power budget is made and it is considered how to operate the satellite in order to have sufficient power available. The power budget will be calculated bearing in mind that the power supply must have a overall efficiency of more than 80% (se section 3 on page 23). Finally the chosen batteries are discussed to reveal the necessary information.

## D.1 Requirements on the Battery of the Satellite

As the life-time of the Cubesat has been determined to one year (se chapter 3.4 on page 28)and because the solar cells can not deliver the kind of power necessary to run the satellite at all times a rechargeable battery is needed. The size, weight, endurance, capacity and voltage are all factors to be considered when finding the right battery for the Cubesat. The size and weight are very limiting factors since the Cubesat is only 10 x 10 x 10 cm and must only weigh 1 kg. The battery can only take up a small amount of this space and weight. When considering the endurance of the batteries the primary factors are the temperature characteristics and the cycle life[1]. When the satellite is in space the temperature is very varying, i.e. the temperature could be more than 80 degrees on the sun site of the sun and when behind earth less than -40 degrees (see section 2.2 on page 22). Thus the battery on the satellite must either be able to endure these temperature differences, both during charge and discharge or the thermal design of the satellite must be done in such a way that the battery will not be exposed to extreme temperatures.

Then there are the two electronic factors: capacity and voltage. The voltage is an important, but not decisive factor since a converter can be used to convert the voltage to the necessary level. A level of 3 - 4 V is normal for a single lithium-iodine battery and 1.5 V for a Nickel Cadium battery [Hansen, 2001].

The final and one of the most important factors is the capacity of the battery. A lot of different factors must be taken into consideration when chosing the capacity for the Cubesat battery. If the battery is chosen too small there will not be enough power to run the satellite and if it is chosen with a too large capacity then the battery will cost more, weigh more and take up more space than necessary.

### D.1.1 Determination the Power Budget of the Satellite

In order to do the power budget properly the orbit is divided into four phases each with its own characteristics. The devision have been done from considerations about power consumption and power input, it is however important to remember that this is a simplified way to model the power budget.

**1 - Active mode in sun** In this phase the satellite is located in the sun and thus has maximum power input from the solar array. The satellite is in active mode which means that the electronics of the satellite are working at maximum and thus have a maximum power consumption.

**2 - Idle mode in sun** In this phase the satellite has maximum power input and the electronics are working at minimum giving minimum power consumption.

**3 - Active mode in shadow** In this phase the satellite is located in the shadow without power input and the power consumption is at maximum.

**4 - Idle mode in shadow** In this phase the satellite is located in the shadow and therefore there are no power input and the power consumption is minimal.

In the first phase all the power produced by the solar cells is consumed by the electronics and the battery is being discharged. In the second phase the battery is being recharged with almost all the power produced by the solar cells. In the third and fourth phase the power input is close to zero and all power consumption in the satellite is taken from the battery. Note that the following calculations are based on the orbit analysis (see appendix C on page 179).

[1] The cycle life of at battery is defined as the number of full charge/discharge cycles the battery can sustain

The orbit time at a height of 600 km is 96 minutes (appendix C on page 179) of which the 61 minutes are spend in the sun and thus 35 minutes are spend in the shadow. Activemode is defined to be when the satellite is capable of transmitting to Denmark, i.e. that the satellite is at least above the horizon seen from Denmark. To find the time in which the satellite is above the horizon a part of formula C.7 on page 181 is used (seefigure D.1):

$$r_s \cdot \sin(90 - \theta_h) = r_e \Leftrightarrow r_s \cdot \cos(\theta_h) = r_e \tag{D.1}$$

where:
$r_s$: is the distance from center of earth to the satellite
$r_e$: is the radius of earth
$\theta_h$: is the angle between $r_s$ and $r_e$

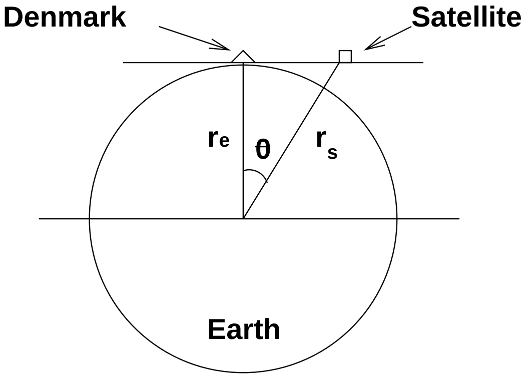


**Figure D.1:** The satellite in the horizon seen from Denmark

Using formula D.1 the angle $\theta$ can be calculated to 24°and therefore the total horizon angle is 48°. With an orbit time of 96 minutes the satellite can achieve about 15 orbits in one day. The inclination of the orbit is approximately 96°[930 and 931, 2001] which is almost perpendicular to equator and therefore the following calculation can be made. In one day the earth rotates 360°around its owe axis and therefore the rotation in one orbit at equator is:

$$\frac{360}{15} = 24\frac{degrees}{orbit} \tag{D.2}$$

Denmark, however, is not situated at equator but at about 56°lattitude, and therefore the angle between each orbit will not be as large as at equator (see figure D.2).

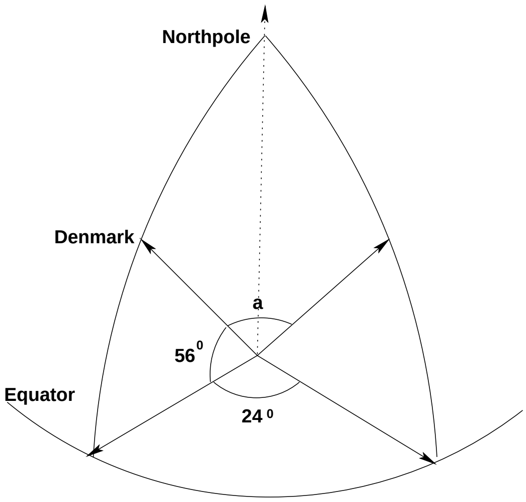


**Figure D.2:** The angular difference between two orbits, 24° at equator and a° at Denmark

If the vectors that determine the orbits at equator have the spherical coordinates: $\theta$ = 0, $\phi$ = 90, r = 1 and $\theta$ = 24, $\phi$ = 90, r = 1. The vectors that determine the orbits at Denmark have the coordinates: $\theta$ = 0, $\phi$ = 34, r

= 1 and $\theta$ = 24, $\phi$ = 34, r = 1. If these spherical coordinates are converted to rectangular coordinates it gives: $[0.56, 0, 0.83]$ and $[0.52, 0.23, 0.83]$.

Then the angle at Denmark can be calculated with formula C.8 on page 181.

$$a = cos^{-1}\frac{[0.56, 0, 0.83] \cdot [0.52, 0.23, 0.83]}{1 \cdot 1} \Rightarrow a = 11.45^0 \tag{D.3}$$

where:
$a$: is the angle between two orbits over Denmark

This implies that in the total horizon angle of 48° the satellite can achieve 4-5 orbits and thus the satellite can achieve 8-10 orbits over Denmark in one day. It is estimated that the satellite is only capable of transmitting to earth in an angle of maximum 32 °and thus only have maximum 4 and normally 3 in each period and therefore the total number of orbits above Denmark in one day is around 6. The angular velocity is $1.074 \cdot 10^{-3}$ rad/s (see equation C.5 on page 180) which equals 0.062 degree/s and therefore the active time for the 32°over Denmark can be calculated to about 10 minutes:

$$\frac{36\ degree}{0.062\ \frac{degree}{s} \cdot 60\ \frac{s}{minute}} = 9.7\ minutes \tag{D.4}$$

Therefore, of the approximately 15 orbits the satellite can achieve in one day, six of them will be close enough over Denmark to go into active mode for about 10 minutes. This means that only 2/5 of the orbits will be containing one of the active phases described above and 3/5 of the orbits will only contain the idle mode-phases, since the satellite is only transmitting and taking pictures above Denmark. It can be seen that the orbits containing active phases comes 3 and 3 and with 12 hours difference giving 3 in sun and 3 in shadow.

The power budget is calculated using a spredsheet which can be found on the enclosed CDROM [2]. At figure D.3 the calculated power budget can be seen. The power consumption of each user can be found in the Interface Description (appendix N on page 233).

### D.1.2 Determination of the Battery Capacity

To find the capacity of the battery the powerloss of the active-orbits in respectively sun and shadow is first compared to find the greatest loss and as it can be seen from figure D.3 the 3 active in the sun orbits in shadow yields a powerloss of 6094 mW while the 3 active orbits only yields a powerloss of 5310 mW. Thus the batteries must be able to store 6094 mW and therefore this is the absolute minimum capacity of the battery and to give a safe margin it is estimated that a factor 2 is appropriate giving a capacity of approximately 12 Wh.

### D.1.3 Cyclelife of the Battery

As it can be seen from the above the battery goes through 2 cycles every day (one for each 3 orbits above Denmark) and thus the cycles it goes through in one year is:

$$2\ cycles/day \cdot 365\ days/year = 730\ cycles/year \tag{D.5}$$

## D.2 The Battery on Cubesat

The chosen battery for the satellite is a lithium-iodine batter called DLP 443573 from Danionics ([DANIONICS, 2001]) with a nominal voltage level of 3.7 V (range from 3 to 4.2 V) and a capacity of 920 mAh (equal to about 3404 mWh). It weighs 26 g and the dimensions are 69 x 39 x 4.9 mm.

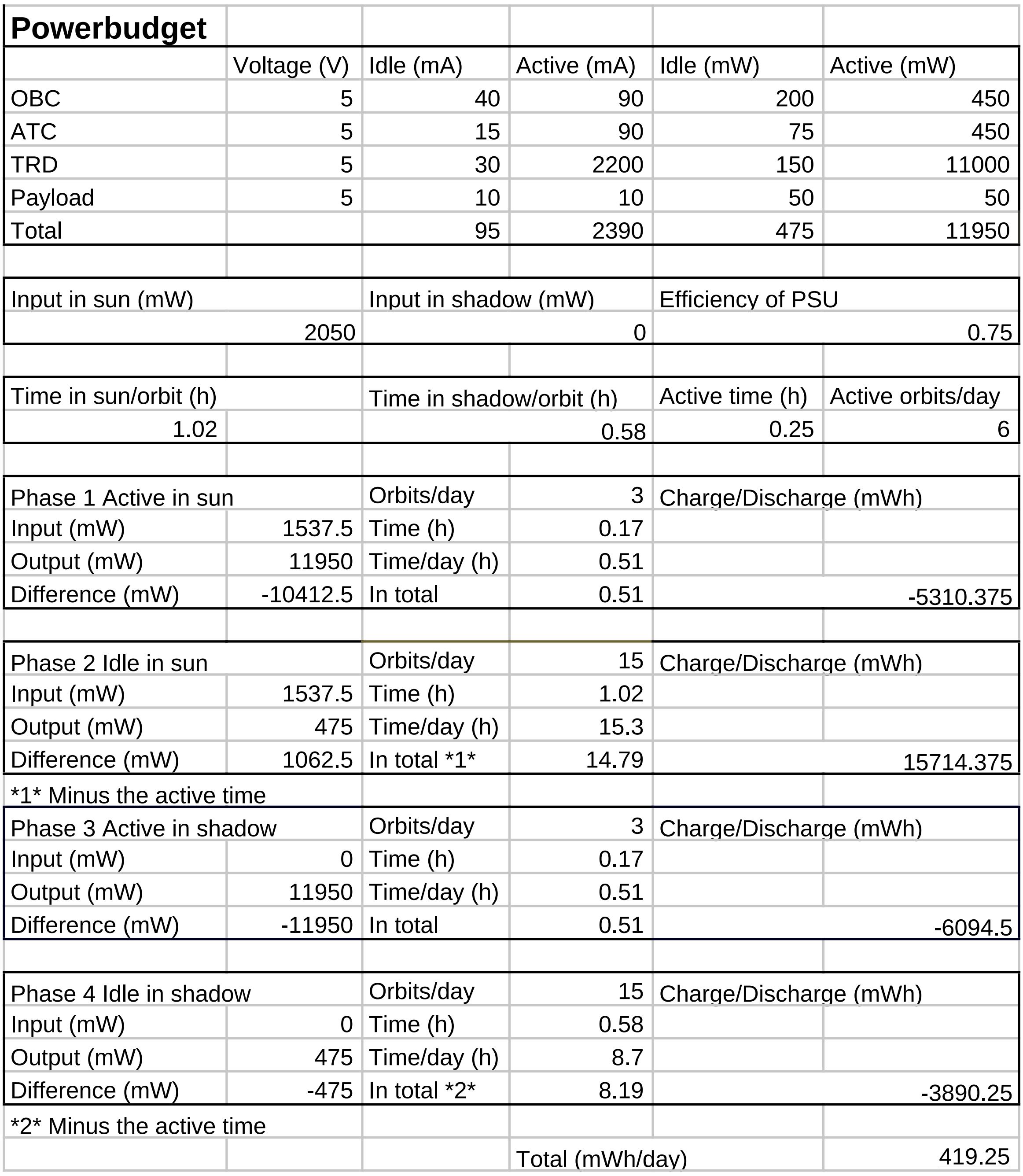
The operating temperature range of the battery is from 5 °C to 45 °C, and is best operated close to 23 °C which puts great demands on the thermal design. The battery will after 730 cycles in one year have a capacity of about 80 % equal to 736 mAh.

A quantum of four batteries will be used to fulfill the capacity-need of the satellite giving a capacity of 13616mWh, which is more than adequate.

**Coupling of the Batteries**

There are basically two different ways of couple the four batteries together: either in parallel or in series. However a hybrid of the two can also be used (see figure D.4 C).

[2] /CDROM/Powerbudget/Powerbudget.gnumeric



| Powerbudget | | | | | |
|---|---|---|---|---|---|
| | Voltage (V) | Idle (mA) | Active (mA) | Idle (mW) | Active (mW) |
| OBC | 5 | 40 | 90 | 200 | 450 |
| ATC | 5 | 15 | 90 | 75 | 450 |
| TRD | 5 | 30 | 2200 | 150 | 11000 |
| Payload | 5 | 10 | 10 | 50 | 50 |
| Total | | 95 | 2390 | 475 | 11950 |
| | | | | | |
| Input in sun (mW) | | Input in shadow (mW) | | Efficiency of PSU | |
| | 2050 | | 0 | | 0.75 |
| | | | | | |
| Time in sun/orbit (h) | | Time in shadow/orbit (h) | | Active time (h) | Active orbits/day |
| 1.02 | | | 0.58 | 0.25 | 6 |
| | | | | | |
| Phase 1 Active in sun | | Orbits/day | 3 | Charge/Discharge (mWh) | |
| Input (mW) | 1537.5 | Time (h) | 0.17 | | |
| Output (mW) | 11950 | Time/day (h) | 0.51 | | |
| Difference (mW) | -10412.5 | In total | 0.51 | | -5310.375 |
| | | | | | |
| Phase 2 Idle in sun | | Orbits/day | 15 | Charge/Discharge (mWh) | |
| Input (mW) | 1537.5 | Time (h) | 1.02 | | |
| Output (mW) | 475 | Time/day (h) | 15.3 | | |
| Difference (mW) | 1062.5 | In total *1* | 14.79 | | 15714.375 |
| *1* Minus the active time | | | | | |
| Phase 3 Active in shadow | | Orbits/day | 3 | Charge/Discharge (mWh) | |
| Input (mW) | 0 | Time (h) | 0.17 | | |
| Output (mW) | 11950 | Time/day (h) | 0.51 | | |
| Difference (mW) | -11950 | In total | 0.51 | | -6094.5 |
| | | | | | |
| Phase 4 Idle in shadow | | Orbits/day | 15 | Charge/Discharge (mWh) | |
| Input (mW) | 0 | Time (h) | 0.58 | | |
| Output (mW) | 475 | Time/day (h) | 8.7 | | |
| Difference (mW) | -475 | In total *2* | 8.19 | | -3890.25 |
| *2* Minus the active time | | | | | |
| | | | Total (mWh/day) | | 419.25 |

**Figure D.3:** The power budget

When coupled in series the voltage difference over the batteries is between 12 and 16.8 V and the capacity is 920 mAh. When coupled in parallel the voltage difference over the batteries is between 3 and 4.2 V and the capacity is 3680 mAh. In the hybrid case the voltage is between 6 and 8.4 V and the capacity is 1840 mA. In all cases the power measured in Watt is the same. In the parallel solution the battery voltage is very close to the bus voltage which gives an advantage of small power loss in the converter between them, but a drawback in difficult control of the bus voltage. In the serial solution the battery voltage is much higher than the bus voltage which gives a greater loss of power in the converter but it is easier to regulate the bus voltage. Also the larger voltage can be used as gatevoltage for the switches in the converters. However another problem arises when coupling the batteries in series because the same charge current runs through all of them and if they are not completely balanced they will be fully charged at different times. When the first batteri is fully charged the

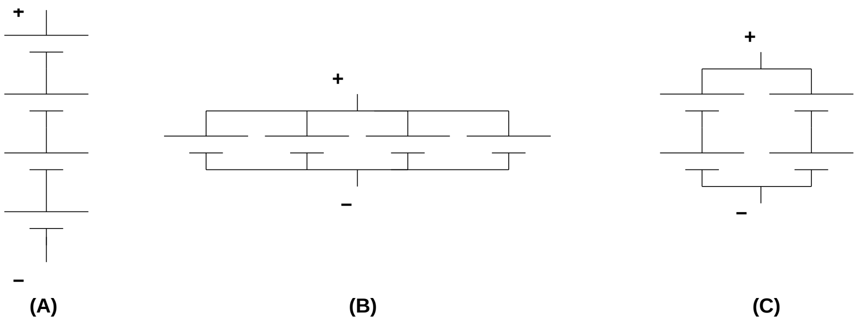


**Figure D.4:** The coupling of the batteries: (A) is in serial, (B) is in parallel and (C) is a hybrid with the batteries in series two and two

charging must be stopped, leaving the other batteries not fully charged. This problem is much smaller when choosing a parallel coupling because the batteries will then balance each other, by drawing less current if they are more charged that the rest. The choice falls on the hybrid coupling because it reduces the charging problem while still delivering a high voltage.

### Charge/Discharge

The charge method to be used with the battery is constant current between 920 mA and 460 mA, but a lower current can be used. A fully discharged battery will take about one hour to recharge at 920 mA and about two hours at 460 mA (at 23 °C). Danionics informes that the loss in a charge/discharge cycle is around 1% in the beginning of the battery life, but the loss rises slowly as the battery goes through its cycle life. The inner resistance of the battery is normally in the range 30 - 50 mΩ.

When the battery is fully charged the charging must be stopped or otherwise the battery may explode or burst into fire. Therefore the charging must be stopped when the voltage across a single battery reaches 4.2 V which is the upper voltage limit of the battery.

The discharge current must at maximum be 1840 mA continuously, but a larger current (4600 mA) can be drawn momentarily (in 30 seconds). In both charge and discharge the battery must be secured, in order to stay within the described voltage- and current limits.

# Appendix E
# Switch Realization

## E.1 Purpose

The purpose of this appendix is to describe the operation of the electronic switches that are utilized in the design of the two DC/DC converters that are developed in this project. Both the physical operation and the switching losses associated with the operation will be described.

The switches that are to be analyzed here are the MOSFET-switch and the diode-switch. Both are Single-Pole-Single-Throw (SPST) switches i.e. they are able to either act as a connection or as a disconnection. Together they can be used to implement a controllable Single-Pole-Double-Throw (SPDT) switch. Figure E.1 shows the ideal implementation of the two mentioned switch types.

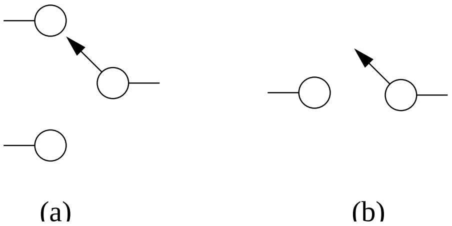


**Figure E.1:** Ideal switches. (a) SPDT (b) SPST

There are many factors that contribute to losses during switching of either the MOSFET or the diode switch. In the following losses due to the following phenomena will be discussed:

- Diode reverse recovery charge
- Losses due to diode equivalent capacitor
- MOSFET gate loading charge
- Losses due to inductive loading of the transistor

## E.2 Freewheeling Diode Switch

A diode will conduct current when exposed to a positive voltage-drop and block current when exposed to a negative voltage drop. It can therefore be used as a switch. Since the diode is controlled exclusively by the voltage of the two terminals it is called a freewheeling diode switch. The device symbol and terminal names are given in figure E.2.

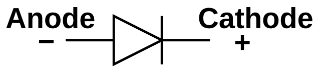


**Figure E.2:** Circuit symbol of a Diode

### E.2.1 Reverse Recovery Charge

The basic diode is a bipolar device and therefore it uses both minority and majority carriers for basic operation[1]. The typically power diode has three regions ($p - n^- - n^+$) compared to the two regions of the standard PN-signal-diode. When the diode is forward biased and operating in steady state an amount of minority carriers are present in the vicinity of the two junctions of the diode. When the diode becomes reverse biased this minority charge must be removed.

Therefore, when the diode becomes reverse-biased a current flows into the cathode of the device until all minority charge has been swept away from the junctions or been removed due to recombination. The waveforms under turn-off operation is depicted in figure E.3

At time $t_0$ a negative voltage is applied across the terminals of the diode, but the voltage-drop across the ($p - n^- - n^+$) junctions remain zero until the minority charge stored in the $n^+$ region has been removed; This happens at $t_1$. The current that floats in the interval $t_0 - t_1$ is used to remove the stored charge and it is lost.

[1] For the PN-junction holes are minority carriers and electrons majority carriers

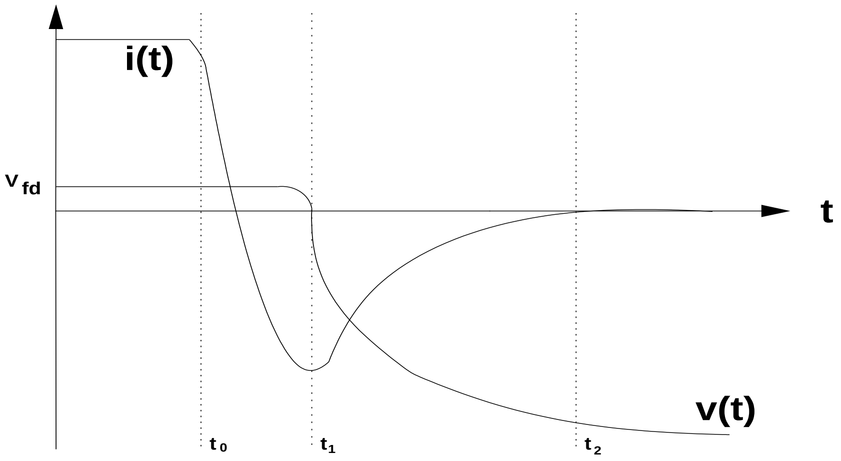


**Figure E.3:** Diode waveforms under turn-off operation

In the interval $t_1 - t_2$ the voltage-drop across the junctions falls and a current floats into the cathode in order to remove remaining minority charge in the $n^-$ region. This charge is also lost. When this charge has been removed completely the diode reaches steady-state and conducts no current, this happens at $t_2$.

The energy that is lost in the diode during the turn-off transition is given by the formula [Erickson, 1999] (page 100):

$$W_d = Q_r \cdot V_d \quad [J] \tag{E.1}$$

Where $Q_r$ is the device specific recovery charge and $V_d$ is the reverse voltage that is applied at the diode terminals. To obtain the average switching loss power the following formula is evaluated:

$$P_d = W_d \cdot f_{sw} \quad [W] \tag{E.2}$$

Where $f_{sw}$ is the switching frequency.

### E.2.2 Diode Equivalent Parallel Capacitor

The typical power-diode will exhibit some amount of input capacitance that may incur losses. In the Diode-transistor implementation of a SPDT switch in e.g. a Buck-converter the input capacitance is in effective parallel with the diode. When the diode turns on, the equivalent capacitor is discharged and this charged is lost. The energy lost in the equivalent capacitor is [Erickson, 1999] page 100:

$$W_{eq-cap} = \frac{1}{2} \cdot C_{eq} \cdot V^2 \quad [J] \tag{E.3}$$

Where $V$ is the applied voltage and $C_{eq}$ is the diode equivalent capacitance. The powerloss due this phenomena is:

$$P_{eq-cap} = \frac{1}{2} \cdot C_{eq} \cdot V^2 \cdot f_{fs} \quad [W] \tag{E.4}$$

### E.2.3 Schottky-Diode

The Schottky diode is a device with the same basic operation characteristics as a typically PN-diode, but the Schottky diode is fabricated by placing a metal-film in contact with a semiconductor such as a P-layer. This makes the Schottky-diode a majority carrier device which mean that the internal operation is controlled mainly by the majority carriers. Therefore no reverse current is needed in order to remove minority carriers from the device. In effect no loss is associated with turn-off operation and faster switching times can be achieved which in effect reduces the loading off the rest of the circuitry during switching.

In addition the forward voltage drop of the schottky diode is less than that of a typically PN-diode which incurs less conduction loss, on the other hand the off resistance off the Schottky diode is less than that of a PN-diode leading to a higher reverse leakage current.

## E.3 The MOSFET switch

The MOSFET switch utilizes a Metal-Oxide-Semiconductor-Field-Effect Transistor in order to vary the line resistance between ideally 0 $\Omega$ and $\infty$ $\Omega$. In effect the in- and output terminals will be either short-circuited or disconnected. In order to control the state of the device a control signal is applied at the gate terminal of the transistor. The device symbol and terminal names are given in figure E.4. The diode is included in the device to protect it from static charges.

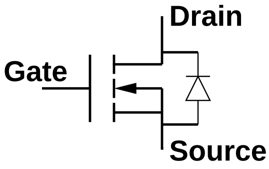


**Figure E.4:** Circuit symbol of a Power MOSFET transistor

### E.3.1 Gate Charge Loss

When a sufficient amount of charge is present at the Gate-terminal of a MOSFET transistor a conducting channel from drain to source is induced because of the field effect. When the transistor is switched off and the gate-terminal discharged this charge is lost. Therefore a source off switching loss is the stored gate charge. The energy lost is therefore:

$$W_{gate} = Q_g \cdot V \quad [J] \tag{E.5}$$

Where $Q_g$ is the device specific needed gate-charge and $V$ is the gate voltage that is applied when switching the transistor off. The power lost due to gate-discharging is then:

$$P_{gate} = Q_g \cdot V \cdot f_{sw} \quad [W] \tag{E.6}$$

### E.3.2 Losses due to Inductive Loading of the Transistor

The voltage across and current through the source-drain terminals of the MOSFET-transistor cannot change momentarily. In the following the waveforms of the switching transistor that has an inductive load connected to the drain terminal as in the Buck-converter will be analyzed. In order to simplify the analysis the inductor current $I_L$ is assumed constant over the complete switching period, all voltages and currents are depicted as piecewise linear and the diode is assumed to be ideal. The waveforms are depicted in figure E.5 (a) for turn on operation.

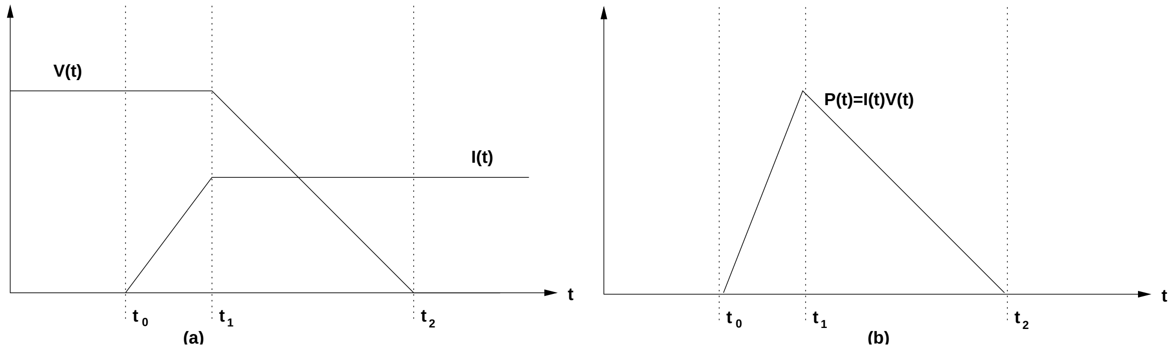


**Figure E.5:** Transistor waveforms during turn on. (a) voltage and current, (b) power dissipation

From time $t_0$ to $t_1$ the gate capacitor is charged and current begin to flow into the drain terminal oftransistor until the current reaches the level of the inductor current $I_L$. Now as current flows into the transistor the drain to source capacitance is discharged and the voltage across the transistor drops to zero. The power dissipated by the transistor in this interval is: $p(t) = i(t) \cdot v(t)$ as depicted in figure E.5 (b). The area of this triangle is the energy lost during switching on. Because of the piecewise approximation this area is given by [Erickson, 1999] page 96:

$$W_{on} = \frac{1}{2} \cdot V \cdot I_L \cdot (t_2 - t_0) \quad [J] \tag{E.7}$$

When the transistor turns off the reverse process takes place. That is the drain to source capacitance is charged while the voltage rises and the current ceases to flow as the gate to source capacitance is discharged. Therefore the loss $W_{off}$ associated with turn-off operation is equivalent to equation E.7. However there may be practically differences between the turn-on and turn-off time because of the gate-driver circuitry.

All in all the power lost due to inductive loading is given by:

$$P = (W_{on} + W_{off}) \cdot f_{sw} \quad [W] \tag{E.8}$$

# Appendix F
# Design of Inductor Coil

## F.1 Purpose

In this appendix a small introduction to fundamental magnetic theory for inductors is first given and then the losses in wires as well as in cores are discussed. Finally a design procedure for inductors is introduced as a step by step design method. As sources for this appendix [Ebert, 1998a], [Ebert, 1998b] and [Erickson, 1999] are used.

## F.2 Magnetic Inductor Theory

The magnetic flux $\Phi$ (equivalent to current) of a magnetic field produced by a inductor is defined as (assuming that the flux density $B$ is uniform, equivalent to current density) [Ebert, 1998b] page 83:

$$\Phi = B \cdot A \quad [Wb] \tag{F.1}$$

where:
$A$ is the area where $\Phi$ passes through [$m^2$]

The electronic equivalent circuit for the physical inductor and core can be seen in figure F.1.

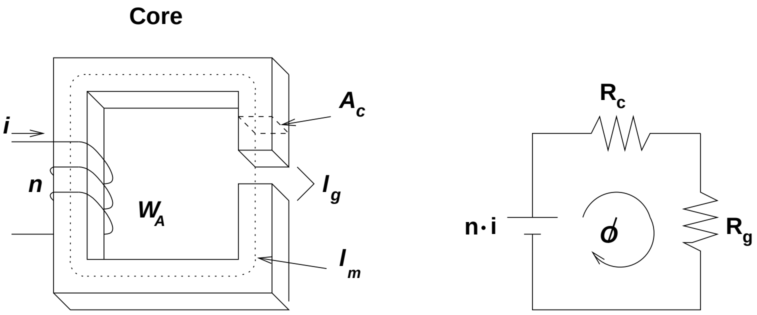


**Figure F.1:** To the left an inductor with core. To the right a electric equivalent circuit of an inductor with core

The symbols in the figure means:

- i; current in the inductor [A]
- n; number of windings [w]
- $W_A$; area in the middle of the core [$m^2$]
- $A_C$; cross section area of the air gap [$m^2$]
- $l_g$; length of the air gap [m]
- $l_m$; length of the wire [m]
- $R_c$ & $R_g$ ; reluctance [A/Wb]

From this figure it can be seen that the magnetic flux in the core is determined by the current in the wire and the number of windings, as well as the magnetic reluctance in the core and air gap. Normally the core reluctance is much smaller than the magnetic reluctance of the air gap and can therefore be ignored. This leads to the following equation for a core with air gap [Ebert, 1998b] page 87:

$$\begin{aligned} ni &= \Phi(R_g + R_c) \Rightarrow \qquad (F.2)\\ ni &= \Phi R_g \quad [Aw] \qquad (F.3)\end{aligned}$$

If a core without air gap is used the equation will look like this:

$$ni = \Phi R_c \quad [Aw] \tag{F.4}$$

The resistivity $\rho$ of a wire determines the resistance $R$ with the following equation [Erickson, 1999] page 507:

$$R = \rho \frac{l_w}{A_w} \quad [\Omega] \tag{F.5}$$

where:
$l_w$ is the length of the wire (see figure F.2) [m]
$A_w$ is the cross section of the conduction part of the wire (see figure F.2) $[m^2]$
The value of $\rho$ for copper is $1.7 \cdot 10^{-8}\ \Omega m$ at room temperature and about $2.3 \cdot 10^{-8}\ \Omega m$ at 100 °C [Erickson, 1999] page 474.

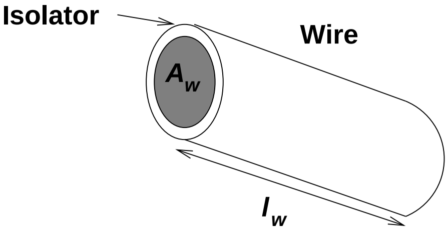


**Figure F.2:** A cross cut of a wire with the length $l_w$ and the cross section area $A_w$

The equivalent magnetic reluctance is given by [Ebert, 1998b] page 84:

$$R = \frac{l_w}{\mu A_w} \quad [A/Wb] \tag{F.6}$$

where:
$\mu$ is the permability of the material [H/m]

It can then be seen that the difference between using a core with or without air gap is that without air gap the resistance of the inductor is dependent on the permability of the core and with air gap the resistance is dependent of the permability of the air. Thus it is preferable to use a core with an air gap since the permability of the core is temperature depended to much larger extend than the permability of the air [Erickson, 1999].

To find the inductance of the inductor Faradays law is used, which states:

$$v(t) = n\frac{d\phi}{dt} \quad [V] \tag{F.7}$$

If equation F.2 and F.6 are substituted into this equation it gives:

$$\begin{aligned} v(t) &= n\frac{d(\frac{ni}{R_g})}{dt} \Leftrightarrow \\ v(t) &= \frac{n^2}{R_g}\frac{di}{dt} \Leftrightarrow \\ v(t) &= \frac{n^2\mu_0 A_g}{l_g}\frac{di}{dt} \quad [V] \end{aligned} \tag{F.8}$$

where:
$A_g$ is the cross section area of the air gap, equal to $A_c$ of the core (see figure F.1) $[m^2]$
$\mu_0$ is the permability of vacuum used as permability for the air gap [H/m]

If the voltage-current relation for a inductor is compared with equation F.8 the inductance can be derived:

$$\begin{aligned} v(t) &= L\frac{di}{dt} \Rightarrow \\ L &= \frac{n^2\mu_0 A_g}{l_g} \quad [H] \end{aligned} \tag{F.9}$$

## F.3 Losses in Inductors

An inductor can be equated by an inductor and a resistor in series, as it can be seen on figure F.3. The value of this resistor at DC can be described by equation F.5.

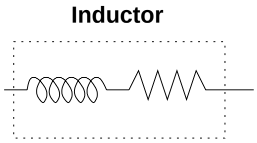


**Figure F.3:** An inductor equivalence by an inductor and a resistor

To describe the power loss in this resistor three effects will be introduced:

- Skin effect
- Proximity effect
- Core loss

### Skin Effect

If the operating frequency rises the resistance also rises. This is partly caused by an effect called current repression or the skin effect, which is when the current only flows at the surface of the wire and to a certain depth called the penetration depth ($\delta$). This effect is illustrated in figure F.4 and is described in equation F.10.

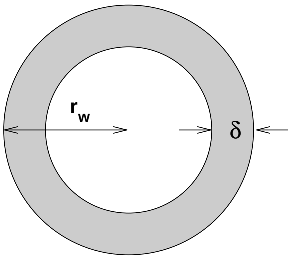


**Figure F.4:** The penetration depth seen in a cross section of a wire

$$\delta = \sqrt{\frac{\rho}{\pi\mu f}} \quad [m] \tag{F.10}$$

From equation F.10 it can be seen that the penetration depth decreases as the frequency rises and thus the conducting area decreases. Hence from equation F.5 it can be concluded that the resistance rises as the frequency rises giving a greater loss, which can be expressed as [Ebert, 1998a]:

$$R_{ac} = \frac{l_w \rho}{2\pi\delta r_w} = \frac{nMLT\rho}{2\pi\delta r_w} \quad [\Omega] \tag{F.11}$$

where:
$MLT$ is the Mean LengTh of wire per turn [m]

Therefore, the loss in the wire can be calculated as:

$$P_{wire} = R_{ac} I_{rms}^2 \quad [W] \tag{F.12}$$

Where:
$I_{rms}$ is the root mean square value of the AC-current that flows in the wire [A]

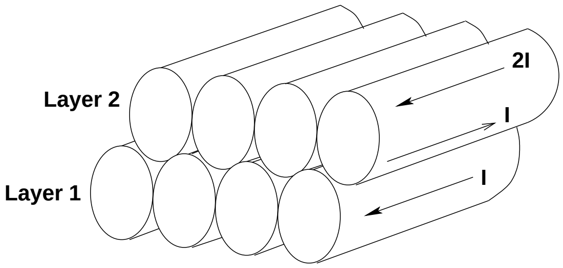


**Figure F.5:** The proximity effect in 2 layers of wire

**Proximity Effect**

The proximity effect is generated when the wire is placed in several layers around the core. In the first layer a current I flows and since the wire in the layers are serially connected the same current will flow in the second layer. The problem arises when Lenz's law is applied stating that the current I in the first layer will induce a equal, but opposite current in lower part of the second layer. Therefore, to maintain the net current of I in the second layer the current flowing in the upper part must be 2I (see figure F.5).

The same procedure repeats in all the following layers, producing a higher and higher current flow. From equation F.12 it can be seen that because the loss in the wire rises with the square of the current flow, given that the power loss in the first layer is P then the power loss in the second layer induced by the first layer is also P. Also in the second layer is the power loss from the 2I flowing in the upper part which contributes with a power loss of 4P, giving a total loss in the second layer of 5P.

The total increase in power loss produced by the proximity effect can be expressed as [Erickson, 1999]:

$$P_{ac} = P_{wire}\frac{M}{3}(2M^2 + 1)[W] \tag{F.13}$$

where:
$M$ is the number of layers

This effect however only takes place when the penetration depth is much smaller than the wire radius, i.e. at high frequencies. At DC the penetration depth is equal to the wire radius and then the same current will flow everywhere in the wire.

**Core Loss**

The core loss comes from the relation between the magnetic field (H) and the magnetic flux density (B), which is determined by the permability of the core material:

$$B = \mu H \quad [T] \tag{F.14}$$

If the core material is air and the permability thereby equals $\mu_0$ then the relationship is completely linear (see figure F.6-a), but as soon as another material is introduced as core the relation looks completely different (see figure F.6-b). As it can be seen the relationship is dependent on saturation and hysteresis. According to [Erickson, 1999] the core loss can be described as:

$$W_{core} = V_{core} \cdot A_{B-H} \quad [J] \tag{F.15}$$

where:
$W_{core}$ is the energy lost in each hysteresis cycle [J]
$V_{core}$ is the volume of the core (equals $A_c l_m$) $[m^3]$
$A_{B-H}$ is the area of the hysteresis loop [TA/m]

To obtain the total core loss, equation F.15 is multiplied with the number of cycles per second, i.e. the frequency yielding:

$$P_{core} = V_{core} A_{B-H} f \quad [W] \tag{F.16}$$

where:
$f$ is the switch frequency [Hz]

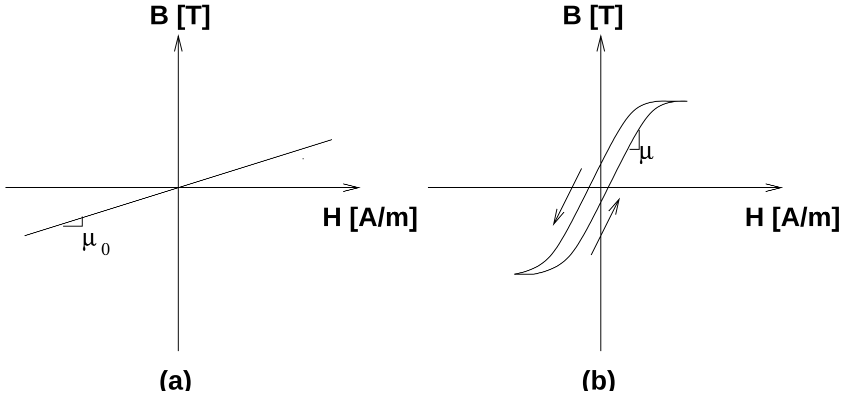


**Figure F.6:** The B-H field relationship (a) in air and (b) in a core material

According to [Erickson, 1999] the core loss for a given core of ferrite can be calculated with the following equation:

$$P_{core} = K_{fe} B_{max}^{\beta} A_c l_m \quad [W] \tag{F.17}$$

where:
$K_{fe}$ is a proportionality constant which increases with the frequency (can be determined from the core data sheet)
$\beta$ is another material constant which is determined from the core data sheet, a typical value is between 2.3 and 2.7

## F.4 Design Method

A method for inductor design will be introduced in the following. The method will introduce a step by step procedure dividing the method into 4 steps:

1. Determination of the core size.
2. Determination of the size of air gap.
3. Determination of the number of windings.
4. Determination of the wire size.

**Determination of Core Size**
The first thing to do is to choose a core large enough to satisfy the following equation:

$$K_g \geq \frac{\rho L^2 I_{max}^2}{B_{max}^2 R K_U} \quad [m^5] \tag{F.18}$$

where:
$K_g$ is the geometerial constant $[m^5]$
$I_{max}$ is the peak current in the inductor [A]
$B_{max}$ is the peak flux density in the core [H]
$R$ is the winding resistance [$\Omega$]
$K_U$ is the fill factor, i.e. how large percentage of the winding area ($W_A$) that is filled with conduction wire. According to [Erickson, 1999] page 507 a typical fill factor for a small inductor is between 0.5 and 0.7.

The geometerial constant is a way of describing the electrical size of the core and it can be described with the following equation:

$$K_g = \frac{A_c^2 W_A}{MLT} \quad [m^5] \tag{F.19}$$

where:
$W_A$ is the effective winding area (see figure F.1) $[m^2]$

### Determination of Air-gap Size

Secondly the size of the air gap must be determined which is done with the following equation:

$$l_g = \frac{\mu_0 L I_{max}^2}{B_{max}^2 A_c} \quad [m] \tag{F.20}$$

where:
$l_g$ is the length of the air gap in the core (see figure F.1) [m]
$\mu_0$ is the permability of vacuum [H/m]

### Determination of Number of Windings

The third step is to determine the number of windings around the core which can be done with two different equations. The first uses the parameters from the core selection:

$$n = \frac{L I_{max}}{B_{max} A_c} \quad [w] \tag{F.21}$$

Another equation that can be used is the following [PHILIPS, 2000] page 10:

$$n = \sqrt{\frac{L}{A_L}} \quad [w] \tag{F.22}$$

where:
$A_L$ is the core factor that depends on the size of the air gap and which determines how many Henry each winding gives on the selected core [H/w]

### Determination of Wire Size

The fourth step is to evaluate the wire size, which is done with the following equation:

$$A_w = \frac{K_U \cdot W_A}{n} \quad [m^2] \tag{F.23}$$

# Appendix G
# I$^2$C Protocol

## G.1 Overview

In this appendix the protocol for the communication on the I$^2$C bus between the PSU and the OBC is defined. This includes a description of all commands, and data that can be send both ways.

## G.2 Requests from OBC

The OBC can issue 6 different requests with each will result in a different response from the PSU, which respectively are shown in the table below:

| Request | OBC request | PSU response | Module |
|---|---|---|---|
| 1 | Set boot port to PROM | Boot port is set to PROM | 31 |
| 2 | Set boot port to EEPROM | Boot port is set to EEPROM | 30 |
| 3 | Reset Watchdog | External Watchdog is reset | 29 |
| 4 | Turn on specific subsystem | Specific subsystem are turned on | 25-28 |
| 5 | Turn off specific subsystem | Specific subsystem are turned off | 21-24 |
| 6 | Send specific housekeeping | Requested housekeeping are send | 1-9 |

### Set boot port to PROM

When the OBC issues the request “Set boot port to PROM” with module number 31 and data length 0 the PSU sets P3.0=0.

### Set boot port to EEPROM

When the OBC issues the request “Set boot port to EEPROM” with module number 30 and data length 0 the PSU sets P3.0=1.

### Reset Watchdog timer

When the OBC issues the request “Reset Watchdog timer” with module number 29 and data length 0 the PSU must reset the watchdog register. Note that the timer must also always be reset after any other requests from the OBC.

### Turn on subsystem

When the OBC issues the request “Turn on specific subsystem” with a module number from 25 to 28 and data length 0 the PSU must turn the specific subsystem on. The subsystem are defined by the module number:

| Module number | 25 | 26 | 27 | 28 |
|---|---|---|---|---|
| Subsystem | OBC | ACS | CAM | TRD |

### Turn off subsystem

When the OBC issues the request “Turn off specific subsystem” with a module number from 21 to 24 and data length 0 the PSU must turn the specific subsystem off. The subsystem are defined by the module number:

| Module number | 21 | 22 | 23 | 24 |
|---|---|---|---|---|
| Subsystem | OBC | ACS | CAM | TRD |

### Send Housekeeping Information

When the OBC issues the request “Send specific housekeeping” with a module number form 0 - 17 and a data length 0, the PSU must return the requested data. The housekeeping data are defined by the module number:

| Housekeeping | Module number | Data length |
|---|---|---|
| Battery voltage & MPPTC current out | 1 | 4 bytes |
| Solar cells current 1 & 2 | 2 | 4 bytes |
| Solar cells current 3 & 4 | 3 | 4 bytes |
| Solar cells current 5 & voltage | 4 | 4 bytes |
| Bus voltage & PSU current | 5 | 4 bytes |
| OBC & CAM & TRD & ACS temperatur | 6 | 4 bytes |
| PSU & T6 & T7 temperatur & Load status | 7 | 4 bytes |
| OBC & CAM current | 8 | 4 bytes |
| TRD & ACS current | 9 | 4 bytes |

The load status uses a 1 for load on and 0 for load off and is depicted at figure G.1.

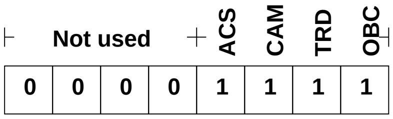


**Figure G.1:** The load status byte

## G.3 PSU messages to OBC

The PSU only have two messaged that it can send to the OBC and that is whether the received data was valid or not:

| Message | Module number | Reaction from OBC |
|---|---|---|
| Data invalid | 20 | OBC retransmit the data |
| Data valid | 19 | OBC does nothing |

# Appendix H
# Project Test Plan

## H.1 Introduction

This project test plan will specify how tests will be conducted throughout this project. Two kinds of tests are discussed, these are the module-tests that are performed throughout the project and the system test that is performed after module integration has taken place.

This test plan has been inspired by the IEEE standard: ”IEEE829-1998: IEEE Standard for Software Test Documentation”[IEEE, 1998b], but it is not specified in accordance with it.

### H.1.1 Objectives

The objectives of this system test plan is to:

- identify which types of tests should be done during the project
- describe the contents of different types of test documents
- describe the overall approach used for testing
- identify who will be responsible for test planning, execution and documentation
- provide a schedule for test planning and work

## H.2 Test Items

As stated in the introduction of this document two kinds of testing will be described: Module tests and the system test. A module test will be specified and conducted for each of the modules identified in the system analysis chapter (refer to chapter 4 on page 31).

The system test will be conducted in order to ensure that the product fulfills all requirements of the requirements specification as specified in chapter 3 on page 23.

Module integration testing will also take place, but these tests will not be formalized or documented and therefore not discussed in this test plan. This choice has been made to conserve space and time.

## H.3 Test Documentation Definitions

Different types of documents are worked on as part of the testing process. The following subsections each describe a specific type of test document with regard to its contents and purpose. It will be described how these document types are used in the test process.

### H.3.1 Requirements Specifications

The “Requirements Specification” identifies all requirements that apply to the specific test item. The system test will be tested in accordance with the ”System Requirements Specification” and each module will include a ”Module Requirements Specification” as the first section of the corresponding chapter.

The requirements specification is considered a part of the test documentation since it serves as the foundation for all test related work.

### H.3.2 Test Specification

The test specification describes how the test should be conducted through a set of test-cases that must be carried out by the people who conducts the test. All test specifications for both the system test and the module tests are enclosed as appendices of the report.

Each test specification must contain the following parts:

**Test Specification Identifier** Should identify the test item that the specification is designed to test. A reference to the corresponding ”Requirements Specification” should be given

**Test Environment** Should describe all equipment and documents that are needed in order to carry out the test

**Test Approach** Should describe the logical build-up of the test e.g. whatever interfaces or functionality will be tested first

**Test Case Specification** Describes each test case that is to be carried out. Each test-case should be uniquely identified and each test-case must contain the following points:

- Test item identifier (states exactly what is tested)
- Necessary equipment to complete the test procedure
- Test procedure description
- Expected output

### H.3.3 Test Log

During testing a log is written by the test crew. This log should describe what actions are taken by the test crew and what results are gained. Furthermore, the log must include information about date and time of the test and it should include names of the test crew. The logs will not be included in the report, but they will be stored in an archive for later reference if needed.

### H.3.4 Test Incident Reports

If a test is terminated because a test-case does not yield the expected output this should be written in the test-log and the occurrence should be described in an incident report, which describes all actions and circumstances that led to the failure of the test-case.

The incident report is then handed to the people responsible for design of the specific module (or system) in order to help them identify the problem. When the designers have resolved the incident the complete test-procedure is reiterated from start.

If one or more incidents have not been resolved before the project is due then the non resolved reports are included in the report in order to document the shortcomings of the product.

## H.4 The Test Process

The work that is to be done in order to prepare and perform tests is not only situated after completed design and implementation, but it is distributed throughout the entire development process. This process is sketched as an UML activity diagram in figure H.1. This diagram focuses on test work and leaves out all details regarding design and implementation.

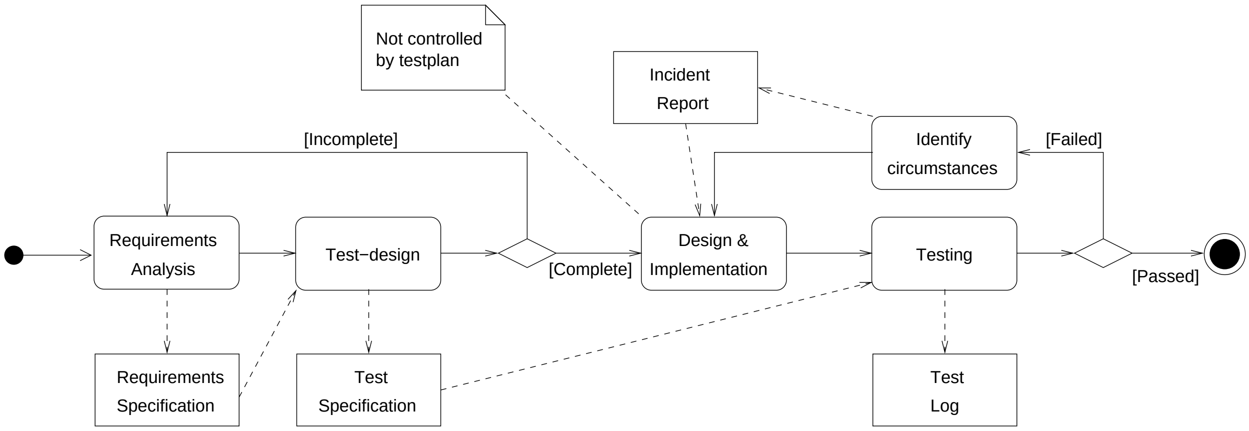


**Figure H.1:** A view of the development-process with focus on testing and test documentation

It can be seen from the figure that both the requirements analysis and test specification are to be finished and evaluated as complete prior to the design and implementation phases of the development process. Complete in this relation means that the product should be fully described by the requirements specification and that all requirements are covered by test-cases in the test specification. If either fails to evaluate as complete then the process is reiterated.

When the system or module has been designed and implemented then the corresponding test is carried out in accordance with the test specification. If the test fails i.e. one of the specified test-cases yields an incorrect result then an incident report is written and it serves as an input for the reiteration of the design and implementation phases. When the test is passed the module or system is through the development phase and it then enters the maintenance part of its life-cycle.

## H.5 Test Deliverables

The following subsections define what documentation must be worked out for each test item. In order to preserve space in the report not all of the documentation is enclosed as part of the report, therefore it is stated for each documentation item if it is to be included in the report.

### H.5.1 System Test

The following documentation items should be worked out in conjunction with the system test.

- A system requirements specification
- A test specification enclosed as an appendix
- If the test has not been fulfilled successfully before the project is due then incident reports describing the circumstances in which the test fail and the test log must be enclosed as an appendix
- Regardless of the success of the system test the main report should contain a summary of the system test results

### H.5.2 Module-tests

The following documentation items should be worked out for each of the module tests.

- A section that states all module requirements. This section is to be enclosed as the first section of the chapter that describes the particular module
- A module-test specification enclosed as an appendix
- Test logs and incident reports are to be worked out when executing the test, but they should not be enclosed in the report unless unresolved incidents exists when the project is due.

# Appendix I
# System Acceptance Test

## I.1 Test Item Identifier

This testspecification specifies the system acceptance test i.e. the test that documents that all requirements are met by the final PSU. The foundation for all testcases in this specification is the power system unit requirements specification as provided in chapter 3 on page 23.

## I.2 Test Environment

The following equipment is needed in order to conduct the system test:

- Oscilloscope
- Multimeters
- A variable resistor that can endure up to 15 W

Further the following documents are needed:

- System Requirements Specification
- Deployment and operation manual

## I.3 Test Approach

This test is a blackbox test that tests that all interfaces of the prototype behaves as specified. The testcases are grouped together in groups that reflect the same interfaces that are specified in the system requirements specification. These are followed by a group of test cases that do not apply to a single interface. In any test where a user is required, this or these users are simuated by a variable resistor so that the load is as expected.

## I.4 General Description

This section describes the tests that has to be made to control the overall input and output from the PSU.

### I.4.1 Test of MPPT algorithm

With a constant illumination on the solar panels, the duty cycle of the MPPTC is measured together with the MPPTC output power. The duty cycle is expected to either rise or fall continuesly for some time, and then stabelize by oscillating between three settings. The illumination of the solar panels is then changed and the duty cycle and power is again followed. The duty cycle is again expected to rise or fall continuesly for some time and then stabelize. If this is the case the test is deemed a succes.

### I.4.2 Test of PSU

The PSU has to deliver power to the other users on board Cubesat. Its capability to deliver this power will be verified by measuring the output voltage and current and see if the equivalent power is sufficient to the worst case, where the subsystems consume the most power. This power is according to table 3.1 on page 26 $2500\,mA \cdot 5\ V = 12.5W$.

To control that the batteries charge when the PSU produces more power than is consumed by its users some of the users are switched out so the input power is bigger than the user's power consumption. If the output power is the required power from the users, then the rest of the power is used to charge the batteries. To control this the output voltage and current are measured and the power calculated and compared to the power consumption from the users.

Likewise the input power to the PSU is lowered, so the power consumption by the users is bigger than the input power to the PSU. If the output power is bigger than the input power the batteries are discharging. This is controlled by measuring the voltage and current on the input and output, then calculate and compare the power on the input and output.

To test the capacity of the batteries, they are charged to their maximum 8.4 V, where after they are discharged with a current equivalent to the expected discharging current. This is keept going during the expected discharging time of the satellite, where after the batteries must still have a voltage over their minimumvoltage of 6V.

## I.5 Interface Functionality

In this section a test for the following three interfaces will be described:

- Power Distribution
- DHS Communication
- OBC Boot Selection

### I.5.1 Power Distribution

The tests that are to be carried out in the power distribution interface are:

- Output voltage on the power bus
- User protection
- Protection for the OBC

#### I.5.1.1 Output Voltage Control

To ensure that the voltage on the main power bus is according to the requirements 3.3.1 on page 26 an oscilloscope is connected to the bus as seen in figure I.1. The measured voltage on the bus must be 5 V but it is acceptable with deviations within 1% i.e. between 4.95 - 5.05 V. It must also be tested that the bus voltage deviations is not more than 2% within 20 ms when a user is shifted on or off. It shall be tested by connecting an oscilloscope to the bus and switching the load on and off.

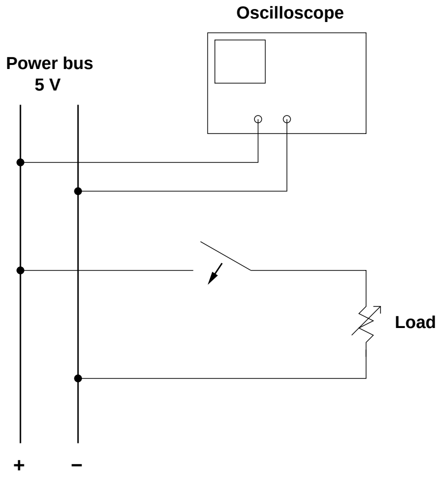


**Figure I.1:** Bus-test

#### I.5.1.2 User Protection

The protection for each user must be tested according to the shutdown current that has been defined in 3.2 on page 24. In table I.1 the shutdown current for the different users is shown.

When a user draws too much current, the PSU must cut the current to this user to prevent it from destruction. Figure I.2 shows a test circuit for a user connected to the power bus.

One after one each user is set to draw a higher and higher current and the current drawn is measured. When a current equal to the shutdown current of the specific user drawn, the PSU must cut off that user.

| User | Shutdown current [mA] |
|---|---|
| OBC | 220 |
| ACS | 88 |
| CAMERA | 55 |
| TRD | 1034 |

**Table I.1:** Table of user requirements

### I.5.1.3 User Protection of the OBC

The OBC has a special user protection that must be tested separate. The difference between the protection in OBC and the other users is that reconnection is carried out by the PSU instead of the DHS. The OBSs shutdown current can be seen in table I.1. The test will be carry out as the other protection with a variable load as mentioned above. After a disconnection of the OBC the PSU waits for 5 minutes before it reconnects the OBC to the power bus.

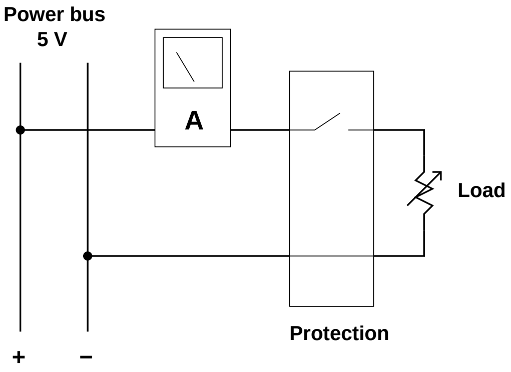


**Figure I.2:** User protection test

## I.5.2 DHS Communication

In this test the following interface will be tested: Communication of house-keeping information and communication of user on/off status.

### I.5.2.1 House-keeping Information

The PSU must be able to measure and send the following measurements to the I2C bus:

- Voltage across parallel connection of solar-panels (1 measurement)
- Currents through solar-panels on each side of the satellite (5 measurements)
- Voltage across parallel connection of batteries (1 measurement)
- Voltage on 5 V bus (1 measurement)
- Currents to each user (OBC, ACS, CAM, TRD & PSU) (5 measurements)
- Temperature measurements (6 measurements)

The voltages, currents and temperature measurements in the PSU are controlled by measuring the voltages and currents with multimeters and the temperature with a thermometer. It is then controlled, that the MCU reads and sends the correct measurements on the I2C bus.

### I.5.2.2 Users On/Off status & control

This controls the messages between PSU and OBC. These messages are specified in appendix G on page 201.

Like in the OCP test the users one after one are made to shut down. The DHS will, simulated with another computer, send a signal out on the I$^2$S-bus with orders to turn the specific circuit back on. The PSU should then turn the circuit back on. Next the PSU is commanded to shut down the users one after one. If the PSU recieves these signals, it should shut down the users and the returned signal to the DHS is verified with the logic analyzer. Similarly when the DHS asks for status on the users from the PSU. The signal is sent from the computer and the returned signals from the PSU is controlled with the logic analyzer.

### I.5.3 OBC Boot Selection

The PSU must be able to boot up the OBC in case of malfunction. The boot sequence must be carried out from the algorithm described in figure 3.2 in section 3 on page 23. The signals between the PSU and OBC is defined in appendix G on page 201.

#### I.5.3.1 External Watchdog Timer

When the OBC has been powered the watchdog timer is started. When the OBC has booted up the **Signal()** command is sent to the PSU through the $I^2C$-bus to tell the PSU that the startup was successfull. Another computer, which will represent the OBC in the test, sends this signal to the PSU. If it is sent before the time has expired the PSU should work normally, and this is controlled by measuring and controlling the output voltage. If the signal is not sent before the time has expired, the OBC should be shut down, and another attempt to boot up the OBC should be done after 5 minutes.

#### I.5.3.2 Selection of Boot Mode

The signals to start up from the PROM or EEPROM is sent to the PSU. If the OBC starts up successfully it is assumed that the boot was from the specified memory, since the boot programs are identical in both memories.

## I.6 Quality of Service

This section describes the test of the requirements defined in section 3.4 on page 28.

### I.6.1 Efficiency

The efficiency of the PSU must according to the requirements be at least 75%. To determine the efficiency the voltage and current on the input and output are measured. Then the power on each side is calculated and compared.

### I.6.2 Accuracy on Housekeeping Data

The accuracy on these data are not critical and therefore no exact requirements are made on them.

### I.6.3 Operation without Batteries

To test the case where the PSU is working with defect or fully discharged batteries, the batteries are switched out of the PSU. Then it is controlled if the PSU can start up from the lower voltage levels of the solar array, which will be simulated with a programmable power supply. This is done by measuring the output voltage on the power bus and control it. If this voltage is 5 V, the start-up from the solar array was successfull. And if this voltage is achieved, it is assumed that the batteries are charging.

### I.6.4 Single Event Upset and Latch-Up

Single Event Upsets and Latch-up can cause a pin on the MCU to change state and thereby mess up the logic. This could cause a system failure, and the watchdog timer has to detect this failure and restart the system. In the test a pin is forced high or low and it is verified, that the PSU restarts.

### I.6.5 Required Lifetime

The Cubesat is scheduled to have a lifetime for at least a year. So this is also the case for the PSU onboard Cubesat. However there are no possibility to test if this lifetime can be achieved, so this requirement will not be tested.

## I.7 Environmental Requirements

The product in this project is only a prototype of the final PSU, that is to be implemented on Cubesat. Therefore the environmental requirements specified in 3 on page 23 will not be met by this prototype and therefore not tested. If our design is chosen to be the one used on Cubesat, these requirements will have to be tested, when the PSU is made.

# Appendix J
# Module Test Specifications

## J.1 Module Test - MPPT Converter

In this appendix the procedure of module test for MPPT converter will be described. The aim of this test is to ensure that the requirement that was made for the converter have been fulfilled. The test will range the prototype of the converter.

### J.1.1 Test Specification Identifier

The test is based on the requirement for the MPPT converter (refer to subsection 6.1.1 on page 41). In this module test the converter will be tested without other module connected to the converter.

### J.1.2 Test Environment

The equipment needed to carry out the module test is list below.

- Power supply, to supply the input to the converter, a power supply will be needed. The power supply will be used instead of solar arrays and therefore it must be able to regulate the voltage.
- Frequency generator, will be needed to make the frequency that is proportional to the frequency that comes from the digital hardware.
- Variable resistors, will be used as load to the converters output
- Oscilloscope
- Ampere meters
- Volt meters

### J.1.3 Test Approach

The test-specification will provide one or more test-cases for each requirement stated in the module requirement specification.

### J.1.4 Test Case Specification

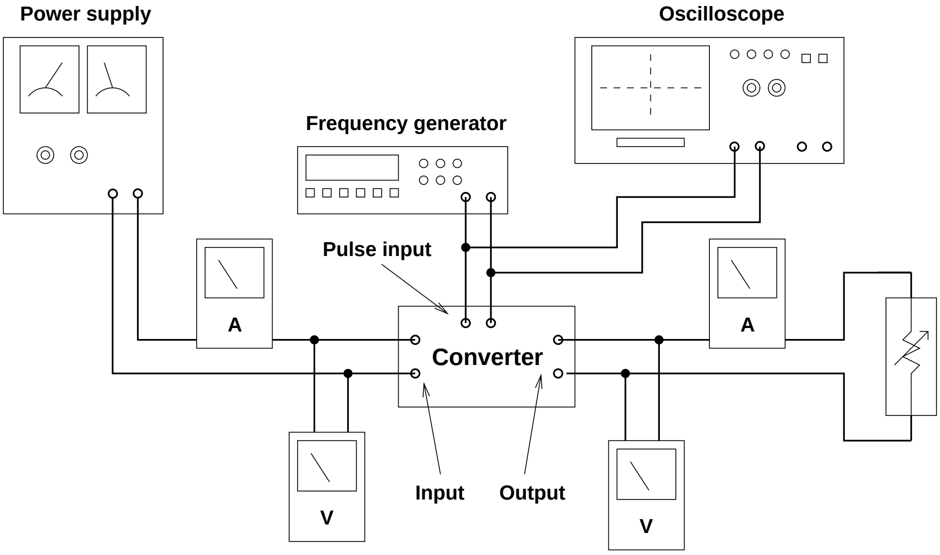


**Figure J.1:** Line-up diagram of the test

In this module test the following items will be tested:

1. The output voltage from the converter

2. The output ripple

3. The converter's efficiency

### J.1.4.1 Output voltage

The equipment placement must be as shown in figure J.1. The output from the power supply must be connected to the input. The oscilloscope must be connected to the converters output and the frequency generator must be connected to the MOSFET (GATE-pin). The ampere meters and volt meters must be connected on the converter's input and output. This test case will be carry out in the low, average and high duty cycles.

To test the high duty cycle the voltage supply must be set to 3.4 V and the volt meter must be adjust so it is able to measure the converter's output voltage (8.4 V). The frequency generator must be set to 50 kHz with a duty cycle of 59.5 %. The variable resistor is set to $23.5\ \Omega$. The output voltage is expected to be 8.4 V.

To test the average duty cycle the voltage supply must be set to 3.8 V. The frequency generator must be set to 50 kHz with a duty cycle of 45 % and the resistor is set to 18 $\Omega$. The output is expected to be 7.2 V.

To test the low duty cycle the voltage supply must be set to 4.2 V. The frequency generator must be set to 50 kHz with a duty cycle of 30 % and the resistor is set to 12 $\Omega$. The output is expected to be 6 V.

### J.1.4.2 Output ripple

The equipment placement must be the same as in the output voltage test case as shown in figure J.2.

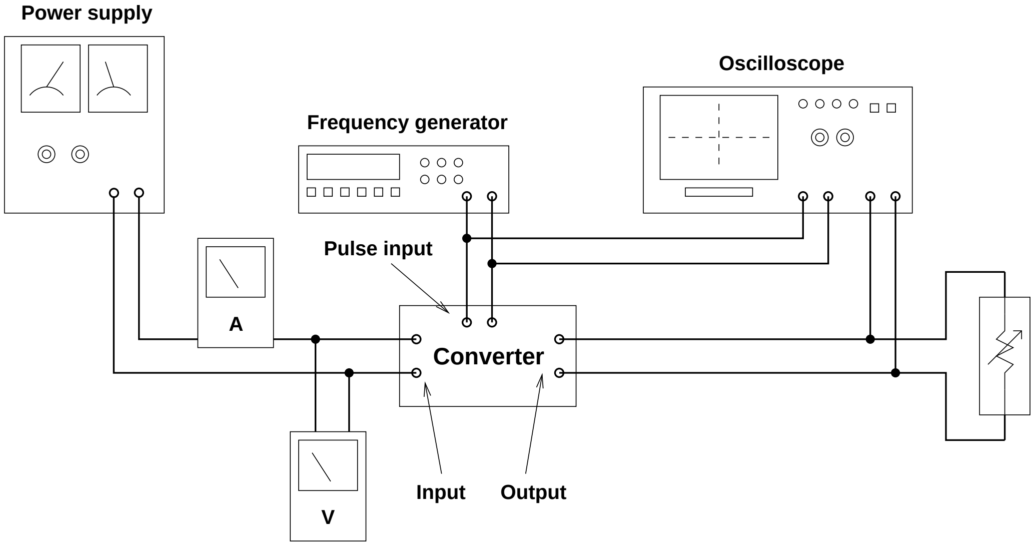


**Figure J.2:** Placement diagram for ripple test case

The output ripple must be tested when the converter is boosting from the lowers t voltage to the highest, with maximum input power. The voltage supply must be set to 3.6 V and the oscilloscope must be adjust so it is able to measure the converter's ripple voltage (approximately 100 - 500 mV). The frequency generator must be set to 50 kHz and a duty cycle of 59.5 %. The ripple is expected to be maximum 2 %.

### J.1.4.3 Efficiency

In this test case the equipment placement must be as shown in figure J.1. As in the voltage test the efficiency must be tested in low, average and high duty cycle.

For the test in high duty cycle the voltage supply must be set to 3.4 V. The frequency generator must be set to 50 kHz, the duty to cycle of 59.5 % and the variable resistor is set to $23.5\ \Omega$. The input and output voltage and current is measured with the volt meters and ampere meters. It is possible that the duty cycle and the resistor must be adjust to get the exact output voltage. The measurement shall been use to calculate the converters efficiency. The efficiency in pro-cent is found by putting the result from the measurement in equation J.1.

$$\eta = \frac{U_{input} \cdot I_{input}}{U_{output} \cdot I_{output}} \cdot 100\ [\%] \tag{J.1}$$

The test for average and low duty cycle must be carried out in the same way as in high duty cycle. The only exceptions are the input voltage must be set to 3.8 V, the duty cycle to 45 % and the resistor to 18 $\Omega$ for the average duty cycle. In the low duty cycle the voltage must be set to 4.2 V, the duty cycle to 59.5 % and the resistor to 12 $\Omega$

## J.2 Module-test Specification of the PCC-converter

This is the test specification for the Power Conditioning converter described in chapter 7 on page 55 and it is specified to test that all requirements stated in section 7.1.1 on page 56 is fulfilled by the module.

### J.2.1 Test Environment

In order to carry out the test the following equipment are needed:

- Power supply 0-20V
- Oscilloscope that can save data to disc
- 2 ampere-meters
- Function Generator
- Variable resistor

### J.2.2 Test Approach

The test-specification will provide one or more test-cases for each requirement stated in the module requirement specification.

### J.2.3 Test Case Specification

The following test-cases makes up the test-procedure.

#### J.2.3.1 Output voltage

This test-case is designed to test the requirement that the output-voltage of the PCC should be 5V. The test configuration is given in figure J.3.

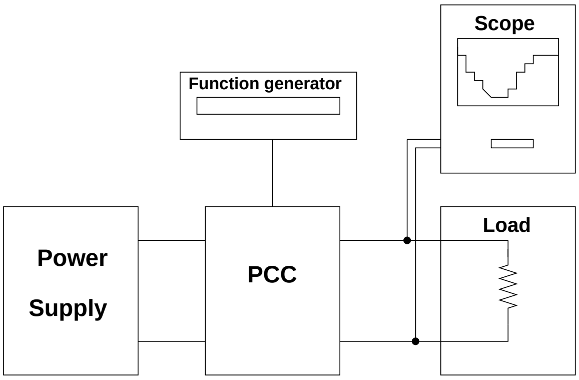


**Figure J.3:** Test-case configuration for test of output voltage

A voltage of 7.2V (battery average) is supplied by the power supply and the function generator supplies a square wave of 21kHz with a voltage 3.6V$_{p-p}$ and duty-cycle 69.4%. The variable resistor is set to 2Ω in order to simulate a worst case load of 2.5A at 5V. The oscilloscope measures output voltage.

Since the converter is unregulated the output voltage experienced at this duty-cycle will be less than 5V due to losses. Therefore the duty-cycle off the frequency-generator must be adjusted manually until the output becomes 5V. Using the oscilloscope the duty-cycle where output is 5V is written down for evaluation purposes.

The input voltage is changed to the worst case, which is 6V and, the test is conducted again. If it is possible to get 5V output voltage in both cases then the PCC fulfills the requirement of 5V output voltage.

#### J.2.3.2 Ripple in Steady-State

This test-case will test that the requirement on output ripple is met for steady-state operation. The test configuration is given in figure J.4.

A voltage of 7.2V is supplied by the power supply and the function generator supplies a square wave of 3.6V$_{p-p}$ with a frequency of 21kHz. The variable resistor is first set to 2Ω in order to simulate a load of 2.5A at 5V, which is the expected worst case load.

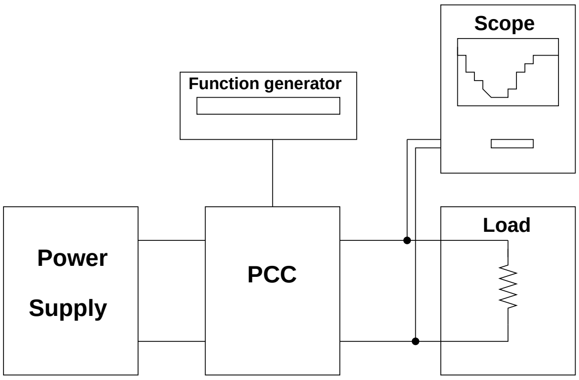


**Figure J.4:** Test-case configuration for test of ripple in steady-state

The oscilloscope measures the output voltage and when the converter enters steady-state a snap-shot of the output waveform is saved to disk. The snap-shot is studied and the ripple of the waveform is found. The test is then re-iterated with an load current of 0.25A. The ripple in both cases must be less than 2%$_{p-p}$ compared to the dc-value.

#### J.2.3.3 Efficiency

This test-case will test that the requirement on converter effeciency is fulfilled. The test configuration is given in figure J.5.

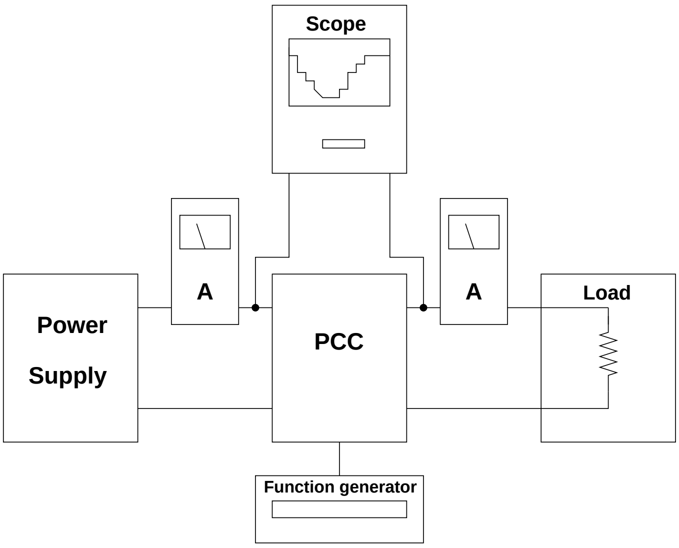


**Figure J.5:** Test-case configuration for test of effeciency

A voltage of 7.2V (battery average) is supplied by the power supply and the function generator supplies a square wave of 3.6V$_{p-p}$ with a frequency of 21kHz. The duty cycle is during the complete test adjusted so that the output is always 5V. The output- and input-current are measured with ampere-meters and the input- and output-voltages are measured by the oscilloscope.

The variable resistor is first set to 20.0Ω in order to simulate a load of 250mA at 5V, which is the expected load under idle operation of the satelitte. When the PCC enters steady-state operation the values of currents and voltages is noted. The loads is then changed to 2Ω to simulate the worst case load and values are noted again. For both loads the test is performed again with input voltages of 8.4V and 6.0V in order to evaluate for for best and worst case batteryvoltage.

The effeciency in percent is found by:

$$\eta = \frac{U_{input} \cdot U_{input}}{U_{output} \cdot U_{output}} \cdot 100 \tag{J.2}$$

The efficiency must be 90% or better in order to fulfill the requirements.

## J.3 Module-test Specification of the OCPC

This is the test specification for the OCPC described in chapter 8 on page 71. The specific module is the one used for the OBC, ATC, Camera and COM.

### J.3.1 Test Environment

In order to carry out the test the following equipment is needed:

- Power supply 0-20V [2 A]
- Oscilloscope that can save data to disc
- Function Generator [square, 5 $V_{pp}$]
- Variable resistor [200 - 0 Ω ,5 W]
- Mechanical switch

The use of the items are described as they are used.

### J.3.2 Test Approach

The test-specification will provide one or more test-cases for each requirement stated in the module requirement specification.

### J.3.3 Test Case Specification

The following test-cases make up the test-procedure.

#### J.3.3.1 Power up close down

When the OCPC is feeded power the loads should remain unpowered until the MCU signals so. This is tested with the circuit in figure J.6. The voltage on the load is monitored and the supply power to the OCPC is switched on. The voltage supplied to the load schould remain zero volts.

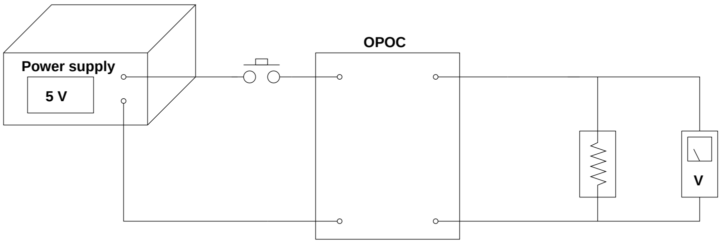


**Figure J.6:** Testing the power up of the system. The voltage supplied to the user should remain 0 V

#### J.3.3.2 MCU control

The MCU has to be able to switch the current flow on and off. Furthermore the MCU has to be able to switch a closed subsystem back on. This is controlled with the circuit in figure J.7. The power is connected and the user schould remain off. The function generator is switched on and is set to supply a square 3.6 $V_{pp}$ signal. The square signal should on positive edges turn on the loads and turn them off at negative edges. This should be visible on the scope as a 5 $V_{pp}$ square with the same frequency as the frequency generator.

#### J.3.3.3 Overcurrent switching

This test is performed to test two things:

- Overcurrent measuring
- Overcurrent switching

Overcurrent measuring is testing wether the circuit is capable of detecting that the current is too high. Overcurrent switching is testing wether the circuit is capable of switching if the current is too high.

A voltage of 5 V is supplied to the OPOC and the voltage supplied to the load is constantly measured with the oscilloscope. The load resistor is lowered and the output from the OCPC is monitored. At a certain point the OCPC will react to the current flow and the current is closed. If the OCPC is working the current flow will stay closed.

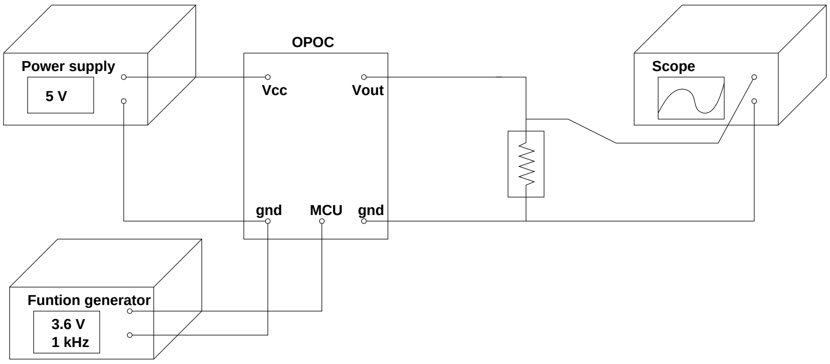


**Figure J.7:** Testing the MCU control of the OCPCs. The MCU has to be able to switch on and off the users

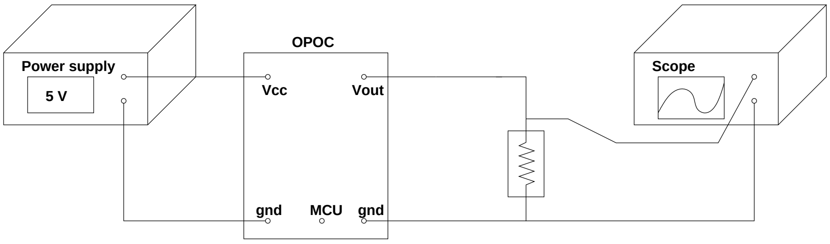


**Figure J.8:** Testing the overcurrent senseing and switching

## J.4 Module-test Specification of the Digtital Hardware

This is the test specification for the digital hardware developed in this project and described in chapter 10 on page 107 and it is specified to test that all requirements stated in section 10.1.1 on page 107 is fulfilled by the module.

### J.4.1 Test Environment

In order to carry out the test the following equipment is needed:

- Power supply 0-5V
- Oscilloscope that can save data to disc
- 3 ampere-meters

### J.4.2 Test Approach

The test-specification will provide testcases for each part of the digital hardware. Since no full implementation of the digital hardware has been performed (as described in section 10.7 on page 122) only those parts which have been implemented are described.

### J.4.3 Test Case Specification

The following test-cases makes up the test-procedure.

#### J.4.3.1 Test of Power Supply Select Circuitry

To test that the digital hardware is able to be powered from both solarcells, batteries and the main power-bus in a prioritized way, as described in 10.2.1 on page 108, the following test is performed, with the testconfiguration given in figure J.9.

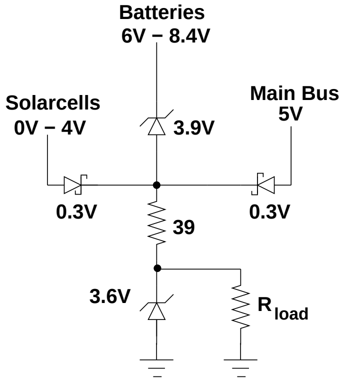


**Figure J.9:** Testconfiguration for test of startup circuitry

The $R_{load}$ load resistor is set to 79Ω to simulate the 50mA current consumption of the digital hardware at 3.6V. One power supply is connected to each of the three possible supply inputs to the digital hardware. Initially all power supplies have 0V of output voltage. The following procedure is performed:

1. The voltage of the solarcell power supply is adjusted to 4V and the voltage over the load is measured. It should be in the range of 3.3-3.7V, meaning that the DHW is adequately powered from solarcells alone.

2. The voltage of the battery power supply is set to 6V and using the amperemeter on the two active power supplies it is checked that the current is now drawn from the battery supply rather than the solarcell supply. Further the load voltage is measured and should be above 3.3V-5V.

3. The voltage of the battery power supply is adjusted to 8.4V and the same measurements from last point is performed.

4. Now the main powerbus voltage supply is set to 5V and it is again controlled that the current now is drawn from the main powerbus supply rather than the others. Again the load voltage is measured and should be between 3.3V-5V.

If the above points can be performed with no incidents it is concluded that the circuitry works.

#### J.4.3.2 Test of Power on Reset Circuitry

The power supply select circuit is bypassed and the MCU and reset circuitry is supplied directly from a power-supply. The followin test procedure is performed:

1. The voltage of the power supply is set to 2.4V. It is measure, using the oscilloscope, that the output of the reset circuitry is asserted meaning that the POR resets the MCU due to a undervoltage condition as required (See subsection 10.2.2 on page 109)

2. Using the oscilloscope it is tested that the POR circuit asserts the RESET pin of the MCU for at least XXX350msXXX when a supply voltage of 3.3V is turned on.

If these test is executed succesfully the circuit meets its requirements.

#### J.4.3.3 Test of Core MCU Implementation

This testcase tests that the MCU itself, as part of the DHW, is able to execute code and therefore connected properly. The code to execute is the following simple test-code:

```
void main()
{ a=0
   while (1) P0=a++;
}
```

The test code simply counts on port o of the MCU. The code is compiled and downloaded to the flash-ROM of the MCU and executed. If it is possible to see the output of the program on port 0 of the MCU then the testcase is passed.

#### J.4.3.4 Test of Pulse-Width-Modulators

This testcase test that all the circuits that make up the interface between the MCU and the gate drivers of the converters work, i.e. hold circuits, pulse-width-modulators and oscillators. This test is only performed on oneof the two PWM-cicuits, since they are similar.

In order to perform this test two pieces of test codes for the MCU is written; One which slowly variates the voltage of the DAC, which interfaces to the circuits, up and down, and one piece of code that sets the DAC to keep a specific voltage at the output. The following test procedure is performed.

1. The first software driver is downloaded to the MCU and executed. It is measured using the oscilloscope that the duty-cycle of the ouptut to the gate driver varies between 0% and 100%.

2. The second driver is used and it is controlled that the duty-cycle observed on the output corresponds to the input voltage.

3. Still using the second driver the MCU is disconnected from the supply voltage and it is measured, using the oscilloscope, how long time elapses before a 10% deviation in duty-cycle can be observed on the output. This time should be above XXX400msXXXX to allow the MCU to perform a reset and initialize before taking over the control of the PWM-modulators again.

#### J.4.3.5 Test of Measurement points

The measurement points will be tested with a power supply and an oscilloscope. The power supply is connnected to the measurement points. The voltage and current from the power supply is measured with multimeters and it is controlled that the measured value on the MCU is the same. The temperature is measured with a thermometer and it is controlled that the value on the MCU is the same.

All the measured values from the MCU must be read from the I$^2$C-bus with another computer.

## J.5 Module-test Specification of the Software and Controllers

In this test specification the software test is described which also includes the test of the controllers for the converters. This module-test deviates a little from the other module-tests since this test cannot be done without the interaction of the other hardware modules. Therefore this module-test is done after all hardware modules are integrated and integration-tests are completed.

### J.5.1 Test Environment

To carry out the test a number of different equipment are needed:

- All hardware modules of the PSU
- I$^2$C compatible PC
- Oscilloscope
- Programmable power supply to simulate solar cells
- Variable resistor to simulate load
- Four batteries (type DLP 443573 from Danionics)
- Solar Cell or Programmable Power supply

### J.5.2 Test Approach

The approach taken for this test is to first test all software specific test-cases, such as I$^2$C and OBC-boot port. Thereafter the controller software, which includes the design of the controllers, are tested.

For the first set of test-cases a software simulator is used since the OBC hardware is not available for this test and no other hardware capable of I$^2$C communication has been available. The second set of test-cases is performed using the complete implemented and integrated hardware.

### J.5.3 Software Specific Test-Cases

To test the I$^2$C bus interface as either a MCU or a PC to function as a I$^2$C master is needed. For the Cubesat project a group was given the task to develope the necessary test equipment, but at the deadline of this projekt the equipment the tools was not ready for use. Therefore the final test of I$^2$C is postponed until the tools are ready.

The I$^2$C will however still be tested using the debugger for the ADuC812 where a I$^2$C connection are simulated. The interface and all communication are then whitebox tested going through the complete protocol step by step. Thereby should eventual bugs be eliminated and software should be ready to implementation.

Since the 3C task are dependent on the I$^2$C ISR the functionality of this task will also be whitebox tested with the debugger. One functionality that can be tested is the external watchdog. After 10 seconds without receiving data from the I$^2$C the OBC reset leg must go low.

### J.5.4 Test-Cases for Software and Controller Design

Since it is already known that the software cannot fulfill the requirements is is supposed to (see subsection 14.1.2 on page 156) then this test will not be an ordinary module-test. It will instead be used as an evaluation test that tests what performance that can be achieved using the chosen processor.

#### J.5.4.1 Frequency of Controllers

As described (in subsection 14.1.2 on page 156) the MCU does not execute the tasks at the speed it is supposed to, but it executes at schedule derived for a MCU running at 89 MHz. Therefore in order to evaluate performance at 16 MHz this test-case is specified as follows.

The software is changed such that each time the 3C task is run then a port pin is toggled. The complete software is compiled and transfered to the MCU and the complete PSU is brought to operate. The frequency at which the pin toggles is measured.

Because it is known that the 3C task is supposed to be scheduled at 1 kHz and because all tasks are executed from a fixed schedule then using the measured value it is possible to derive the frequencies at which each task is scheduled.

#### J.5.4.2 Performance with all Tasks Running

Using the same software as from the previous test the following is performed in order to evaluate the behavior of the complete PSU with the processor running at 16MHz. Initially a resistor of 20 Ω is connected to the main power-bus to simulate a load of about 300 mA.

**Steady State** Voltages of solarcells, intermediate and main power-bus are measured and it is observed if the PSU is stable enough to bring itself to steady state operation, which will be defined as operation with stable voltages.

**MPP Tracking** The MPPTC current, solarcell- and battery voltage are measured. The input from the solarcells/simulator is changed and it is observed if and how the PSU tracks the new powerpoint.

**Battery Protection** The load current on the main power-bus is lowered to ensure that the batteries are being charged. The battery voltage is observed and it is checked that the voltage stops to rise when it reaches 8.4V

**Dynamic Load Change** The load on the main power-bus is switched from about 300 mA to 2.5 A and the voltage and current waveforms on the power-bus is saved by oscilloscope. The result is evaluated against the requirements given in table 9.6 on page 98

**Discontinuous Conduction Mode** The load resistor is removed, such that only the PSU draws power from the main power-bus. It is measured if the converter enters DCM at this current and if it does it is observed how it performs.

#### J.5.4.3 MPP Tracking

In order to evaluate performance under better circumstances (speed) the software is changed so that only the MPP-tracking tasks as well as the ADC-sampler are running. The MPP-tracking and battery-protection tests given above are then performed again and it is hereby possible to evaluate the performance of the algorithms when they are scheduled at their correct frequencies.

#### J.5.4.4 PCC controller

The software is altered so that only the inner- and outer-loop tasks which control the main power-bus as well as the ADC-sampler are scheduled. Further the port-pin-toggle code is inserted in the outer-loop task. The Steady-state test, from above, is performed and the frequency of the outer-loop task is measured. Then the dynamic-load-change test is performed and finally the DCM test.

These test with only the PCC control tasks running allow evaluation of the PCC controller design under better, but not satisfactory, conditions.

# Appendix K
# Simulink Models

## K.1 Models of MPPTC and PCC

At figure K.1 and K.2 the simulink models of the MPPTC and PCC can be seen. The two models are build using the Power Toolbox available in Simulink.

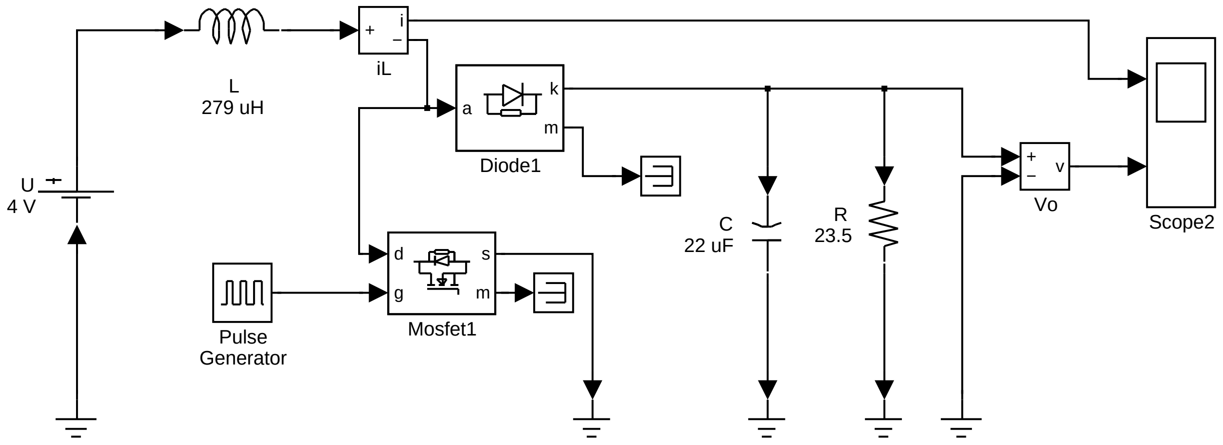


**Figure K.1:** The simulink model of the MPPTC

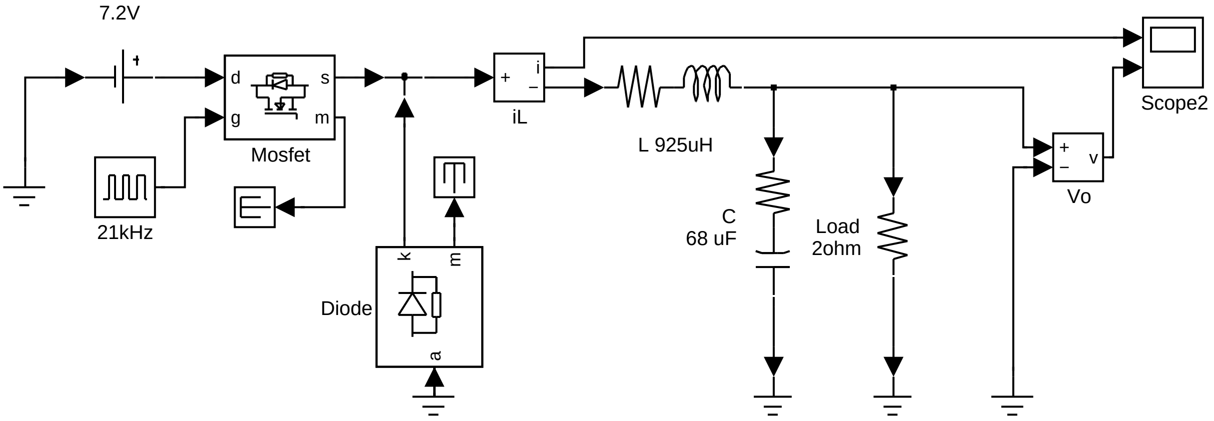


**Figure K.2:** The simulink model of the PCC

## K.2 Models of Controlled MPPTC

At figure K.3 the model of the MPPT controller are shown including prefilter and at figure K.4 the MPPTC with controller and solarcell model are shown.

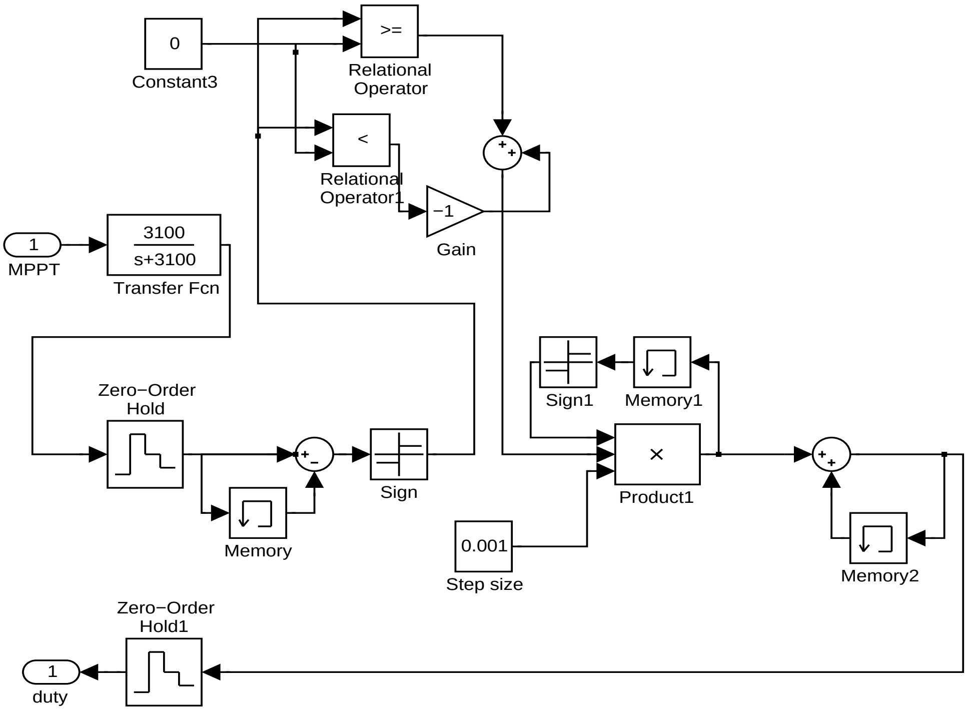


**Figure K.3:** The simulink model of the MPPT-controller for the MPPTC

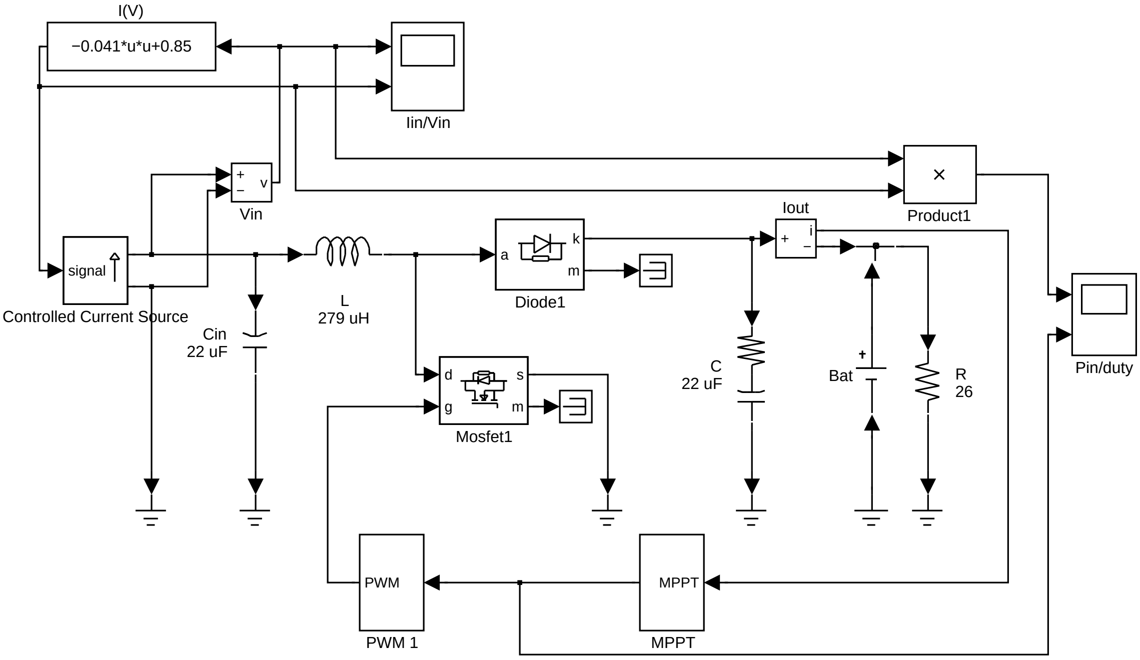


**Figure K.4:** The simulink model of the MPPTC with controller and solar cells model to the left

## K.3 Models of Controlled PCC

At figure K.5 the controller for the PCC are shown and at figure K.6 the PCC with controller, filters, sample-hold circuit and load switch are shown.

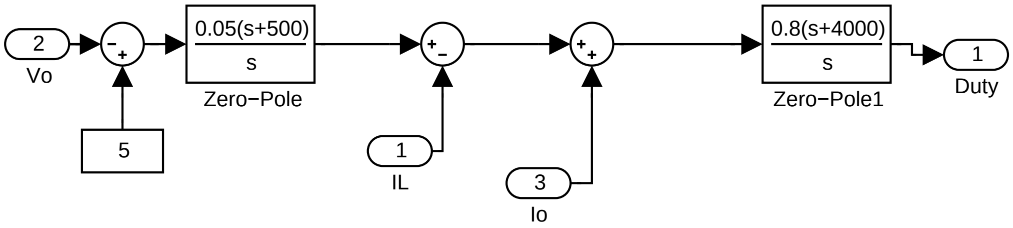


**Figure K.5:** The simulink model of the controller for the PCC

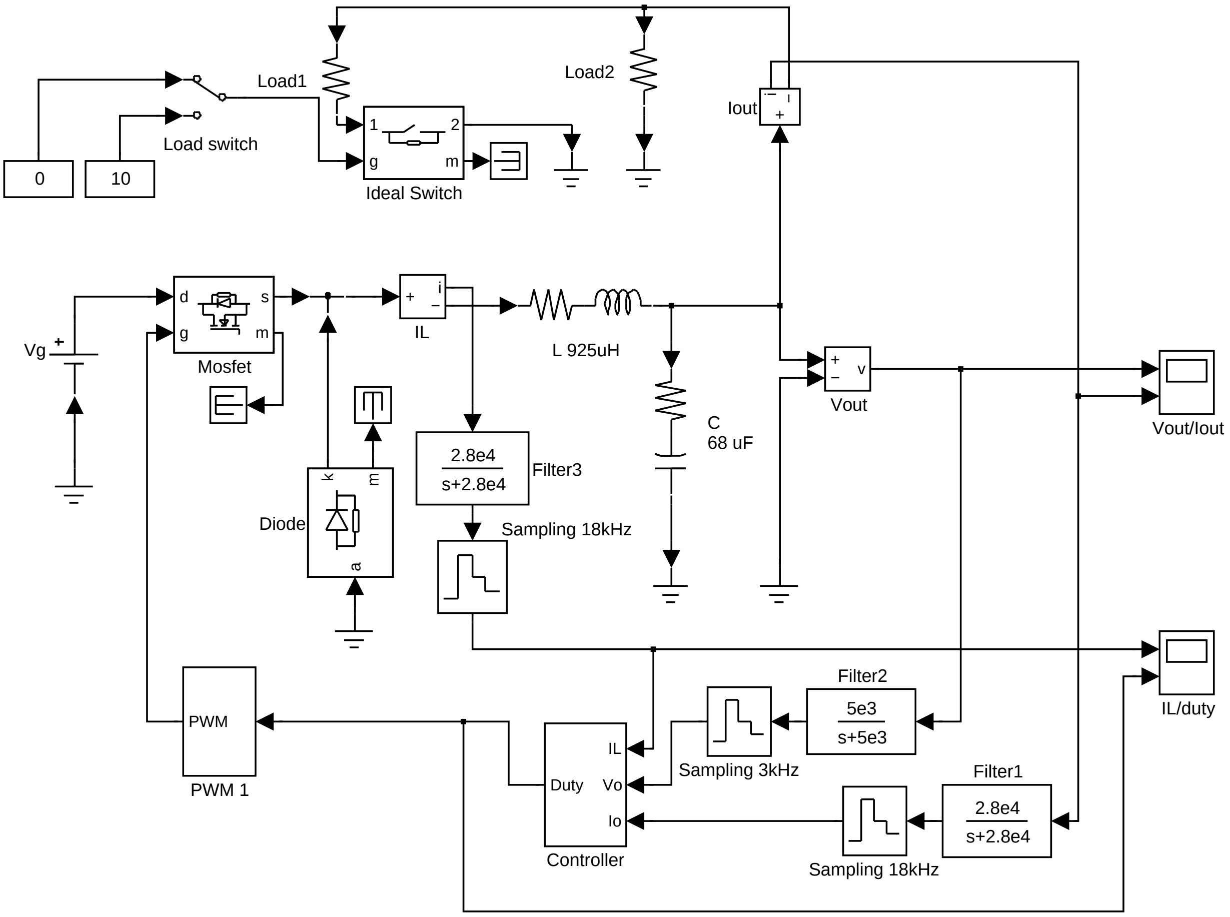


**Figure K.6:** The simulink model of the PCC with controller, PWM modul and load switch in the top left corner

# Appendix L
# Schematics

MCU Schematic

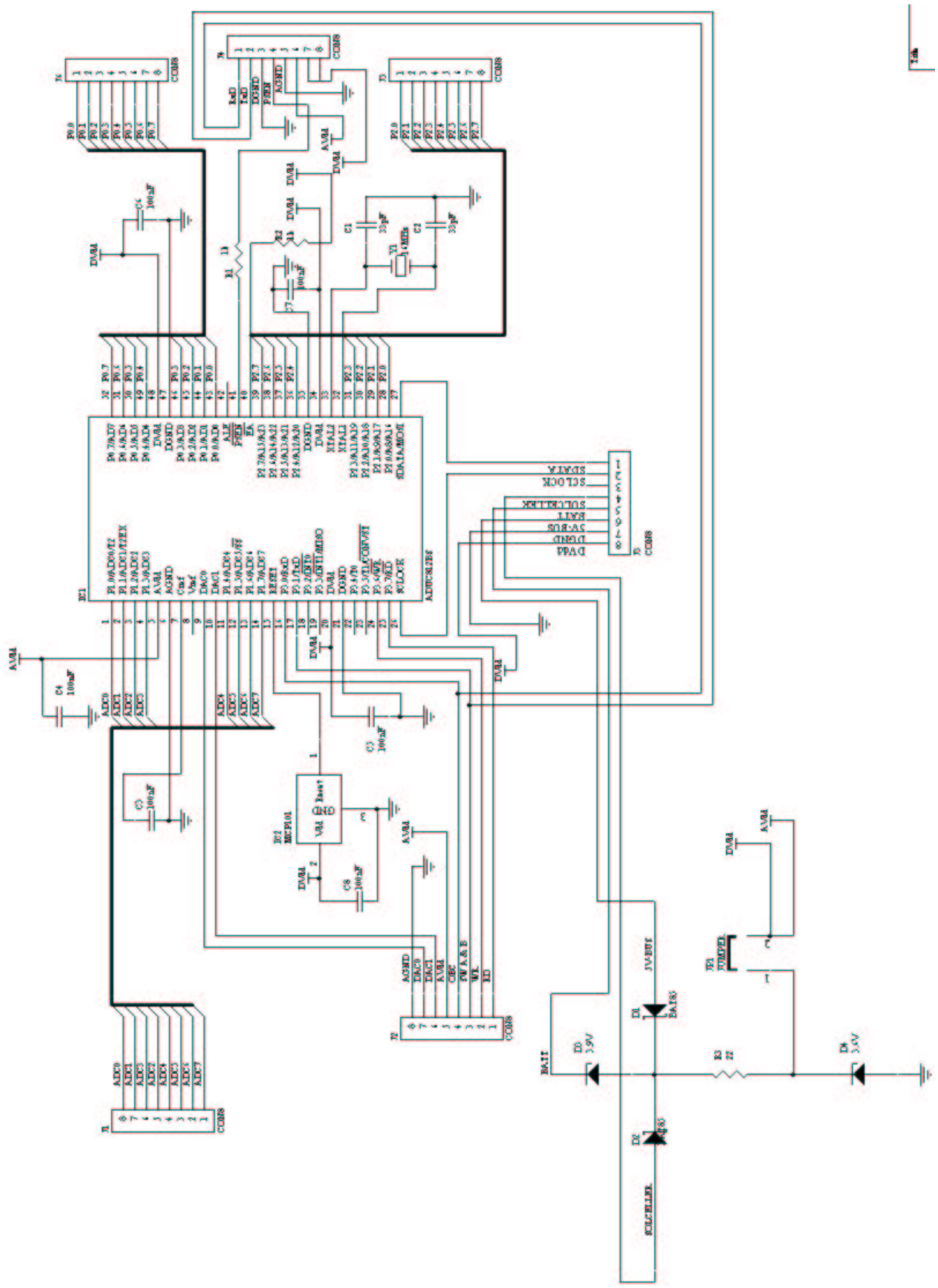


**Figure L.1:** MCU Schematic

# RS232 Schematic

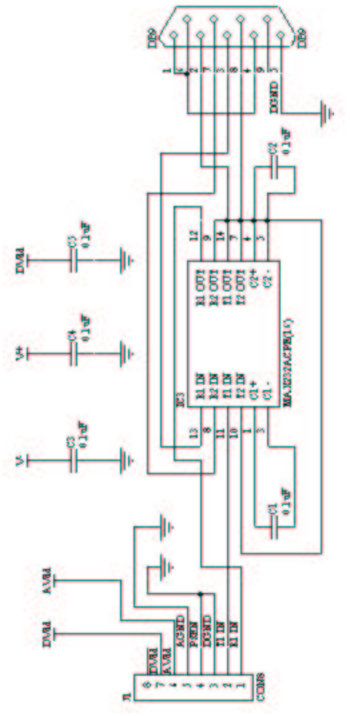


**Figure L.2:** RS232 Schematic

# PWM Schematic

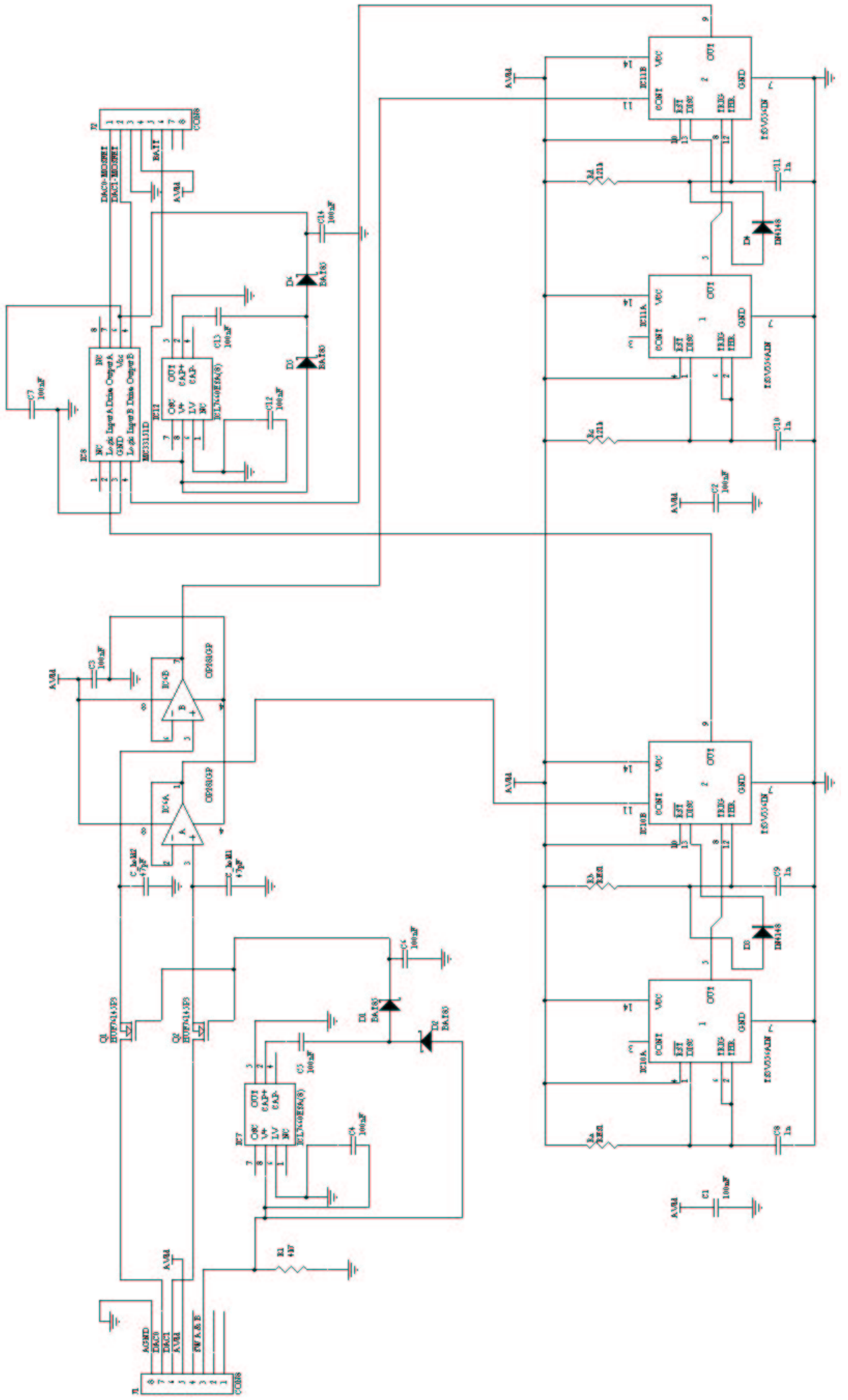


**Figure L.3:** DAC/Timer Schematic

## Converter and Measurement Schematic

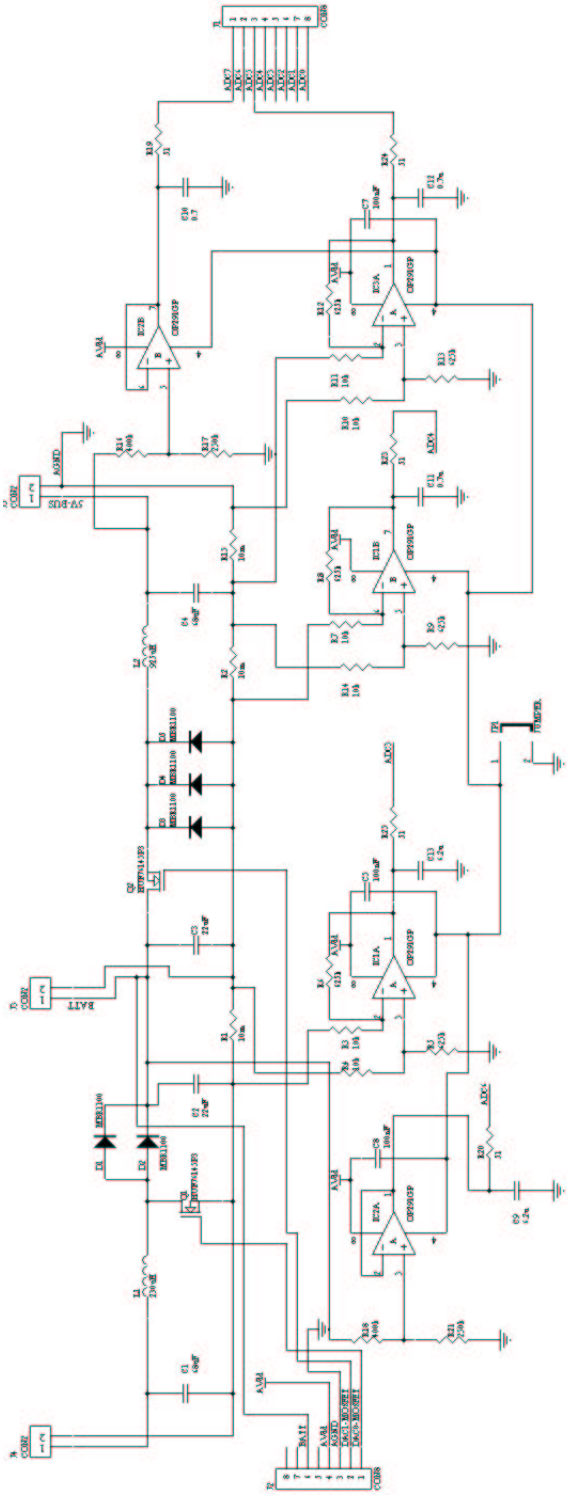
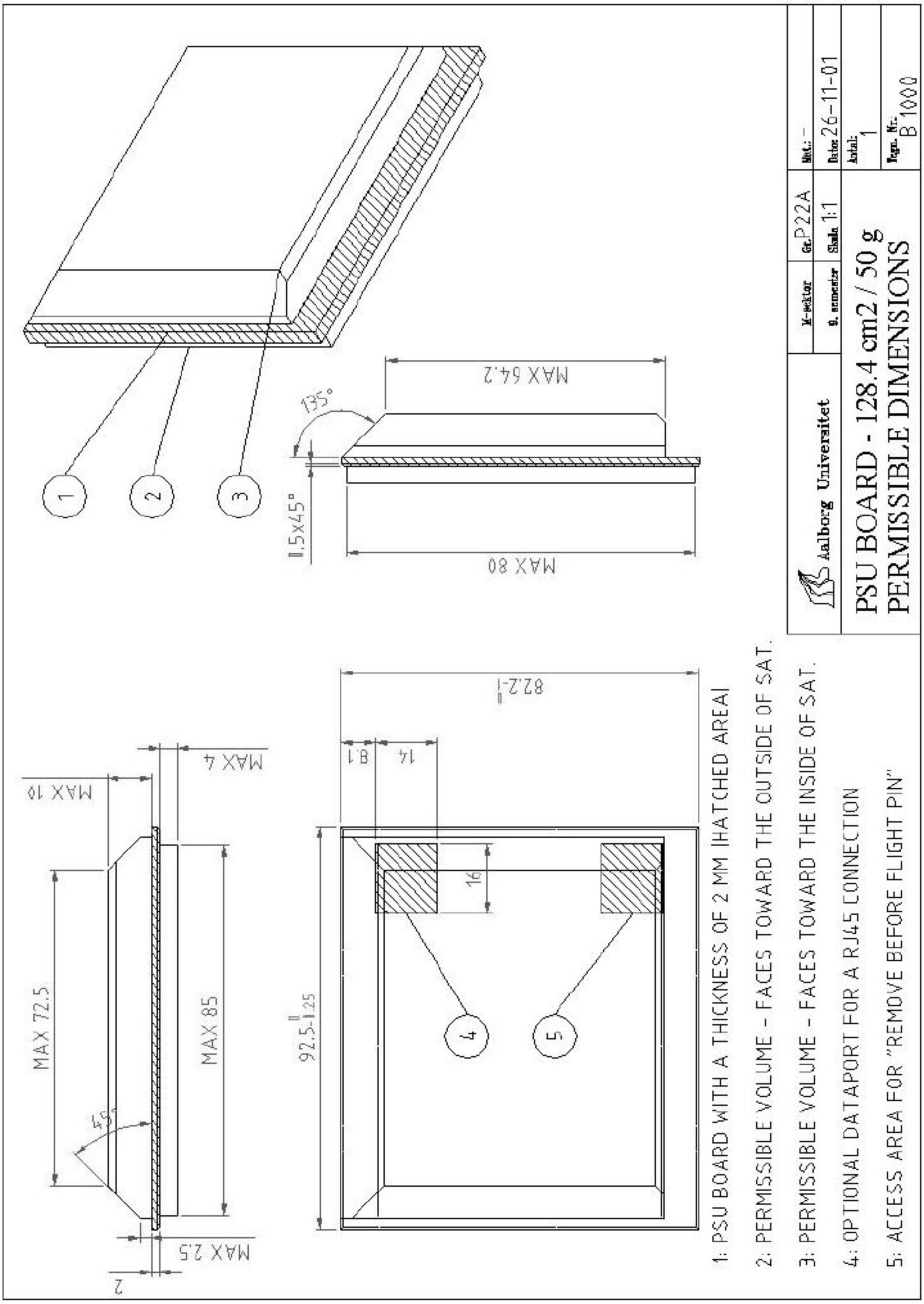


**Figure L.4:** Converter/Measurement Schematic

# Appendix M
# Technical Drawing of the PSU PCB

The following technical drawing of the dimensions of the PSU PCB has been obtained from the group responsible for the mechanival design of the cubesat.


1
2
3
4
5
MAX 64.2
135°
1.5x45°
MAX 80
82.2-1
8.1
14
16
92.5-1.25
MAX 4
MAX 10
MAX 72.5
MAX 85
45°
MAX 2.5
2
1: PSU BOARD WITH A THICKNESS OF 2 MM (HATCHED AREA)
2: PERMISSIBLE VOLUME - FACES TOWARD THE OUTSIDE OF SAT.
3: PERMISSIBLE VOLUME - FACES TOWARD THE INSIDE OF SAT.
4: OPTIONAL DATAPORT FOR A RJ45 CONNECTION
5: ACCESS AREA FOR "REMOVE BEFORE FLIGHT PIN"
Aalborg Universitet
PSU BOARD - 128.4 cm2 / 50 g
PERMISSIBLE DIMENSIONS
M-sektor
Gr. P22A
9. semester
Skala 1:1
Mat.: -
Dato: 26-11-01
Antal: 1
Tegn. Nr.: B1000

# Appendix N
# Power Subsystem Interface Proposal

## N.1 Introduction

### N.1.1 Document Revisions

The following lines state the document history.

| Revision: | Date: | Comment: |
|---|---|---|
| 1.0 | 2/10-01 | First draft |
| 1.1 | 21/10-01 | Major update due to newsgroup discussions and meetings between the three PSU groups |
| 1.2 | 25/10-01 | Update due to feedback on version 1.1 |
| 1.3 | 30/10-01 | Updates due to internal meetings, a meeting with ACS and mail correspondence with comm. system. |
| 1.4 | 18/11-01 | Changes due to information from PDR |

### N.1.2 Purpose

This document serves as an interface specification for the power subsystem of the AAU-Cubesat. It is intended to be used both for internal coordination of interfaces between the three groups that develop a power supply unit and as a reference document for subsystems that interfaces to the power supply unit.

### N.1.3 Maintenance

This document is maintained by group 01gr509 [1] on behalf of the three power subsystem groups and all questions and/or comments should be directed to this group. Any revision of this document is to be approved by all three power supply groups before submission to the public AAU-Cubesat documents.

### N.1.4 Definitions

The following definitions are used throughout this document:

**PSU** abbreviation for Power Supply Unit which is the unit that is developed by the three power supply groups

**OBC** refers to the On Board Computer of the satellite

**DHS** abbreviation for Data Handling Software

**ACS** abbreviation for Attitude Control System

**TRD** abbreviation for Transmitter/Radio Device

### N.1.5 PSU main functionality

In context of interfacing the main functionality of the PSU are to:

- provide loads with current from a voltage regulated power-bus
- provide load-protection if a load-current exceeds a predefined limit
- communicate with DHS in order to:
  - support DHS decision making by providing information about the power status and thermal data
  - inform DHS if a load has been shut down by the protection mechanism
  - allow DHS to make the PSU turn loads on and off
  - provide DHS with house-keeping information that are to be sent to the ground station

[1] Mail: 01gr509@control.auc.dk

- acts as boot-selector and external watchdog timer for the on-board computer

The following loads are to be connected to the PSU:

**On Board Computer (OBC)** is responsible for running the data handling software which monitors and controls the activities of the satellite

**Attitude Control (ACS)** is responsible for controlling the orientation of the satellite such that camera and antennas are pointed in the right direction

**Camera Payload** is responsible for the mission of the satellite which is to take pictures of Denmark

**Transmitter/Receiver Device (TRD)** is responsible for transmitting and receiving communication between the satellite and the ground-station. The TRD consists both of the specific radio-device as well as interface circuitry

The following table states for each load the requirements on voltage, minimum power (idle mode), maximum (active) power and the current-drain at which the subsystem is to be shutdown:

| Load | Voltage [V] | Minimum [mA] | Maximum [mA] | Shutdown current [mA] |
|---|---|---|---|---|
| OBC | 5 | 100 | 200 | ? |
| ACS | 5 | 14 | 80 | ? |
| CAMERA | 5 | 0 | 50 | ? |
| TRD | 5 | 90 | 940 | ? |
| Total | - | 204 | 1270 | - |

**Table N.1:** Table of load requirements

These figures are based on the information that is supplied by those responsible for each subsystem. If it is found that these numbers are outdated or invalid it is the responsibility of each subsystem to provide new numbers.

A conceptually diagram of the PSU can be seen on figure N.1.

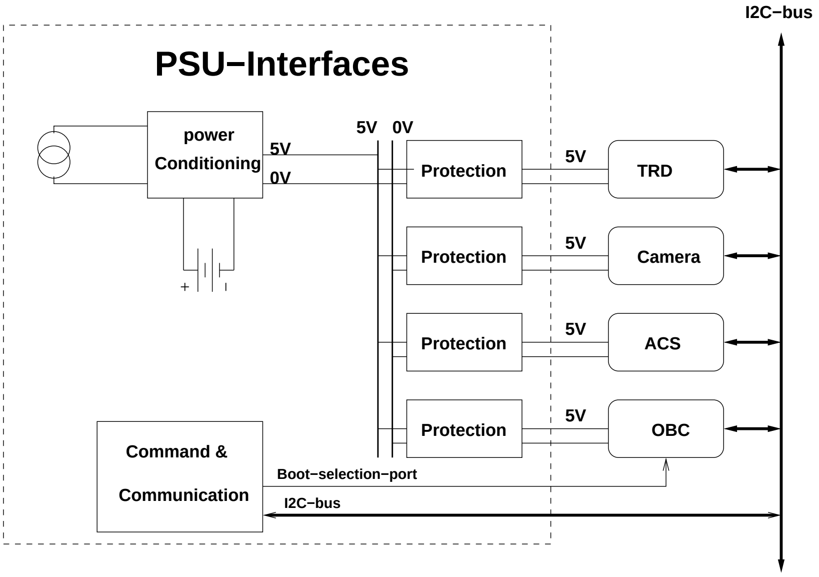


**Figure N.1:** Conceptual diagram of the PSU. No internal details shown

## N.1.6 Interface Descriptions

The following sections each describe an interface that is associated with the power PSU. The purpose of the interface is first described where after physical implementation and functional behavior are explained. Finally issues that are not yet determined are mentioned and it is stated how, when, and by who these points should be determined.

## N.2 Power Distribution Interface

### N.2.1 Purpose

The purpose of this interface is to distribute power to each load and provide load protection for each load.

### N.2.2 Physical Interface

This interface consists of one power wire and a ground wire for each load. The voltage difference between the two lines are 5.0 V for all loads.

### N.2.3 Interface Functionality

#### N.2.3.1 Output Voltage Regulation

The voltage drop between the two output terminals must be controlled such that it under normal operation remains within 2% of accuracy, i.e. between 4.95 V and 5.05 V. When loads are shifted on or off the powerbus then deviations of ±2% are accepted for a duration of less than 20ms, i.e. between 4.9 V and 5.1 V

#### N.2.3.2 Load Protection

Each connected load has a specified maximum current that it is allowed to consume, if this limit is exceeded then the load is disconnected from the power-bus by the PSU. This event is informed to the DHS and the load is turned on again when requested by the DHS. This communication uses the communication interface that is described later.

#### N.2.3.3 Load Protection of the OBC

Since the DHS is situated on the OBC the above protection mechanism cannot be used because obviously the DHS cannot tell the PSU to turn on the OBC again. Therefore another mechanism is used for the OBC load protection.

If the OBC-load current exceeds the threshold it is shutdown by the PSU and then again turned on when 5 minutes have elapsed. After turn-on the event is communicated to the DHS. The waiting is meant to allow the temperature of the OBC-circuitry to drop in case the error condition was due to thermal effects.

### N.2.4 Issues to be Determined

- The exact maximum currents for each load must be identified in order to implement the protection circuitry. These numbers are not expected to be available before all subsystems have been implemented.

## N.3 DHS Communication Interface

### N.3.1 Purpose

The purpose of this interface is to provide DHS with information from the PSU. This information includes:

- information on power-status
- information on status of loads
- general housekeeping information from PSU

It is also possible for the DHS to shut loads on or off using this interface.

### N.3.2 Physical Interface

The physical interface is the I2C-bus that is chosen as the main internal communication bus of the satellite.

### N.3.3 Interface Functionality

#### N.3.3.1 Communication of House-keeping Information

The following data is available for the DHS (Numbers in parenthesis shows the number of data):

- Voltage across parallel connection of solar-panels (1)
- Currents through solar-panels on each side of the satellite (5)
- Voltage across parallel connection of batteries (1)
- Voltage on 5V bus (1)

- Currents to each user (OBC, ACS, CAM, TRD & PSU) (5)
- 7 temperature measurements (7)

Above values are all described with 12bit words and information is updated on the PSU at least once every 10 seconds. The temperature measurements will be used to measure hot-spots in the satellite and this gives about 1 or 2 temperature measurements for each load.

#### N.3.3.2 Communication of Load On/Off status

If the PSU load-protection mechanism shuts down a load because it exceeds its maximum current a signal is sent to the DHS in order to notify it of the event and identify the load that was shutdown. If the PSU has shut down the OBC and turned it on again after the specified period of time then a signal is sent to the DHS after reactivation in order to inform about the event.

The DHS can by sending a signal order the PSU either to shut-down or turn on a connected load. This does however not apply for the OBC-load, which can only be shut down by the PSU in case of a protection fault. DHS can at all times request information on which loads are turned on or off.

### N.3.4 Issues to be Determined

- it is to be decided where hot-spot temperature sensors are to be located on the satellite.

## N.4 P-POD Kill-switch Interface

### N.4.1 Purpose

The satellite must be equipped with one or two switches that makes sure that no on board systems are turned on before deployment of the satellite from the launch vehicle. This is a requirement of the P-POD deployment mechanism. Further there must be a delay of 3 seconds after deployment before any electrical components are activated.

### N.4.2 Physical Interface

*Is to be determined by Mech-group and power groups*

### N.4.3 Interface Functionality

When the satellite leaves the P-POD deployment tube the activation of the kill-switch must turn on the PSU which after successful initialization turns on the OBC-load.

### N.4.4 Issues to be Determined

- the physical interface is to specified.

## N.5 OBC Boot Selection Interface

### N.5.1 Purpose

The purpose of this interface is that the OBC needs an external subsystem to help it select boot-mode for the OBC. Further the OBC needs an external watchdog.

Since it is the PSU that must cycle the OBC power in case of a malfunction the PSU has been chosen to implement the above functionality.

### N.5.2 Physical Interface

The physical interface consists of a boot-selection port which is simply a connection from an output port on the PSU digital hardware to the OBC. Further the I2C-bus is used for both the watchdog functionality and to set the value of the boot-selection port.

### N.5.3 Interface Functionality

The functionality of this interfaces is described in the OBC documentation and will not be stated here.

### N.5.4 Issues to be Determined

- The period of the external watchdog functionality is to be determined by the OBC-group