NTNU
Norwegian University of
Science and Technology

Power Distribution and Conditioning for a Small Student Satellite

Design of the NUTS Backplane & EPS Module

Dewald De Bruyn

Master of Science in Engineering Cybernetics
Submission date: June 2011
Supervisor: Amund Skavhaug, ITK
Co-supervisor: Roger Birkeland, IET

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Problem Description

The NUTS (NTNU Test Satellite) project was started in September 2010. The project is part of the Norwegian student satellite program run by NAROM (Norwegian Center for Space-related Education). The projects goal is to design, manufacture and launch a double CubeSat by 2014.

As part of the NUTS satellite design, there is a need for a system backplane, which different sub-modules can be plugged into. The backplane will form the basis of the satellite, and must provide distribution of power and communication buses to the rest of the system.

The Electrical Power System is a very important part of any satellite mission, and must handle power conversion and battery charging, as well as provide regulated supply to the rest of the system. Together with the system backplane, an EPS module has to be designed for the NUTS satellite.

The candidate will:

- Gain an overview of the project and outline the necessary requirements and constraints.
- Become familiar with previous work on the subject.
- Propose a solution for a system backplane and electrical power system.
- Implement and test prototypes of the solutions, as extensively as allowed by the project time frame.
- Evaluate the proposed design and the results obtained from testing.

Assignment given:	10. January 2011
Supervisor:	Amund Skavhaug, ITK

Summary & Conclusion

The NTNU Test Satellite project aims to launch a 10x10x20 cm nanosatellite, weighing less than 2.66 kg. The satellite, which follows the CubeSat standard, is a relatively low cost project, and explores the use of comercially available, low cost electronic components in space. This work is mainly concerned with power management and distribution for NUTS, but also describes the design of a system backplane.

The thesis is split into two parts. The first describes the design and evaluation of the backplane with main focus on power distribution, and the second part describes battery charging and power conversion in the Electrical Power System (EPS) unit.

The backplane has been designed with flexibility, fault-tolerance and simplicity in mind. The design is based on a single I^2C bus with bus repeaters for each submodule, with the ability to isolate individual modules from the system in case of a malfunction. Power is distributed with dual 3.3 V and dual 5 V busses working in active redundancy, ensuring continued operation should a voltage converter fail. Power distribution for each module consists of three parts: power supply or-ing, current-limit switch and power monitor, and is integrated into the backplane. The state of the power switches and bus repeaters are controlled from two master modules, and a watchdog timer ensures return to a default state should both master modules be disabled.

The proposed solution integrates all the management logic and power distribution circuitry into the backplane itself, leaving the valuable PCB space on the submodules free for module implementations. The power distribution unit has been evaluated with a separate prototype, and displays good performance in terms of power losses and response to over-current conditions. A prototype implementation of the backplane has been produced.

The electrical power system of the satellite consists of solar panels, batteries, and voltage converters. With very limited power available, the main focus of the design has been to reduce losses in power conversion. A battery charge regulator (BCR) has therefore been proposed, which charges the battery while keeping the solar panels at their maximum operating point. This integrates a *Maximum*

Power Point Tracking stage with a battery charger into a single converter, greatly increasing efficiency. Redundancy can be achieved by using separate converters for separate panels in parallel. A prototype BCR has been tested, and initial results show over 91% converter efficiency.

Contents

List of Figures

Part I

Overview

Chapter 1

Introduction

Access to space for universities has traditionally been limited by the extreme costs associated with space missions. Large communication satellites can have a total cost of ownership of more than $100 million. Recent developments in nano- and picosatellites has opened up a whole new frontier for universities and other organizations with small budgets and time frames, allowing payloads to be launched with a fraction of the cost of incorporating it into a traditional satellite.

A nanosatellite is defined as an artificial satellite with a mass between 1 and 10 kg, while a picosatellite weighs less than 1 kg. These small satellites are usually launched as passengers alongside larger satellite missions. [16]

As solar cells increase in efficiency, the amount of useful work that can be achieved with nanosatellites is also increasing. These small satellites can be used to test novel components for future space missions, or to perform specific scientific missions on a small scale. Many universities around the world have taken an interest in the development and launching of nanosatellites, which provides unique educational opportunities spanning many fields of engineering. [9]

A typical nanosatellite mission differs greatly from the development and deployment of tradition satellites in a number of ways. Firstly, the development costs and development time is drastically reduced by scaling the design down, and by using *commercial off the shelf* (COTS) components. Secondly, the mission duration of nanosatellite missions is often very short compared to commercial missions, usually around $3-6$ months of operation instead of $5-10$ years. The combination of low cost and short mission lifespan allows new and innovative components to be used which would not normally be considered for a satellite mission.

A number special considerations and limitations apply to nanosatellites that do not normally apply to larger satellites, requiring special design. For example, in a commercial satellite mission, the solar panels are dimensioned so that the end of life capacity is high enough to support the payload until the end of the mission timespan. In a nanosatellite, the amount of available power is very limited, and

not much can be done to increase this, so the payload and mission length has to be adapted accordingly, and power-efficiency in the satellite systems becomes much more important. Other limiting factors are volume and weight constraints.

Despite having serious constraints on available power, volume and weight, nanosatellites have some great advantages. Since the mission duration is relatively short, and the investment is small, requirements on reliability and redundancy are greatly eased. This allows for greater flexibility in the system design, and allows other factors such as simplicity and power-efficiency to be weighted. Because of the very low cost of launching a nanosatellite, the common approach held by many developers is that redundancy can simply lie in the ability to launch another satellite.

Although the development of nanosatellites differs significantly from traditional satellite design, it also differs from the design earth-based systems. The space environment can be very harsh, as radiation, vacuum and temperature effects have to be taken into consideration. Possibilities for maintenance or recovery are virtually non-existent once the satellite has been launched. Since the satellites are usually launched as passengers together with other satellite missions, considerations has to be taken to prevent the nanosatellite from influencing the main mission. This includes thermal vacuum bakeout to ensure proper outgassing of components, which can otherwise effect the solar cells, optics or other systems of other satellites.

1.1 NTNU Test Satellite Project

The NTNU Test Satellite, also abbreviated "NUTS" is a project aiming to launch a nanosatellite into Low Earth Orbit (LEO). The satellite, measuring only $10 \times 10 \times 20\,\text{cm}$, and weighing less than $2.66\,\text{kg}$, conforms to the CubeSat Standard, which allows easy and affordable access to space for universities and other organizations.

The NUTS (NTNU Test Satellite) project was started in September 2010. The project is part of the Norwegian student satellite program run by NAROM (Norwegian Centre for Space-related Education). The projects goal is to design, manufacture and launch a double CubeSat by 2014. The national student satellite program involves three educational establishments, namely the University of Oslo (UiO), Narvik University College (HiN) and NTNU.

As main payload, the NUTS project will fly an IR-camera for atmospheric observations. In addition, a concept for a wireless short range data bus connecting different subsystems will be added. For communication, the satellite will use the common amateur radio bands and fly one transceiver for each frequency. During the first half of 2011, ten students from different departments and curriculums were involved in the project.

The work outlined in this thesis has been done in the early stages of the project's

development, and can in a way be viewed as a bottom-up approach to satellite design. Many high level mission parameters and decisions are left for later, when more information about the payload and launch parameters become available.

The main goals of this thesis work has been to lay down a basis with which further development can proceed. The overall architecture of the satellite is based around a backplane with multiple module cards. The backplane provides common power and communication buses, while the rest of the satellite functionality and payloads are divided into several modules. Because of the limited development time available, it has been very important to quickly get a working prototype of the system backplane up and running, so that development of other modules can proceed.

A very important part of any satellite design is the power management system, as the satellite cannot function at all without power. The power system must provide constant regulated supply to all the modules, but must also be able to control the power drain in order to keep the battery voltage within safe limits. Furthermore, the system must be protected from anomalous situations such as short circuit faults.

1.2 Related Work

NTNU has previously been involved in the NCUBE student satellite project together with other Norwegian universities. Two satellites were constructed, NCUBE-1 and NCUBE-2. NCUBE-2 was launched first on October 27, 2005, but radio contact with the satellite was never achieved. It has later been tracked and identified by radar. NCUBE-1 was launched on July 26, 2006, but a problem with the second stage of the rocket caused a launch abort. Both NCUBE satellites were single CubeSats, measuring $10 \times 10 \times 10\,\text{cm}$, and weighing around 1 kg.

In the autumn semester of 2006, three master students started from scratch with a new specification for the design of a small student satellite. The new satellite was specified to be a double CubeSat unit, with dimensions of $10 \times 10 \times 20\,\text{kg}$. It is their report, "Design of a Small Student Satellite" [5] that forms the basic grounds for this project. With this report, a new CubeSat project at NTNU was initiated, largely moving away from the previous designs of the NCUBE satellites.

Since then a few other students have written project reports or thesis around the design and specification of different sub-systems for the NUTS project. A brief overview of previous work at NTNU concerning power management for NUTS is presented here.

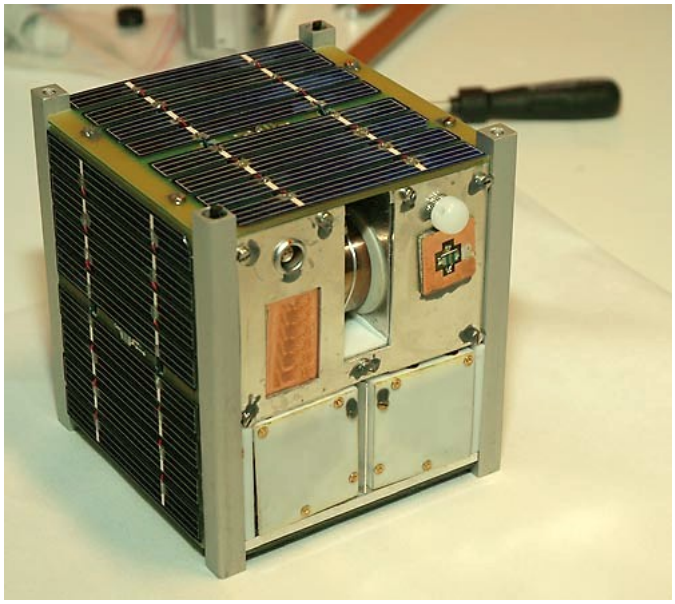


Figure 1.1: NCUBE2, a 1U CubeSat designed in Norway

1.2.1 Birkeland, Blom & Narverud, autumn 2006

The report by Roger Birkeland, Elisabeth Blom and Erik Narverud still serves as the main specification document for the NUTS satellite project. The choice of satellite platform was evaluated, and it was decided to conform to the CubeSat standard, with the double cube configuration.

They provide a general system specification, including a preliminary weight and energy budget for the satellite. It includes a considerations on the mechanical system and structure, antenna systems, power management, data handling, and the communications system. Thermal and orbital considerations are also taken into account. A distributed architecture is chosen for the organization of different system modules in the satellite, with modules communicating together on a shared communication bus. [5]

1.2.2 Edgar Elden, autumn 2008

Elden addresses the power management problem, and proposes a strategy for charging the batteries from the solar cells, and distributing both batter power and raw solar cell voltage to the payload. The design is centered around a Master Power Manager, which has the responsibility to control the power of individual system modules, as well as perform error recovery should the on board computer (OBC) encounter problems. He proposes separate power management controllers

for the payload and other system modules, consisting of power switches, fuses and a microcontroller for individual payload power management.

In his design, the Master Power Manager controls the charge and discharge of the battery, but leaves the responsibility of voltage regulation to the individual modules. [10]

1.2.3 Ingulf Helland, autumn 2008

In his project report, Helland addresses the problems and design of a central communication bus. He proposes a redundant bus system, with two equivalent busses providing both communication and power. This solution provides full redundancy in case one of the busses should fail. Each module contains two slave bus controllers, two sets of power switches and fuses, a bus selection multiplexer/demultiplexer and an intelligent power regulator. Support also exists for several addressable units within each module card, requiring additional multiplexing/demultiplexing circuitry. In his design, Helland estimates about 50% of the PCB area on each module to be occupied by bus logic and power regulation & control.

Helland also presents a significant effort in failure analysis of the design, and also presents some general guidelines for failure analysis for the NUTS project as a whole. [11]

1.2.4 Lars O. Opkvitne, spring 2009

The report by Lars Opkvitne is the most recent contribution to the work on a Power Management System for the NUTS project. Opkvitne presents a thorough study of power management strategies for small satellites. A few different strategies for power management is discussed including the use of MPPT (Maximum Power Point Tracking) and DPPM (Dynamic Power Path Management).

The design he proposes is again based on two redundant power buses. He proposes a system where half the solar cells are used to charge half the batteries, with two power systems in hot redundancy. Figure 1.2 is an overview of the power management system as proposed by Opkvitne. A prototype has been constructed which tests the battery charging and state of charge (SOC) monitoring circuits. [17]

1.2.5 Commercial Solutions

The related work mentioned so far has only included previous projects at NTNU. As the interest for CubeSats is steadily growing several companies have started to offer complete platform solutions in the form of satellite kits. This has evolved to the extent that one can purchase an entire satellite, incorporate one's own payload, and launch in a very short time span. This is worth mentioning because

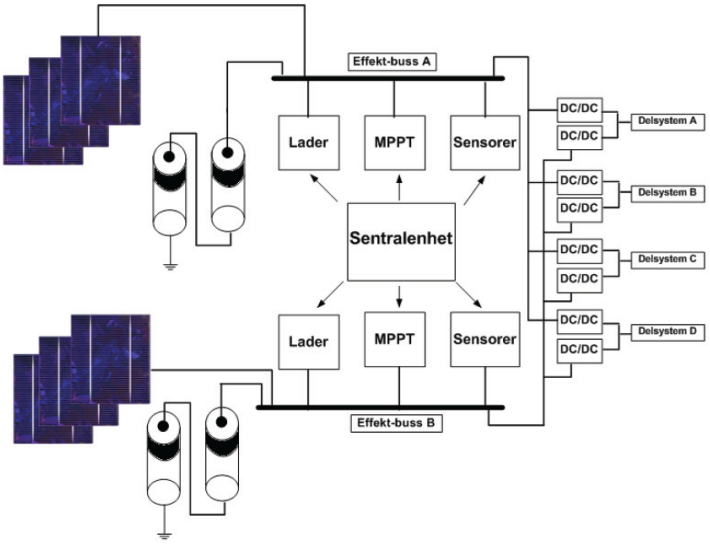


Figure 1.2: Power System Architecture by Lars Opkvitne

many of the problems encountered in this project has already been addressed, and commercial solutions are available. For example the website `www.cubesatkit.com` sells a complete system solution, including chassis, solar panels and several avionics boards from Pumpkin Inc.

Other manufacturers of CubeSat components worth mentioning are GOMSpace, stemming from the University of Aalborg in Denmark, and UK-based Clyde-Space. Both these companies are also involved in small satellite projects other than CubeSats. [1, 2]

The main reason for choosing not to base the NUTS project on existing solutions is the educational value present in designing a satellite from scratch. The wish to design a system around a backplane instead of using the standard PC-104 stacking system also eliminates many possiblities of choosing existing solutions for system components.

Lastly, since the CubeSat Project consists of a collaboration between more than 40 universities, high schools and private firms around the world, it is easy to obtain a lot of useful information about other CubeSat missions. Universities usually release their student's reports, and since many of the problems solved are similar to the ones encountered in the NUTS project, a lot can be learned from studying other successful and unsuccessful projects.

1.3 Scope & Disposition

Most of the previous work done in the NUTS project has been mainly theoretical work. Many important aspects of the design of a small student satellite has been covered in the different reports, but little actual hardware for further development in the project has been produced. Furthermore, many of the proposed solutions has focused mainly on reliability through redundancy, with less focus on simplicity and power, weight and area efficiency.

In this work, an overview is presented of the problems and challenges associated with power management in a small satellite, and a design proposal is made where the power management problem is split into two parts. The two parts are power distribution and power conditioning, to be implemented as separate hardware units in the system. The power distribution part is integrated into a satellite backplane together with a communication bus and control logic, and the power conditioning is designed as a module that plugs into the backplane.

This work presents the design of the entire backplane, including power distribution and protection, communication bus and control logic. A fully functional hardware prototype has been produced that will form the basis of further satellite development. A proposed design for the electrical power system (EPS) module will also be presented, which handles charging batteries from solar cells and converting power for use in the rest of the system. A functional prototype of a battery charge regulator (BCR) with maximum power point tracking has been produced, which forms an important part of the EPS.

Chapter 2 presents some important background information and theory necessary to understand the problems posed. This includes information about the CubeSat standard, general theory about satellite power systems, a simple design philosophy and an overview of the different NUTS systems.

Part II is dedicated to the NUTS Backplane. Chapter 3 gives an overview over the backplane design requirements and specifications and chapter 4 details the implementation of the required functionality and the design of the circuit board. Chapter 5 presents a test setup used to test the power distribution module, as well as a functional verification of the backplane prototype. The results are presented in chapter 6, and discussed in chapter 7, and a short conclusion with recommendations for further work is given in chapter 8.

Part III is dedicated to the EPS module. Chapter 9 describes the design and architecture of the power conditioning system, and presents some functional requirements. Chapter 10 presents the implementation of a BCR prototype, and testing of the BCR is presented in chapter 11. The results are presented in chapter 12, and discussed in chapter 13, and a short conclusion with recommendations for further work is given in chapter 8.

Finally, part IV provides a combined summary and conclusion for the project

Chapter 2

Background

To put this work in perspective, a brief overview of satellite power systems is given in this chapter. This will help to map the specific requirements and challenges related to power management for the NUTS nanosatellite. The CubeSat Standard is also presented, as this provides a set of external requirements to the project to make it eligible for launch.

2.1 The CubeSat Standard

The CubeSat Project is a concept developed by the Polytechnic University of California (Cal-Poly) and Stanford University to ease the development and deployment of picosatellites. By conforming to this standard, a satellite can be launched using the Poly Picosatellite Orbital Deployer, or P-POD, developed at Cal-Poly. The P-POD is a tubular, spring loaded mechanism designed for deploying up to three CubeSats, and can be integrated into almost any launch vehicle.

A basic CubeSat is a $10 \times 10 \times 10$ cm cube weighing less than 1.33 kg. A satellite with the dimensions of two or three cubes can also be constructed and launched, called a double CubeSat (denoted 2U) or a triple CubeSat (denoted 3U). The P-POD can then either be loaded with three 1U satellites, one 2U and one 1U, or one 3U Satellite.

2.1.1 Mechanical Requirements

The CubeSat Design Specification [6] consists of a set of requirements that need to be fulfilled in order to ensure compatibility with the P-POD, and to protect the launch vehicle, it's primary payload and other CubeSats. Conforming to this design specification is necessary to be able to launch a CubeSat.

Some important mechanical specifications from [6] are:

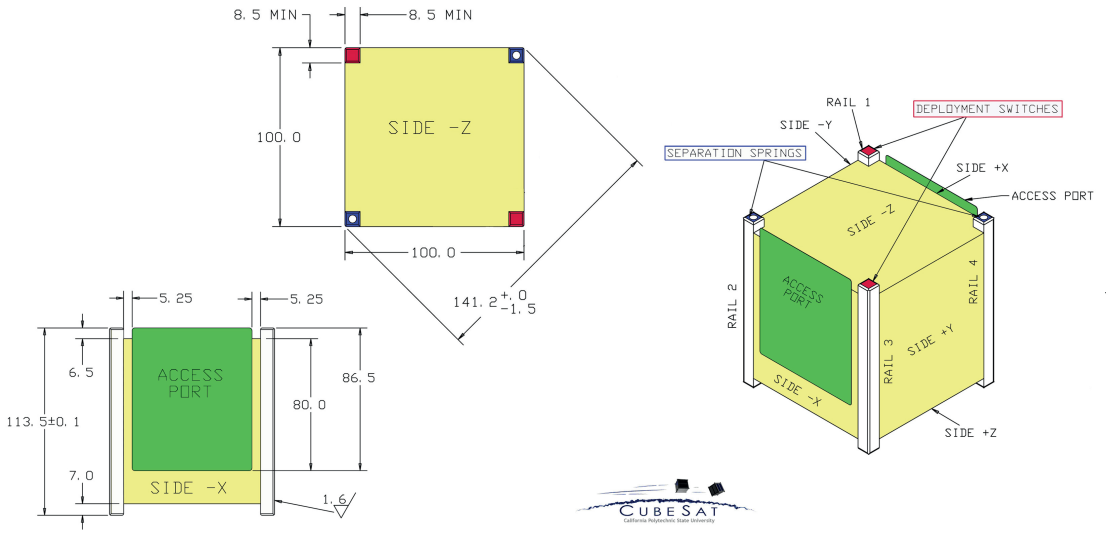


Figure 2.1: 1U CubeSat Dimensions

- The CubeSat shall be 100.0±0.1 mm wide.
- A single CubeSat shall be 113.5±0.1 mm tall, a triple CubeSat shall be 340.5±0.3 mm tall.
- Contact with the rails in the P-POD shall have a minimum width of 8.5 mm.
- Each single CubeSat shall not exceed 1.33 kg mass, each triple CubeSat shall not exceed 4.0 kg mass.
- The CubeSat center of gravity shall be located within a sphere of 2 cm form it's geometric center.

Figure 2.1 contains information about the coordinate system used to define the CubeSat dimensions. Although the official specification only contains information about single and triple CubeSats, the dimensions for a double CubeSat can easily be derived: A double CubeSat shall be 227.0±0.2 mm tall, and shall not exceed 2.66 kg mass.

In addition, the mechanical specification also defines the use of separation springs for separating the different CubeSats after they have been deployed by the P-POD.

2.1.2 Electrical Requirements

The CubeSat Design Specification also contains requirements to the electrical system, mainly to prevent interfering with the launch vehicle, the primary payload

or other CubeSats. No electronics shall be active during launch to prevent any electrical or RF interference. A deployment switch, which is depressed when the satellite is in the P-POD, is used to turn off satellite power during launch. According to the specification, all systems has to be turned off, including real time clocks. As the satellite is deployed, the deployment switches are released, and the satellite can be powered up.

In addition to the deployment switches, the CubeSat must also implement a Remove Before Flight (RBF) pin. This pin disconnects all power, and is removed after the CubeSat is integrated into the P-POD. The CubeSat Design Specification document includes designated areas for the RBF pin and for umbilical connectors, which can be used for diagnostics and battery charging after the CubeSat has been integrated into the P-POD.

After being ejected from the P-POD, the satellite must wait at least 30 min before it can begin transmitting with RF transmitters greater than 1 mW. Deployment of antennas or other structures also have to wait a minimum of 30 minutes after leaving the P-POD. Another requirement present in [6] is that CubeSats with batteries shall have the capability to receive a transmitter shutdown command.

2.2 Satellite Power Systems

The electrical power system of any satellite mission is considered one of the most critical components. Without electrical power, a satellite in orbit becomes nothing more than space debris. Almost all spacecraft orbiting the earth today use solar cells as the primary energy source, backed up with batteries to sustain power during eclipse.

Some alternatives to solar power exist for special cases, such as hydrogen fuel cells used on board the space shuttles, or radioisotope thermoelectric generators (RTG) used on board the Cassini and Gallileo missions. These are however considerably more expensive, and are only used when solar cells become impractical, for example if the spacecraft is too far away from the sun.

In the design of satellites, the size of solar panels and batteries often become the driving factor in dimensioning the satellite size and weight. The required size of solar panels and batteries is dependent on the requirements of the payload, and the lifetime of the satellite mission. Both solar cells and batteries lose efficiency over time, so it is important to dimension them so that the end of life (EOL) capacity is sufficient to sustain the payload and satellite systems. For long missions, this generally leads to significantly larger solar panels and batteries compared to the requirements at the beginning of life. [4]

For a satellite in LEO the light from the sun will frequently be blocked by the earth as the satellite travels behind it. This is called eclipse, and causes the

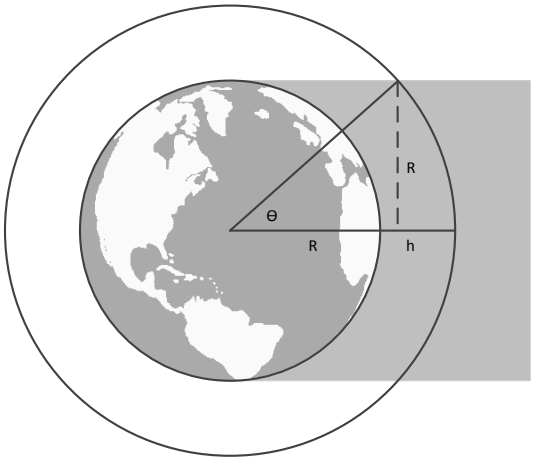


Figure 2.2: Eclipse

satellite to rely completely on battery power for a portion of every orbit. Knowing the maximum eclipse duration is important in dimensioning the power budget and the on board battery.

As an example, consider a satellite in LEO, at a height of 600 km above the earth surface. To gain a rough estimate for the maximum time spent in eclipse, it is sufficient to approximate both the earth and the orbit as spherical. The time for one orbital period is given by Kepler's Third Law as:

$$\tau^2 = \frac{4\pi^2}{\mu} a^3 \Leftrightarrow \tau = 2\pi a \sqrt{a/\mu} \tag{2.1}$$

with the semimajor axis a given in km and

$$\mu = GM = 398\ 601\,\mathrm{km}^3/\mathrm{s}^2 \tag{2.2}$$

(the universal constant of gravitation $G = 6.6742 \times 10^{-11}\,\mathrm{m}^3/\mathrm{kg}^2/\mathrm{s}^2$ and the earth's mass $M = 5.9736 \times 10^{24}\,\mathrm{kg}$.) With an orbital height of 600 km, $a = (R+600)\,\mathrm{km} = 6971\,\mathrm{km}$, the orbital period becomes $\tau = 5792\,\mathrm{s} = 96.5\,\mathrm{min}$.

To determine the longest possible time in eclipse, consider figure 2.2. The time spent in eclipse can be calculated as

$$t_{ecl} = \frac{2\theta}{2\pi} \cdot \tau \tag{2.3}$$

with:

$$\theta = \sin^{-1}\left(\frac{R}{R+h}\right) = \sin^{-1}\left(\frac{R}{a}\right) \tag{2.4}$$

Inserting 2.1 into 2.3 gives:

$$(t_{ecl})_{max} = 2a\sqrt{\frac{a}{\mu}}\sin^{-1}\left(\frac{R}{a}\right) \tag{2.5}$$

With a height of 600 km this gives $(t_{ecl})_{max} = 2125\,\mathrm{s} = 35.4\,\mathrm{min}$.

2.2.1 Power Supply

The most common source of power for satellites is photovoltiac solar cells, which consists of a doped semiconductor p-n junction, and is protected by a UV filter and a layer of cover glass. The cell releases electrons when irradiated with sunlight. Today's most efficent cells uses gallium arsenide (GaAs), with the highest efficiencies achieved by using multiple layers to utilize energy from different parts of the sun's spectrum. Each cell provides a limited voltage and current, so cells are stringed together and strings are placed in parallel to obtain the desired voltage and current ratings.

The amount of power available depends on the solar power flux and the efficiency of the solar cells. The solar power flux in the earth's vicinity varies between $S_{max} = 1399\,\mathrm{W/m^2}$ in early January and $S_{min} = 1309\,\mathrm{W/m^2}$ in mid-summer. An annual mean of $S = 1353\,\mathrm{W/m^2}$ is often assumed for design purposes. [4] The angle of the sun to the panel defines the area of irradiance, and scales the amount of power available. Many larger satellites have deployable solar panels, which are folded up when the satellite is inside the launch vehicle. The solar panels can often be oriented towards the sun using actuators, to maximize the irradiated panel area.

The solar cell efficiency depends on the cell construction, temperature and age. The efficiency of a modern GaAs solar cell is only about 30%. The cell efficiency drops as they heat up, giving the most power when the satellite emerges from eclipse and the panels are still cold. The performance of a solar cell also degrades over time due to damage caused by the influx of high-energy protons and electrons in the Van Allen belts, and due to ultraviolet radiation from the sun. [4]

The IV charactaristic of a commercially available solar cell is shown in figure 2.3. These cells have have a short circuit current $J_{SC} = 17.76\,\mathrm{mA/cm^2}$, and an open circuit voltage $V_{OC} = 2.633\,\mathrm{V}$. Since power $P = I \cdot V$, the power at open circuit and at short circuit are both zero. The cell is exploited most efficiently if the working point is chosen at the "knee" where the output power reaches a maximum. In order for the cells to be used efficiently, a voltage bias is required to keep the operating point close to the maximum power point (MPP). According to the datasheet, the voltage at the MPP is about $V_{mpp} = 2.348\,\mathrm{V}$ and the current $J_{mpp} = 17.02\,\mathrm{mA/cm^2}$. [20]

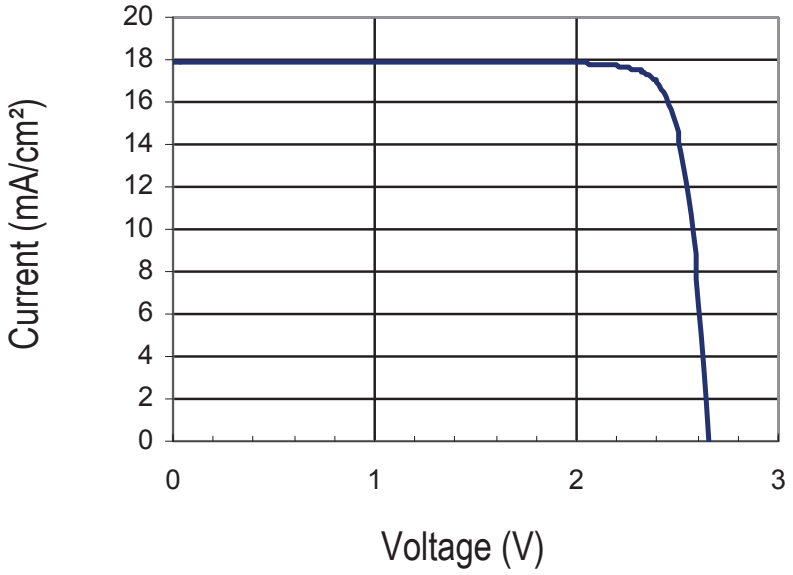


Figure 2.3: Spectrolab 29.5% XTJ Solar Cell Typical I-V characteristic [20]

Orientation	Available Power
One side panel only	4.80 W
Top panel only (minimum)	2.40 W
Two side panels	6.79 W
One side panel & top panel (25° angle)	5.36 W
Two side panels & top panel (maximum)	7.20 W

Table 2.1: Produced power at different orientations to the sun

2.2.2 NUTS Solar Cells

NUTS-1 will most likely have solar cells on five out of six side panels. Four of these panels are approximately $10 \times 20\,\mathrm{cm}^2$ and the top panel is $10 \times 10\,\mathrm{cm}^2$. With standard cell dimensions, four solar cells can be placed on each of the long panels, and two on the top, equaling to a total of 18 solar cells. Normally only a number of the panels will be exposed to sunlight at any given time, resulting in a varying power output depending on the satellite orientation to the sun. In the best case, two side panels and the top panel are exposed at the same time, giving a significantly higher output power than the worst case, where only the top panel is exposed. Table 2.1 shows some possible power outputs based on a configuration with 18 cells from Azurspace. [3]

As can be seen from table 2.1, the available power changes significantly with

the satellite orientation towards the sun. An important calculation would be to determine how much power is available in a worst-case situation. Assume that the satellite is stabilized so that the end without solar cells always faces towards the earth, and that it spins around it's z-axis (refer to figure 2.1). The worst-case scenario occurs when the time in eclipse is the longest possible. A simple illustration of the generated power is shown in figure 2.4. Here the angle θ is defined as in figure 2.2, with $\theta = 0$ being half way through eclipse, and at $\theta = \pi$ the top panel points directly towards the sun. The blue line illustrates the produced power, and has it's characteristic form due to the rotation around the satellite's z-axis. The green line illustrates the average produced power. From this the worst-case average available power can be calculated, and equals to about $P_{avg} = 5.42\,\mathrm{W}$. As power is only produced during the time the satellite is illuminated by the sun, the average power over the entire orbit becomes:

$$P_{avg,orbit} = P_{avg} \cdot \frac{\tau - t_{ecl}}{\tau} = 3.43\,\mathrm{W} \tag{2.6}$$

It should be noted that these calculations have not taken into account the effects of temperature changes of the solar cells as the satellites moves through orbit. As previously mentioned, solar power efficiency decreases with increasing temperature, so having a slight rotation around the satellite's Z-axis is preferable to prevent a single panel from becoming too hot.

It should also be noted that this is a worst-case estimate for the average produced power, and exact numbers will depend on the exact orbital parameters, which are not known at this point in time. This is however a good starting point in determining a power budget for the satellite, as it gives a rough indication of the minimum available power.

2.2.3 Energy Storage

Rechargeable batteries are used on-board satellites to allow satellite equipment to function in eclipse. The energy storage requirements vary with different orbits. For example, a satellite in Geostationary Earth Orbit (GEO) has eclipse transits limited to max 72 minutes during a 45-day window around spring and autumn equinox. The batteries are only used to supply power during these occasional eclipses. On the other hand, a satellite in LEO is in eclipse up to 40% of every orbit, and will have many more charge-discharge cycles. Battery capacity gradually decreases as the batteries are subject to many charge-discharge cycles, but this capacity loss depends on the depth-of-discharge (DOD) of the cycles. Since satellites in LEO are subject to many more charge-discharge cycles, the allowable DOD is much lower than for GEO satellites. In practice this means that the battery capacity for a satellite in LEO has to be larger than for a satellite in GEO

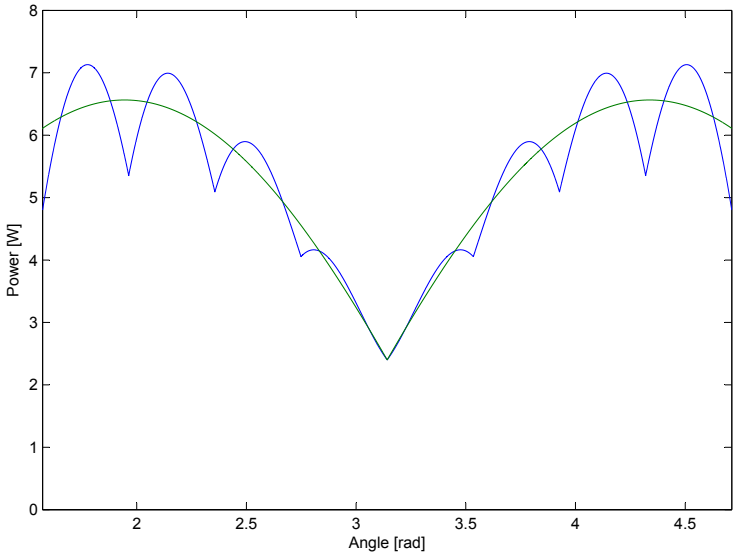


Figure 2.4: Worst-case power vs. θ

with the same power requirements.

Traditionally, satellites use nickel-cadmium (NiCd) or nickel-hydrogen (NiH_2) batteries. They have energy yields of about 40 Wh/kg for NiCd and about 60 Wh/kg for NiH_2. Both battery types have a cell voltage of around 1.2 V, and batteries are placed in series to obtain the desired bus voltage. Both battery types are said to possess a "memory" effect, causing their capacity to deteriorate over time depending on how well earlier charge-discharge cycles were managed. This requires that the batteries be periodically reconditioned by forcing a 100% DOD cycle. [4]

Both the low energy to weight ratio, and the presence of the "memory" effect make traditional battery technology problematic in nanosatellites. New battery technology based on lithium-ion (Li-Ion) chemistries are therefore more common in CubeSats. Li-Ion batteries have a much higher energy to weight ratio ($100 - 250$ Wh/kg), and do not possess the "memory" effect. The typical cell voltage of lithium-ion batteries is usually around $3.6 - 3.7$ V.

Although Li-Ion batteries are not plagued by the "memory" effect, they are still prone to capacity losses over many charge-discharge cycles, and this will eventually limit the lifetime of a small satellite. These batteries can also take permanent damage if discharged below safe limits, and overcharging could lead to the batteries exploding or catching fire. Proper battery protection and charge control is therefore necessary when using Li-Ion batteries.

For the NUTS project, new battery cells from A123 Inc. are being evaluated.

These cells are lithium-ferrite-phosphate cells ($LiFePO_4$), and have a typical cell voltage of 3.3 V. These batteries use a more stable chemistry than traditional Li-Ion batteries, and show great promise in terms of safety, number of charge-discharge cycles and temperature stability. The penalty however, is a slightly lower energy yield compared to Li-Ion batteries, resulting in heavier battery packs.

To calculate the expected worst-case *depth of discharge* for NUTS-1, assume we have $4 \times 1.1\,\text{Ah}$ cells, two in series and two series in parallel. This results in 2.2 Ah @ 6.6 V. Assume that the satellite has a continuous average power consumption equal to the worst-case average power production, calculated in the previous section. The battery capacity used during eclipse becomes:

$$\frac{3.43\,\text{W}}{6.6\,\text{V}} \cdot 35\,\text{min} = 0.52\,\text{A} \cdot 35\,\text{min} \approx 0.3\,\text{Ah} \tag{2.7}$$

Compared to the total capacity of 2.2 Ah, the maximum DOD is:

$$\frac{0.3\,\text{Ah}}{2.2\,\text{Ah}} = 13.9\%\,\text{DOD} \tag{2.8}$$

This is considered suitable for a satellite in LEO. The number of charge-discharge cycles are roughly 15 per day, so over a lifetime of 6 months the number of charge cycles is about:

$$\frac{24\,\text{h} \cdot 60\,\text{min}}{\tau} \cdot 6\,\text{months} \cdot 30\,\text{days/month} = 2700\,\text{cycles} \tag{2.9}$$

Figure 2.5 shows the battery discharge capacity of A123 cells with low current, 100% DOD charge-discharge cycles. This figure shows the main reason why the A123 cells are being evaluated for the NUTS project: a far greater number of charge cycles is possible compared to conventional Li-Ion batteries.

2.2.4 Power Distribution & Conditioning

Because of the varying amount of power available from solar cells, and the charging and discharging of batteries, power regulation and conversion is required to provide a stable power supply to satellite systems. Power conversion and regulation is known collectively as power conditioning, and many different strategies are in use. In a large satellite, a regulated bus voltage is usually distributed to on-board electronics at a voltage of $28 - 100\,\text{VDC}$. The high voltage is chosen to reduce resistive power losses, as $P = R \cdot I^2$.

Three basic topologies, as presented in [4], are illustrated in figure 2.6. The *unregulated bus* basically connects the batteries directly to the solar cells, and provides the lowest level of bus regulation. The bus voltage will vary with varying

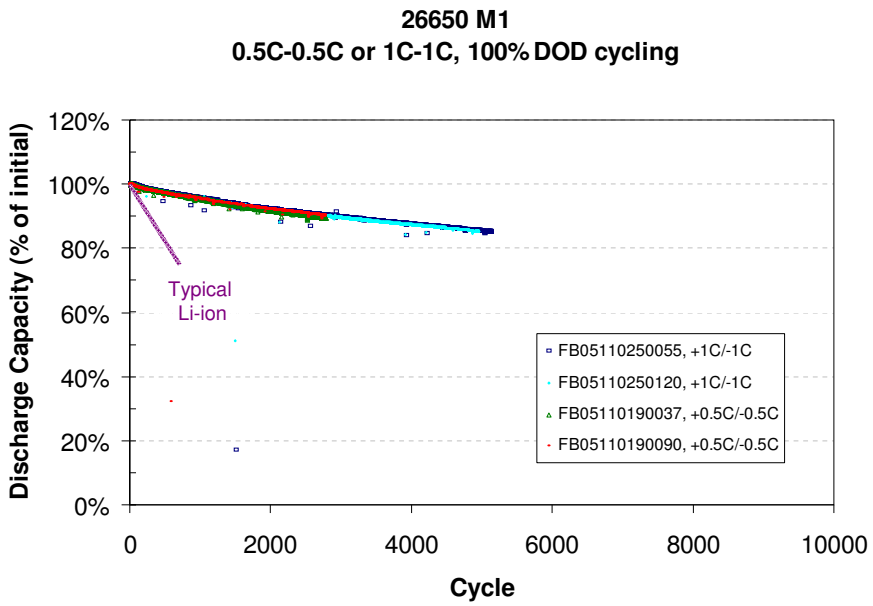
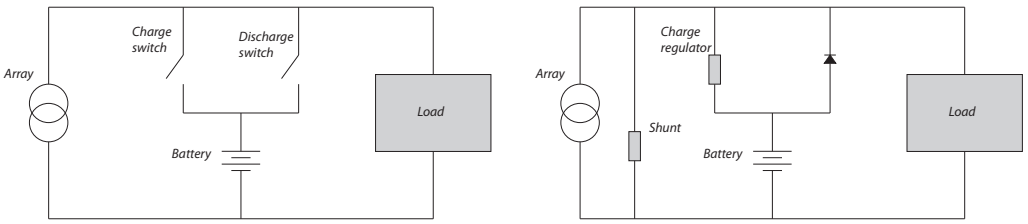
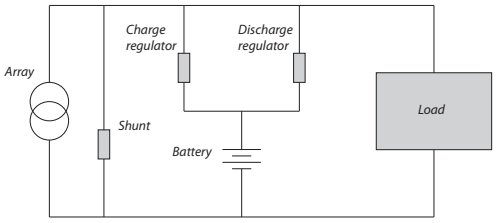


Figure 2.5: A123 Systems Inc. Capacity vs. Charge Cycles [12]



(a) Unregulated bus

(b) Sun-regulated bus



(c) Fully regulated bus

Figure 2.6: Basic levels of bus regulation [4]

solar conditions and battery state-of-charge (SOC). The main challenge with this topology in relation to smaller satellite missions is matching the battery voltage to the solar panel voltage such that the panel's operating point is close to it's maximum power point. As the battery voltage varies, the operating point will move away, giving reduced solar panel efficiency. With the *sun-regulated bus*, voltage regulation is provided during the sun-lit phase of the orbit, while the battery's relatively constant output voltage provides adequate regulation during eclipse. The *fully regulated bus* provides voltage regulation in all phases of the orbit.

More advanced bus regulation strategies involve maximum power point tracking (MPPT), or array switching regulators, where only a number of array strings are connected as the power requirements vary. Another popular strategy is to use a sequential switching shunt regulator (S3R), where a main error amplifier controls the number of shunts connected across the solar panels in order to maintain an operating point close to the MPP. [21]

The power and bus regulation requirements of a small nanosatellite are considerably different than those of a large satellite. Firstly, the amount of power required is in the order of a few watts for a nanosatellite, while large communications satellites can have power requirements in the order of several kilowatts. Secondly, the electronics of large satellites are constructed in separate modules and spread out across the entire satellite. This constitutes a high voltage bus to overcome resistive losses. In a nanosatellite, the maximum distance for power transmission is a few centimeters (about 30 cm for a 3U CubeSat).

These fundamental differences allows for a few simplifications to the bus regulation in a nanosatellite. The requirement for a high bus voltage becomes impractical as the number of available solar cells is not enough to get a high bus voltage. It also becomes somewhat unnecessary, since resistive losses are small as both the required power and the length of power transmission are small.

For a nanosatellite, the use of an MPPT is highly desirable to get the most out of the available solar power, and battery charge regulation is very important as Li-Ion batteries are likely to be used. Commercial CubeSat power systems such as those provided by Gomspace or Clyde-Space typically use a battery regulated bus where the battery is always connected to DC/DC converters powering the load. Power from the solar cells are fed through one or more battery charge regulators (BCRs), which also employ maximum power point tracking.

2.3 Design Philosophy

Several factors govern the design choices taken throughout this project, such as simplicity, reliability, efficiency and cost. The design is constantly weighed against

these factors, and sometimes compromises has to be made.

2.3.1 Simplicity

Design simplicity has been one of the most important driving factors during this work. The general opinion is that simplicity is closely related to reliability, as overly complicated solutions generally increase the probability of design errors or unanticipated system states.

Large satellite missions often use redundancy and majority voting systems to ensure reliability in critical systems, and implementing such systems usually leads to enormous complexity. In CubeSat projects redundancy and complicated fault tolerance mechanisms are often discarded for the sake of overall simplicity.

Simplicity in the proposed solutions is also important considering the project scope and time frame. Many different students will end up working on the same platform, and overly complicated solutions will make integration of the different students' work very challenging and time consuming. Simplicity means designing solutions that are easy for other students to pick up and continue working on.

2.3.2 Reliability

Reliability is another factor that drives the design of systems for the NUTS project, even though this can sometimes be in conflict with the desire for simplicity. In practice this means avoiding single points of failure, and protecting the global satellite mission from local faults.

The main strategy implemented in this project has been to isolate faults by providing the possibility to cycle power to individual modules. In some cases this can also provide error recovery, depending on the nature of the fault. Isolating errors is important, as it can allow the satellite to continue to function, though with possibly reduced functionality.

The desire to optimize for reliability also places some guidelines on component choices. Sufficient de-rating is required to ensure components are not over-stressed, and the operating temperature range of components must be acceptable.

2.3.3 Power Efficiency

Because of the very limited available power in a nanosatellite, power efficiency is one of the key concerns. Implementing important system functions with as little power consumption as possible allows for more of the available power to be used for the payload and radio subsystems.

This work focuses mainly on power conditioning and distribution, and it is very important that these functions be implemented with minimal losses. As an

example, this has driven an approach where the number of regulators and switches in the power path is kept low, since each component incurs losses. The desire for good power efficiency also influences the choice of components and converter topologies.

It is possible that the need to optimize power efficiency influences the simplicity or the reliability of the system. For example, the desire for power efficiency constitutes the use of an MPPT, but this will inevitable reduce the simplicity of the solution, and possibly also the reliability, as the added component has a chance of failing.

2.3.4 Cost & Availability

The last factor that plays an important role in driving the design parameters is the cost and availability of components. Most space missions use special space grade or radiation hardened components that drives the cost of electronics to extremes. As an example, a simple logic circuit in the 74xx series can cost several hundred dollars each in space-grade variants, compared to a few cents for normal components. This is usually deemed necessary, as it is unacceptable for components to fail during the mission time span. In addition to being very expensive, space grade components are also generally harder to obtain than their commercial counterparts. Long production and qualification lead times limits the availability.

Since this is a university project on a comparatively low budget, industrial grade components are preferred. The industrial temperature range is usually specified as -40 to $+85\,^{\circ}\mathrm{C}$. This allows relatively low cost prototypes to be constructed to evaluate the design. The availability of components is also an important factor if functional prototypes are to be constructed within reasonable time frames, as basing the design on industrial grade components allows components to be chosen which are on stock at distributors.

Choosing industrial grade components is definitely a trade off between reliability and cost/availability. However, the low total cost of the project and the limited lifetime of the mission constitute that this trade off is acceptable. [1]

2.4 NUTS System Overview

The NTNU Test Satellite is being designed around a distributed architecture, with a common backplane that system modules can be plugged into. Several necessary subsystems have been defined and design work on them is in progress.

[1] It can also be mentioned that commercially available COTS-solutions for CubeSats, such as products from Gomspace and Clyde-Space specify an industrial temperature range (-40 to $+85\,^{\circ}\mathrm{C}$).

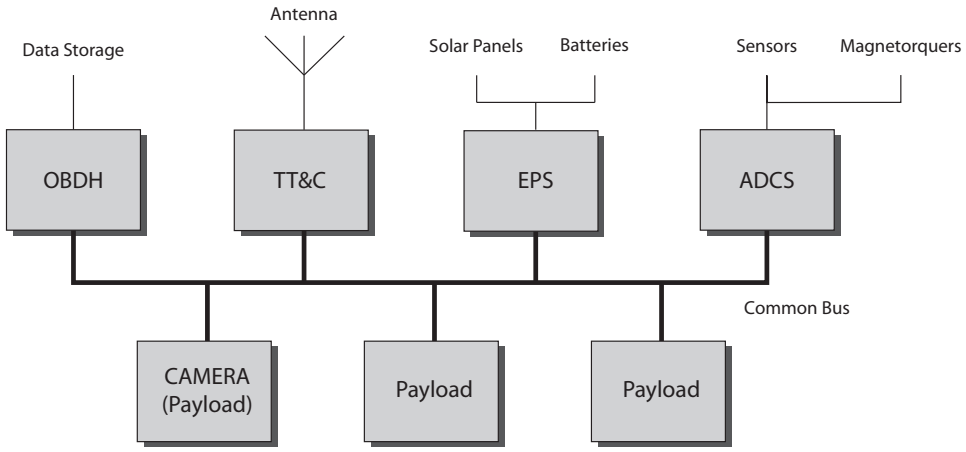


Figure 2.7: NUTS Sub System Modules

Some of the modules that have been defined are:

- On Board Data Handling (OBDH)
- Telemetry, Tracking & Control (TT&C)
- Electrical Power System (EPS)
- Attitude Determination & Control System (ADCS)

In addition to these modules, several different payloads have been suggested and some are being worked on, including a software defined radio and a camera for taking pictures of the earth.

2.4.1 Backplane

The backplane is the medium used to connect the different modules together, and provides communication and power interfaces for the rest of the system. It also provides protection by allowing individual modules to be isolated, reset or powered off.

The design and implementation of a backplane prototype is the main focus of part II of this work. Another student at the Department of Electronics and Telecommunications, Marius Volstad, has also been involved in the design of the control logic on the backplane.

2.4.2 OBDH

The OBDH module is one of two master modules, in the sense that master modules have control over the backplane, while slave modules do not. This module contains the main mission computer and memories for software and data storage. It's main tasks are to monitor the system health, perform logging of flight data and to issue commands to the rest of the system in accordance with the programmed mission.

The module also contains copies of all the system software in a radiation-tolerant read only memory (ROM), to allow the restoring of corrupt instruction memories. Marius Volstad, is currently working on a prototype implementation of the On Board Computer.

2.4.3 TT&C

The TT&C module is the other master module, and provides communication with the satellite. This is considered a very critical system, as loss of communications will mean the end of the mission. The TT&C module contains two radio transceivers, one 145 MHz radio and one 437 MHz radio. In addition to receiving commands from the ground station and transmitting data back, this module also implements a periodic beacon signal, which can be picked up by amateur radios around the world.

This module is implemented as a system master to allow complete control over the satellite from the ground station if the OBDH should fail. The student Asbjørn Dahl at the Department of Electronics and Telecommunications is currently working on the TT&C module.

2.4.4 EPS

The EPS module is responsible for providing a regulated power supply to the rest of the satellite. This includes conditioning power from the solar panels, charging and monitoring the batteries, and voltage conversion from the batteries to the desired bus voltages. The EPS module also contains protection circuits to prevent battery over-charging or over-discharging.

A small low power microcontroller provides telemetry data such as battery state of charge and available power from solar cells. The microcontroller can also be used to implement a simple maximum power point tracking algorithm to increase the solar panel's efficiency.

Part III of this work includes a system level proposal for the implementation of the EPS module, together with a prototype of a battery charge regulator (BCR) used in the EPS.

2.4.5 ADCS

The ADCS module's tasks are twofold: to obtain estimates about the satellite orientation, and to provide de-tumbling and stabilization. Different sensors are used together with powerful algorithms to obtain good estimates about the satellite's orientation. The orientation data is then passed to attitude controller, which uses magnetorquers as actuators to stabilize the satellite flight. Stabilization of the satellite is important to ensure that antenna systems and cameras are pointed in the correct direction.

As the satellite leaves the launch vehicle, it will initially have an uncontrolled rotation about one or more axises. The de-tumbling controller is designed to counteract the rotation by providing torque in the opposite direction, eventually stabilizing the satellite flight.

Currently several students from the Institute of Technical Cybernetics at NTNU are working on the ADCS system.

Part II

NUTS Backplane

Chapter 3

Backplane Introduction

The system backplane is a key component in the NUTS project. It serves as the physical and electrical interconnect between cooperating system modules.

The backplane handles the Power Distribution part the power management problem. In addition to providing power to all system modules, important protection and monitoring features are integrated into the backplane. Additionally, the functionality needed to isolate individual modules to prevent a failed module to affect the rest of the system is also provided by the backplane.

Here the requirements and specifications of the backplane are presented, as well as an explanation of the important design choices made.

3.1 Purpose

The purpose of the NUTS backplane is to:

- Provide a physical connection to system modules.
- Provide a common communication bus to system modules.
- To provide power to system modules.

3.1.1 Physical Connection

The backplane itself is a printed circuit board (PCB) containing connectors for all the sub modules in the system. The connectors on the backplane will define an interface standard that has to be followed by developers of system modules. This interface standard will ensure compatibility with the backplane and other modules if followed correctly.

Aside from defining a set of electrical connections, the backplane connectors also serve in mechanically connecting modules, and thus becomes part of the mechanical structure of the satellite. This means that the choice of connectors has to be evaluated also for their mechanical as well as electrical properties.

3.1.2 Communication Bus

The backplane has to provide a means for system modules to communicate together. This includes the physical communications lines, but also buffers and isolation switches needed to prevent malfunctioning modules from disrupting the rest of the system. The fact that part of the backplane purpose is to provide the ability to isolate individual modules from the communication bus requires that special control logic need to be integrated into the backplane.

3.1.3 Power Distribution

The last main responsibility of the backplane is to provide power to all system modules. This includes protecting modules from short circuits, or more importantly, to protect the rest of the system from a short circuits in any module.

The backplane should also provide power monitoring of individual modules. Monitoring the power use of modules can allow faults to be detected early, before the batteries are discharged to critical levels.

3.2 Requirements

The requirements to the backplane design have been developed based on the defined purpose of the backplane and the design philosophy mentioned earlier. A summary of the backplane requirements are:

- 8 module connectors: two for backplane master modules, one for EPS and 5 general purpose slave module connectors.
- Short-circuit protected 3.3 V and 5 V supplies for each module.
- Power supply protection should include a latch-up recovery mechanism.
- Dual redundant power supply buses for both 3.3 V and 5 V supplies.
- Voltage and current monitoring for both supply voltages for each module.
- Power switches to disable power to individual modules.

- Multi-master capable communications bus with bus isolation for individual modules.
- Debug lines to allow master modules to update program memory of other modules (JTAG/SWD).
- Backplane power & bus isolation switches must be controllable from two master modules.
- Default *reset-state* of backplane must be to enable all modules.
- Integrated watchdog circuit to revert to reset-state.
- The backplane electronics must be protected from short-circuits by the same mechanisms as system modules.

3.2.1 Connector Requirements

Although the backplane should be designed to provide a uniform interface to all modules, some exceptions are necessary. The different types of connectors are: master connectors, slave connectors and an EPS connector. The slave connectors are the default generic module connectors, while the master connectors require extra pins for controlling the backplane. The EPS connector is different because the EPS module has to supply power to the rest of the system instead of being supplied with power from the backplane.

The number of master connectors is limited to two, as it is considered good practice to limit the number of system masters. This is beneficial both in terms of simplicity, as it simplifies master cooperation, and in terms of reliability, as a master module will have a larger chance of disrupting the rest of the system. At the same time, having only a single master in the system could easily cause that module to become a single point of failure, hence the choice of two master modules.

Some requirements exist to the positioning of modules on the backplane. It is desirable that the TT&C module be placed close to the bottom of the satellite, as this will allow for shorter cables from the PCB to radio antenna. It is also required that the EPS connector be placed close to the center, so that the EPS module can reside close to the batteries.

The mechanical requirements to the connectors are mainly that they provide a rugged connection that can withstand the vibrations and shocks experienced during launch and deployment. It is therefore desirable to only use hole-mounted connectors, and larger pitch[1] connectors are preferred. Additionally the current

[1]The pitch is referred to as the distance between pins. Larger pitch usually also means larger diameter and thus sturdier connections.

handling capability of the pins should exceed the expected maximum module current draw.

3.2.2 Power Distribution Requirements

It is required that both 3.3 V and 5 V supplies are distributed to modules, which are common voltage levels in low power systems. This will give good flexibility to the module designer to choose the appropriate voltage level for electronics and sensors.

The requirement for redundant power buses comes from the desire to duplicate voltage regulators, as the loss of the voltage regulator could otherwise lead to the loss of the entire satellite mission. This is again because of the design decision to employ central voltage regulation instead of local regulation on each module. Having redundant power buses will require some mechanism for combining the two separate rails at each module, and it is preferred to incorporate this into the backplane. Combining the supply rails correctly is important, as simply connecting two supplies in parallel will likely cause both to fail at the same time if one of the supply regulators fail.

Short-circuit protection is required for two reasons: to protect modules from short circuits which can cause permanent damage to components, and to protect the rest of the system, as short circuits can damage the central voltage regulators. Current limiting is therefore required on all module supply lines. Additionally, it is desirable to be able to handle *single event latch-up* (SEL) events before they can cause damage, as these errors are likely to be encountered. The basic strategy to provide SEL protection is:

1. Limit the current draw during SEL.

2. Cycle power to the affected module by turning it off for a short time before attempting to turn on again.

Power switches are required to allow the system to selectively power down individual modules. This can be required to save power, or to prevent a faulty module from disrupting the rest of the system.

The requirement of power monitoring is a secondary function, not critical to the backplane operation. It will however provide important telemetry data, which can be useful in determining which modules should be disabled to save power. It is required that the data from telemetry be made available on the system communication bus.

3.2.3 Communication Bus Requirements

The communication bus is needed for coordination and data transfer between system modules. The bus should be multi-master capable, meaning that any module in the system can initiate communication to any other module, regardless if it is a backplane master or not.

Isolation of modules from the communication bus is important, as two types of errors can be prevented with this: a module permanently occupying the bus (continuously pulling at bus lines), and a module randomly disturbing the bus (blabbering idiot). Isolation is also important to allow module power to be turned off without the risk of current leaking into unpowered components.

3.2.4 Backplane Logic Requirements

The backplane must allow the two master modules full control over the power switches and bus isolation buffers. It is a requirement that this functionality be implemented using only standard logic components, to avoid the use of a microcontroller with software on the backplane. This is a general reliability requirement, as a faulty microcontroller can easily become a single point of failure. Furthermore, the backplane logic power supply must be protected by the same mechanisms as system modules.

The default state of power switches and isolation buffers should be to turn power *on* for all modules, and to allow all modules to communicate. A watchdog circuit should be implemented to reset to this state if both the master modules are unresponsive, as it could potentially mean that all system modules are powered down. It is required that the system return to the default state even if the backplane logic loses power. This can be a particularly difficult requirement to meet, but is important to ensure that the backplane itself does not become the single point of failure in the system.

3.2.5 Other Requirements

In addition to the functional requirements outlined above, certain requirements apply to the choice of components for the backplane. Firstly, the temperature range for all components, and thus for the whole system must be at least the industrial range of -40 to $+85\,^{\circ}\mathrm{C}$. Secondly, to keep the power consumption low, all backplane logic should run on $3.3\,\mathrm{V}$ instead of $5\,\mathrm{V}$, and logic families featuring low current consumption should be chosen. Lastly, availability of components is important for rapid prototyping, so components must be available and on stock from distributors such as Farnell or Digikey.

3.3 Specifications

Based on the requirements for the NUTS Backplane, a set specifications describing the different parts of the system can be designed. The backplane design has been partitioned into two parts: a power distribution module, and a logic module. The power distribution module controls the module power and provides protection and monitoring. The logic module implements the control structure needed for the master modules to control the state of the backplane, as well as the bus isolation buffers. The basic idea is to develop the circuits for one module, and then to copy these circuit units for the other modules.

3.3.1 Communication Bus

The choice of communication bus is a very important choice, and a few different suggestions have previously been mentioned as suggestions for the NUTS project. During this work we have decided to base the system on the Inter-Integrated Circuit bus, or I^2C bus. This is a very common communication bus used for low-bandwidth communication between ICs on a circuit board, using two bidirectional lines, one for clock (SCL) and one for data (SDA).

The choice fell on the I^2C bus mainly because it is multi-master capable, and for the fact that many electronic components can interface directly to the bus. Using I^2C for inter-satellite communication is also very common within the CubeSat community, and have been successfully used in other CubeSat missions. The bus implemented should be run at 400 kHz, thereby classifying as a *fast* I^2C bus.

I^2C bus isolation buffers will be placed to allow each module to be isolated, as described in the requirements section. In addition to allow bus segments to be disconnected, the buffers also reduce the capacitance on the bus lines, as the total bus is split into different buffered sections. This is important when a large number of components are connected to the bus at the same time.

Several improvements can be made to increase the reliability of the bus, such as implementing timers to prevent slaves from delaying the clock more than a certain amount of time. Higher network layers should also employ checksums to verify address and data bytes. These measures are important, but are beyond the scope of this work.

3.3.2 Power Distribution

Figure 3.1 shows the basic elements of the power distribution module that is repeated eight times on the backplane. It has the basic specifications:

- Combination of the redundant 3.3 V and 5 V supply buses with minimal losses.

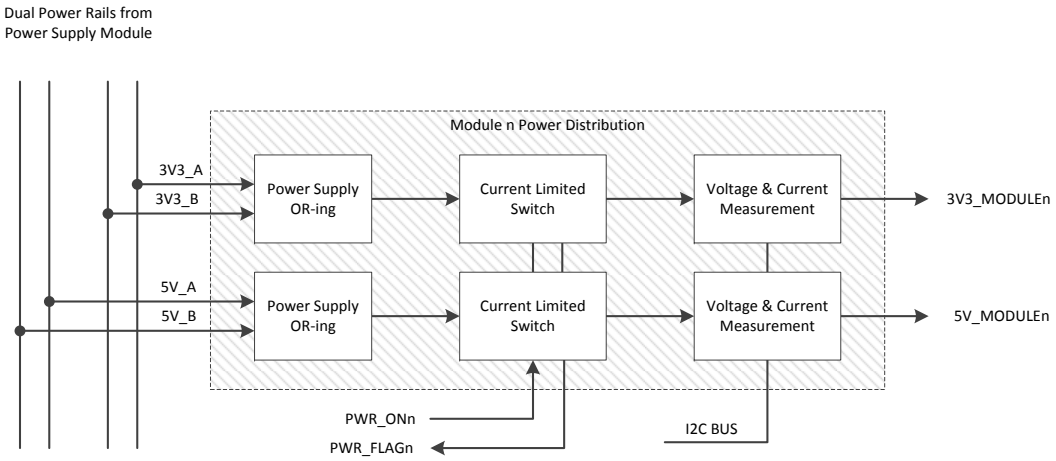


Figure 3.1: NUTS Backplane power distribution module.

- Adjustable 250 mA to 1.5 A current-limit power switches with auto-retry function.
- Voltage and current monitoring with data presented on I^2C bus.

The combination of redundant supplies is done with special OR-ing controllers acting as ideal diodes, to reduce the power loss. The current-limit switch implements both the short-circuit and SEL protection functions, as well as allowing the module to be powered down. The monitoring circuit uses a high-side shunt resistor with a current sense amplifier to measure current. The component includes an *Analog to Digital Converter* (ADC) and an I^2C interface.

Chapter 4

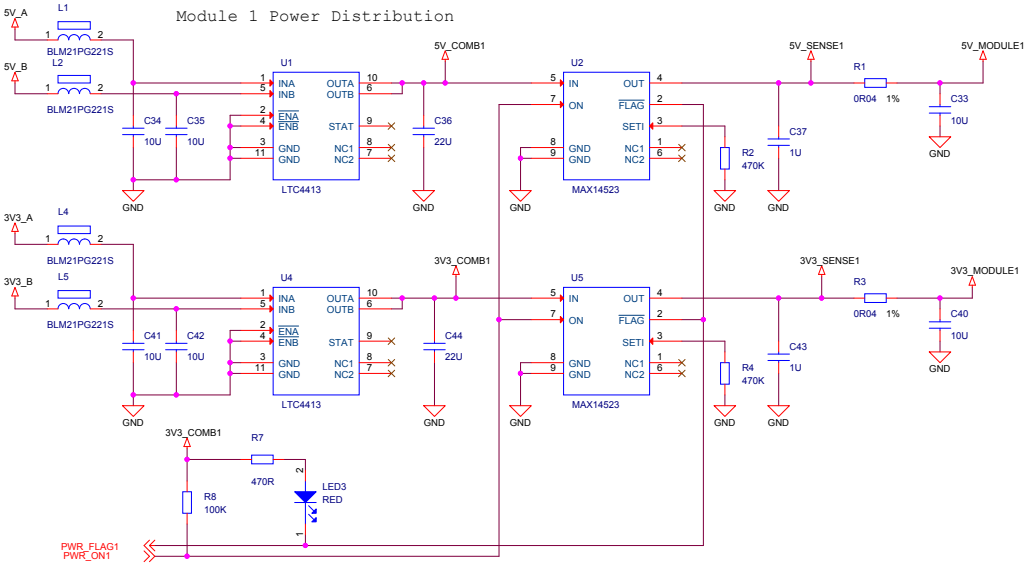
Backplane Implementation

As mentioned in the previous chapter, the design of the backplane functionality has been split into two parts: the logic and bus control, and the power distribution modules. Since the backplane itself consists of several module connectors and accompanying electronics that for the most part implement the same functions for each module slot, the circuit board becomes highly regular. This means that the functionality could be developed for one module slot first, and then copied the correct number of times, with some exceptions (the master and EPS modules). Here, the implementation of the backplane power distribution is first described, followed by the bus and control logic.

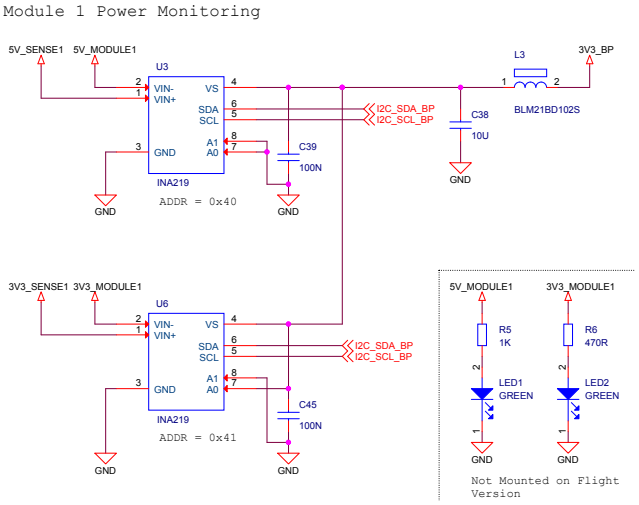
4.1 Backplane Power Distribution

The schematics for the power distribution circuitry for one module slot is shown in figure 4.1 (in this case Module 1, as it provides `3V3_MODULE1` and `5V_MODULE1`). Three main components are present for each of the 3.3 V and the 5 V power paths: an power supply or-ing controller (LTC4413), a current-limit switch (MAX14523) and a current monitor (INA219).

At the input to the module, a simple filter consisting of a ferrite bead and a 10 μF capacitor helps to reduce high frequency noise on the supply lines. The ferrite bead is a passive component which has a very small DC resistance, but a much higher impedance at high frequencies. The components selected for the backplane feature an impedance of 220 ohm at 100 MHz. These filters are in place both to block high frequency switching noise from the regulators, and to prevent different modules from disturbing each other by creating high frequency noise on the power supply lines.

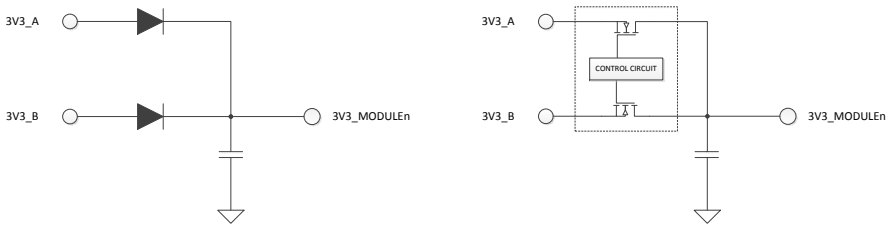


(a) OR-ing & Current-limit Switch



(b) Power Monitoring

Figure 4.1: Module Power Distribution



(a) Diode OR-ing (b) Transistor OR-ing

Figure 4.2: Power Supply OR-ing

4.1.1 Power Supply OR-ing

To allow the use of dual redundant supply buses, the voltage lines needs to be combined in such a way that current cannot flow from one supply regulator into the other, as this will cause the failure of one regulator to completely take down the system. This is a problem often encountered in high reliability, $N + 1$ redundant power supplies, used for example in telecom or other high-availability applications. ($N + 1$ meaning that if the load can be supplied with N units, $N + 1$ units are used to ensure continued operation if a unit should fail).

The simplest way to solve the problem is to use diodes to connect the two supplies, as shown in figure 4.2(a). The diodes allow both supplies to power the load, but prevents current from flowing back from an active supply into a failed supply.

The main problem with using diodes is the power lost due to the diode forward voltage, V_{fwd}. The power lost over each diode can be calculated as:

$$P_{loss} = V_{fwd} \cdot I_{load} \tag{4.1}$$

With the V_{fwd} varying slightly depending on the current load and temperature. Even with modern Schottky rectifier diodes this is still around 0.375 V at 500 mA (with power lost: $P = V \cdot I = 187.5\,\text{mW}$). In addition to the loss of efficiency, the forward voltage drop also means that the module is not supplied with the intended supply voltage.

To overcome the problems from using diodes in $N+1$ redundant power supplies, transistors can be used together with special OR-ing controllers, as shown in figure 4.2(b). The controlling circuit monitors the current through the transistor by measuring the voltage drop over the transistor on-resistance ($R_{ds(on)}$). As soon as the current becomes negative, the controlling circuit switches the transistor off to prevent reverse current from flowing, thereby emulating the behavior of a diode. The main advantage of using specialized OR-ing circuits is that the

voltage drop across the MOSFET $R_{ds(on)}$ is much lower than the forward voltage of a conventional diode, resulting in significantly lower power losses. Components implementing this functionality are often referred to as "ideal diodes".

The backplane implementation uses the LTC4413 components from Linear Technology Inc. Each LTC4413 contains an dual OR-ing controller and two integrated MOSFETs, and can handle up to 2.6 A of current at up to 5.5 V input voltage. These devices are designed to allow a load to be powered from two independent sources. They have a typical on-resistance of 140 mΩ, meaning that the voltage drop across the device is only around 70 mV at 500 mA (with power lost: $P = V \cdot I = 35\,\text{mW}$).

In figure 4.1(a), U1 and U4 are the LTC4413 devices for the 5 V and 3.3 V supplies, respectively. The enable inputs are permanently tied low, as power switching is done with a subsequent device. The `STAT` pins are unused, as this merely give information about which supply is powering the load. The LTC4413 is available in a very small 10-lead, 3 mm × 3 mm DFN package, and operates over the −40 °C to +85 °C industrial temperature range. [8]

4.1.2 Current-limit Switch

The current-limit switches implement two functions: module power switching and over-current protection. The switches consist of a MOSFET switching element together with a control circuit that monitors the output current.

In his report, Lars Opkvitne has previously evaluated and tested the use of power-distribution switches from Texas Instruments, the TPS230x. This device switches to constant current mode when the current draw exceeds a given threshold, and includes thermal protection to turn off the load under a prolonged over-current condition. The switch provides an enable input and a status pin indicating if the current limit is active. [17]

This component choice implements the basic protection features, but is somewhat unpractical for a few reasons. Firstly, the amount of current flowing during a prolonged over-current condition is equal to the current limit threshold, which can quickly drain the batteries if the switch is not explicitly turned off. This situation can for example arise if the module responsible for turning off power switches experiences a fault. Secondly, the thermal protection requires the component to heat up before shutting down the load. In a vacuum environment, there is no cooling due to convection, so removing the generated heat can be somewhat problematic.

To improve on these issues, the MAX14523A from Maxim have been chosen for the backplane. The switches feature a low 70 mΩ on resistance and an adjustable current limit between 250 mA and 1.5 A with an auto-retry feature. Like the TPS203x, the load current is limited by a constant current mode when the load current exceeds the programmed threshold. If the over-current condition persists

after a "blanking time" t_{BLANK} (typically 17.5 ms), the switch latches off for the "retry time" t_{RETRY} (typically 560 ms). At the end of t_{RETRY} the switch turns on again, and if fault still exists, the cycle repeats. If the fault has been removed, the switch stays on.

According to the datasheet [19], the ratio between t_{BLANK} and t_{RETRY} is fixed at 32, meaning that the average current during a prolonged over-current condition becomes:

$$I_{LOAD} = I_{LIM} \cdot \frac{t_{BLANK}}{(t_{BLANK} + t_{RETRY})} = \frac{I_{LIM}}{32} \tag{4.2}$$

This equals to about 97% lower average power compared to continous current limiting, and also reduces the chances of the switch heating up. These devices also provide SEL recovery in the NTUS application, as the faulty module is automatically power-cycled.

The current limit is programmed with an external resistor, and is shown in figure 4.1(a) as R2 and R4. The equation for calculating the resistor is given in [19]:

$$R_{SETI}\,(\mathrm{k\Omega}) = \frac{141400\,\mathrm{V}}{I_{LIM}\,(\mathrm{A})} - 2.48\,\mathrm{k\Omega} \tag{4.3}$$

The current limit for each module should be specifically tailored to fit the required peak current draw of the module. A default resistor of 470 kΩ is placed on the backplane, as limited information about module requirements is available. This should give a current limit of around 300 mA.

4.1.3 Power Monitoring

Power monitoring is implemented by measuring the voltage drop across a shunt resistor. High-side current measuring is preferred, as it does not introduce problems with shifted ground potential which occurs with a low-side current shunt. The voltage across the shunt is usually amplified using a differential amplifier, and converted to a digital value with an analog to digital converte (ADC).

The INA219 is a high-side current shunt and power monitor with an I^2C interface. It contains a programmable gain amplifier (PGA), an ADC, and a multiplier. A calibration value can be multiplied with the ADC result to allow measured current to be directly read out in amperes. An additional multiplier register contains the calculated power as the product of the measured bus voltage and shunt current. The PGA can be used to set the full-scale operating range of the current monitor, with the lowest possible full-scale shunt voltage as low as 40 mV. [14]

The INA219 supports up to 16 different I^2C addresses, allowing 16 separate devices to be used in a system at the same time. This is useful, since the NUTS backplane has eight module slots with two supply voltages each, equaling to 16

A1	A0	Slave Address
GND	GND	1000000
GND	V_{s+}	1000001
GND	SDA	1000010
GND	SCL	1000011
V_{S+}	GND	1000100
V_{S+}	V_{S+}	1000101
V_{S+}	SDA	1000110
V_{S+}	SCL	1000111
SDA	GND	1001000
SDA	V_{s+}	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	V_{s+}	1001101
SCL	SDA	1001110
SCL	SCL	1001111

Table 4.1: INA219 Address configurations

measurable power values. Table 4.1 shows the different address configurations that can be obtained with different configurations of the A1 and A0 pins. The power distribution module for slot 1, as shown in figure 4.1(b), use the first two addresses in table 4.1. A shunt resistor of 40 mΩ has been chosen to give the best resolution for a measured current below 1 A. This gives a voltage drop of:

$$V_{shunt} = I_{load} \cdot R_{shunt} = 40\,\text{mV@1 A} \tag{4.4}$$

Should a module exceed 1 A current draw, the PGA can be configured to measure across the wider range. A ferrite bead (L3) and bypassing capacitors (C38, C39 & C45) form a simple power filter, which is useful since these are sensitive analog components.

4.1.4 LED Indicators

LED indicators are included on each power distribution module to give an overview of the power state of the entire system during development and debugging. This includes power indicators for 3.3 V, 5 V (green LEDs) and a fault indicator on the status pin of the current-limit switch (red LED). This is deemed very useful during debugging, as the first step is always to check if the target has power. The fault indicator will specify whether the module has lost power due to the switches being turned off, or due to an over-current condition on the module.

These LEDs should **not** be mounted on the engineering and flight models, as the extra current drawn from lighting up 16 LEDs during normal operation will be a considerable contribution to the total system power consumption. Each LED uses around 10 mW of power, so 16 LEDs equals to around 0.16 W of wasted power.

4.1.5 Backplane Power Supply

Power for the backplane logic is provided by an almost identical power distribution module as for the other system sub-modules, providing the same protection to the backplane circuitry. The backplane supply voltages are called `3V3_BP` and `5V_BP`, and cannot be disabled by command like the module supplies.

Although the backplane cannot be powered down explicitly, it is still possible that a short-circuit in the backplane logic can cause the protection circuit to turn power off. The power switches of all the distribution modules have pull-ups to a local voltage, allowing them to remain on by default even if the backplane is powered down.

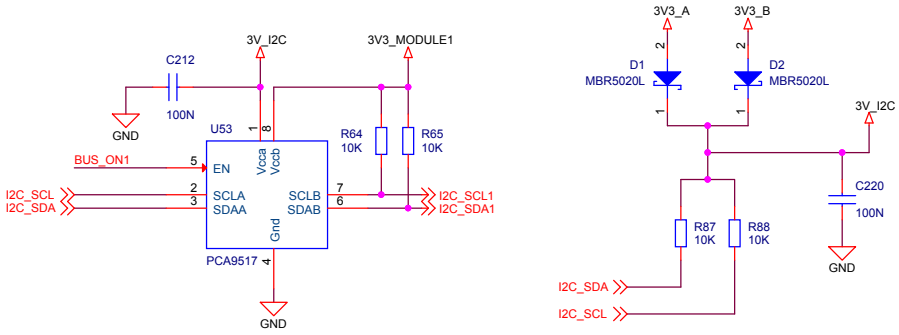
4.2 Communication Bus

The I^2C bus used for communication between system modules is implemented as several isolated segments in a star topology. Each module has it's own bus segment that can be disconnected from the rest of the system. The number of I^2C devices on each module card is limited only by a maximum bus capacitance of 400 pF, and possibly by the total number of available addresses (with standard 7-bit addressing: $2^7 = 127$ addresses).

4.2.1 Bus Isolation

Bus isolation is implemented with I^2C repeater circuits, which are essentially bidirectional level-shifting buffers. The PCA9517 from NXP is such an I^2C bus repeater which allows the two different sides (the *A-side* and *B-side*) to operate at different voltages, and also includes an *enable* input to disable the buffers.

According to the data sheet, the A-sides of several devices can be connected together to form a star configuration. As the different system modules are powered through separate power distribution switches, the bus voltage of the different modules may vary slightly between modules. Using level-shifting I^2C repeaters ensure reliable and safe bus operation despite voltage differences.



(a) Module 1 Bus Repeater (b) 3V_I2C supply

Figure 4.3: Backplane I^{2}C bus repeaters

4.2.2 Pull-ups

As the I^{2}C bus itself is based on open-drain connections to two common lines, pull-ups are required to pull the lines to a defined voltage when released. The size of the pull-up resistors is usually a trade-off between power consumption and bus capacitance together with desired bus speed. When a bus repeater is used, pull-ups are required on both sides of the repeater. This means that the bus lines for each module are pulled up to that module's supply voltage.

A separate supply voltage, `3V_I2C`, is used on the backplane for the A-sides of the bus repeaters and for the pull-ups on the I^{2}C lines running through the backplane. This supply is provided through Schottky diodes from the redundant `3V3_A` and `3V3_B` rails, as shown in figure 4.3(b). The reason for using this instead of the `3V3_BP` supply voltage used by the rest of the backplane logic, is the desire to ensure continued communication even if the backplane logic loses power. This will prevent a short circuit on the backplane from disrupting the communication bus and becoming a single point of failure.

Figure 4.3(a) shows the I^{2}C repeater for the module 1 slot. The B-side of the PCA9517 is powered by the module supply voltage, `3V3_MODULE1`, and pull-ups to this voltage are present on the `I2C_SCL1` and `I2C_SDA1` lines. On the A-side, the `I2C_SCL` and `I2C_SDA` run throughout the backplane and are connected to the A-sides of all the other repeaters. These lines are pulled up to the `3V_I2C` supply as shown in figure 4.3(a).

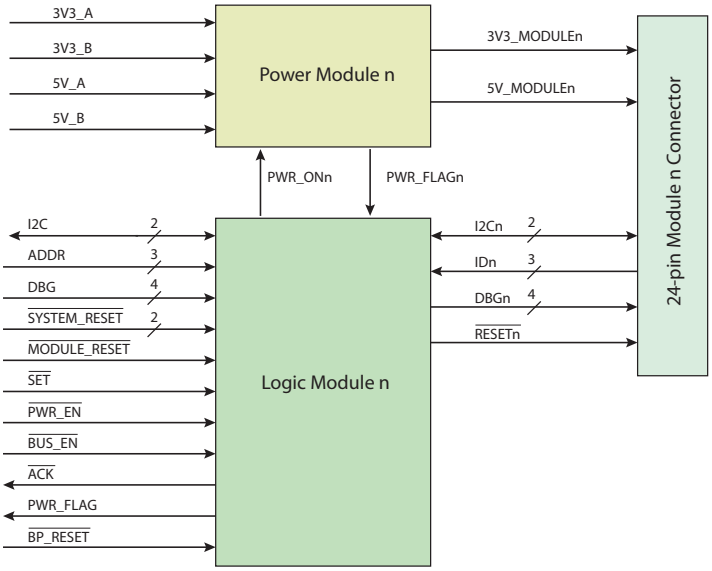


Figure 4.4: Backplane Logic Unit

4.3 Backplane Bus Logic

To allow the two master modules to control the state of module power and bus isolation switches, a simple logic circuit is implemented at each slot. The basic principle is to use flip-flops as registers to remember the state of each power and each bus isolation switch. The switches associated with each module are addressed from the master modules by using a simple 3-bit address matching scheme. The addressing is not fixed per module position, but requires the modules to provide 3 `ID` signals, giving a unique module ID that is matched with the 3 `ADDR` lines. If the module is present, the `ACK#`[1] signal is pulled low to indicate that the module has been selected. The master can then assert the `PWR_EN#` and/or the `BUS_EN#` lines, and store the result by pulling `SET#` low.

When the module is selected, 4 JTAG/SWD lines are connected through a bidirectional buffer, allowing direct debug access to the selected module. Modules can be reset with two main mechanisms:

- *Module Reset* allows each module to be individually reset by first selecting it and then pulling `MODULE_RESET#` low.
- *System Reset* resets all system module except the master module initiating the reset.

[1]Throughout the design the hash (#) postfix is used to indicate active-low signals.

Finally, the PWR_FLAG# signal, which indicates whether a over-current condition is present or not, is routed through a tristate buffer which becomes active when the module is selected. An overview of the different signals and their relationships are shown in a block diagram in figure 4.4.

4.3.1 Addressing

Addressing of modules is done with the simple circuit shown in the top part of figure 4.5. Three XOR circuits and a three-input OR-gate realize the logic function:

$$\overline{\text{SELECTn}} = \overline{(\text{ADDR0} = \text{IDn0}) \cdot (\text{ADDR1} = \text{IDn1}) \cdot (\text{ADDR2} = \text{IDn2})}$$

SELECT#n is a local active-low select signal which is used by the rest of the logic unit to only allow operations when the module has been selected. The logic equation has been simplified with De Morgan's law to produce the circuit in figure 4.5.

4.3.2 State Flip-flops

The circuit block used for remembering the state of power and bus switches is shown in the top part of figure 4.5. The data inputs of the 74LVC74 dual-flip flop IC are driven directly by the BUS_EN# and PWR_EN# lines from the master module. The SET# signal is gated with the SELECT#n signal coming from the address matching circuit, and fed into the clock inputs. This ensures that the state is only latched when the module has been selected, and there is a high to low transition on the SET# signal (positive-edge triggered flip-flops). The BUS_ONn signal is passed to the I^2C repeater (figure 4.3(a), and the PWR_ONn signal is connected to the power distribution unit for slot n.

A reset signal, BP_RESET#, is provided to reset all flip-flops to their default state, which is to turn on all power and bus repeaters. This allows a watchdog circuit (detailed below) to reset the backplane to a default state should both master modules become unresponsive.

4.3.3 Debug & Reset

The DEBUG signals are switched in to one module at a time by the use of a quad bilateral switch (74LV4066). The signals are labeled TCK, TDI, TDO and TMS according to the JTAG standard, but can also be used to debug targets which only support serial wire debugging (SWD)[2]. This allows most common microcontrollers to be used in modules, as most modern devices support either JTAG or SWD.

[2]SWD is common in modern ARM microcontrollers, like the new Cortex-M3

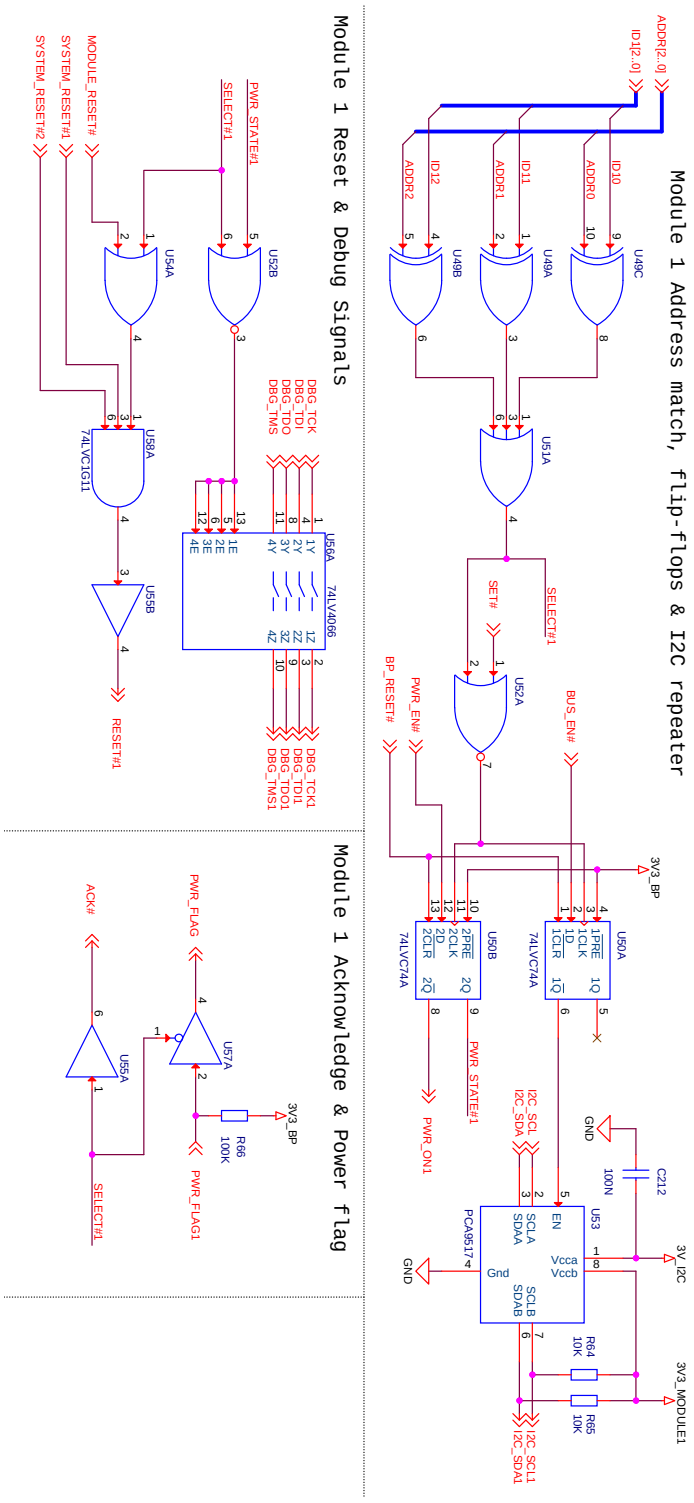


Figure 4.5: Backplane Logic Circuit for Module 1

The SELECT#n signal is gated with the PWR_STATE#n to prevent the debug lines from being connected when the module power is disabled, as this can cause the devices to be powered through the debug lines and can cause damage to the module electronics.

To generate the module reset signal, the MODULE_RESET# signal is gated with SELECT#n and combined with two SYSTEM_RESET# signals. The two SYSTEM_RESET# signals come from the two separate master modules in the system. The generated reset signal is passed through an open-drain buffer (U55B in figure 4.5), to prevent the logic circuit from driving the reset signal high, as this can cause problems if other circuits are also connected to the same line (for example local watchdog circuits). This also means that each module has to employ a local pull-up resistor on it's reset line.

4.3.4 Acknowledge & Power Flag

The ACK# signal is fed back to the masters to indicate that a module is present, and the POWER_FLAG signal indicates an over-current condition from the power distribution module. Since these output signals from all 8 modules are connected together, it is important they are either open-drain, or high-impedance when the module is not selected. The circuit is shown in the lower right of figure 4.5. The PWR_FLAGn signal coming from the power distribution module is an open-drain signal, so a local pull-up is added.

4.4 EPS & Master Module Slots

Most of the module slots on the backplane are identical, allowing the same circuit to be copied a number of times. This regularity greatly eases the time required to place and route all components. There are however necessary exceptions to this regularity, and this comes into play with the EPS module (which supplies power to everything else) and the two master modules.

4.4.1 EPS Module Slot Exceptions

The EPS is a slave module in the same way as the rest of the slave modules in the system, but does not require the same power distribution circuit. The backplane logic is therefore slightly simplified:

- The dual flip-flop is replaced with a single flip-flop (74LVC1G74) which only controls the bus repeater.
- The POWER_FLAG# signal and corresponding tristate buffer is removed.

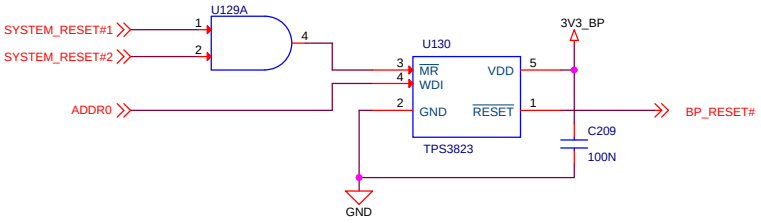


Figure 4.6: Backplane Reset Monitor

Also, since the EPS does not require a power distribution circuit, the PCB area that would normally be used to implement the module's power distribution circuit is now used for the backplane power distribution.

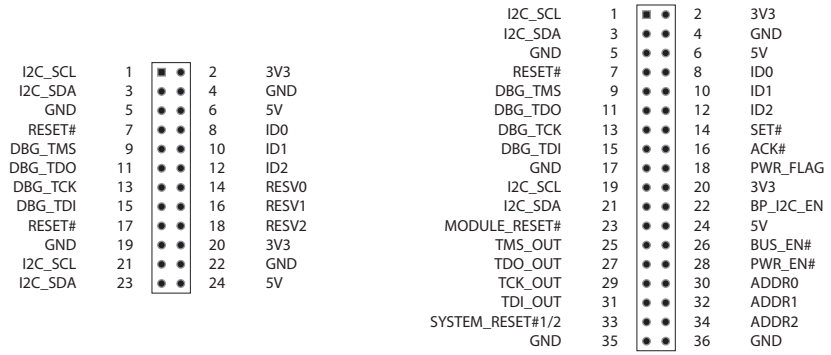
4.4.2 Master Module Slot Exceptions

To give both master modules the power to monitor and control each other, the master modules employ almost exactly the same logic and power distribution circuits. The main difference from other system modules lie in the connectors, which in addition to the normal signals also has to accommodate the signals necessary for controlling the backplane.

Apart from the different connector, the only difference in the logic lies in the *reset* circuitry. The first master supplies the `SYSTEM_RESET#1` signal, and the second master supplies the `SYSTEM_RESET#2` signal, while both masters can pull the `MODULE_RESET#` low. The `SYSTEM_RESET#1` signal resets all modules *except* the first master module, and `SYSTEM_RESET#2` resets all modules *except* the second master module. This prevents a master module from resetting itself when trying to reset the whole system, as this will likely not result in a proper reset pulse. The change in reset circuitry is implemented by exchanging the 3-input AND gate (U58A in figure 4.5) with a 2-input AND.

4.5 Backplane Reset Monitor

A reset monitor circuit with a built-in watchdog timer is used on the backplane to prevent accidental disabling of all modules. Should this happen, the watchdog timer will time out, and reset all bus and power switches to the default ON-state. The circuit is shown in figure 4.6. The circuit also ensures that all modules are reset to the default state when the backplane is powered up. Both `SYSTEM_RESET#` signals are connected to the reset monitor, so that a system reset also resets the backplane state (a module reset will not).



(a) Slave Connector Pinout

(b) Master Connector Pinout

Figure 4.7: Connector Pinouts

There is no dedicated watchdog triggering signal, as it was considered that the pin count on the master modules was already too high. Instead, one of the address signals are used, requiring periodic changes in module addressing as a way to notify the backplane that one or more master modules are active.

Note that a watchdog reset will only reset the state of the backplane power and bus switches, this will NOT generate a system reset to the connected modules.

4.6 Connectors

The connectors used to interface modules to the backplane are standard 2-row, 2.54 mm pitch, right angle pin headers on the modules, and matching sockets on the backplane. The slave connectors have 24 pins each, and the master connectors 36 pin.

Several high priority signals have two or more redundant pins, but this is limited to power, reset and I^2C bus signals. The redundant pins are connected together on the backplane to help ensured operation in the event of a minor mechanical failure, and the intention is that these signals are also connected together on the module card.

The slave connectors have three extra pins, labeled RESV0 to RESV2, and are connected together for all slave modules. These signals are not specifically used for any backplane-related functions, and are reserved for use by the modules. Unfortunately the master module connectors do not include these signals, as the pin count was already very high (36). This limits somewhat the possible usefulness of the reserved signals.

4.7 Schematic Design & PCB Layout

The actual schematic design was drawn up in Cadence OrCAD Capture, and the PCB layout was done in the accompanying program Cadence Allegro PCB-editor. The backplane schematic spans eleven A3 sheets, including a title page. The final bill of materials (BOM) counts a total of 553 components, with 375 of these being passives (Resistors, Capacitors, LEDs & Ferrite Beads).

Each logic module has been implemented using standard logic devices, with as few as possible gates unused, to save space. In many cases, so-called *little logic* has been used, which is basically only one or two logic gates in an IC package, instead of the standard four or six gates. The logic families used has been carefully chosen based on a number of criteria:

- The ability to work with a 3.3 V power supply.
- Low static and switching power consumption.
- Must be available in industrial operating temperature range (-40 to $+85\,^{\circ}\mathrm{C}$).
- Should allow system partial power down.

The last criterion requires that the current leakage into an input or output pin is limited when the device is not powered. Traditionally, ESD protection diodes form a path from the output pin to ground in CMOS devices when the device power is removed. This will cause a high logic level on the pin to cause excess current leakage and perhaps even damage to the device. To allow parts of the system (modules) to be safely powered down, this leakage current must be limited. In logic data sheets, the parameter I_{off} is used to specify the maximum leakage current when $V_{CC} = 0\,\mathrm{V}$.

A small ceramic bypass capacitor (100 nF) is placed between GND and VCC of each logic component for proper decoupling.

4.7.1 PCB Layout

The PCB design was done with a 4-layer circuit board, with two layers (TOP and BOTTOM) for signal routing, one internal layer for a ground plane spanning the entire board, and one internal layer for a power plane (`3V3_BP`). The ground plane has been kept completely free from other traces to have the best possible coverage, and the power layer contains only a few traces. Almost all the components has been placed on the top layer, with only a few passive components on the bottom. Most of the traces on the top layer follow a horizontal path, while most of the traces on the bottom layer follow a vertical path. This allows the top layer to contain

the internal connections for each logic and power module, while the bottom layer is used to carry the signals that run throughout the entire backplane.

The module connectors have been laid out 19 mm apart, with around 40 mm free space in the center of the backplane to make room for the batteries. The logic and power distribution electronics fit neatly above each connector, and the regularity of the circuit board allowed whole sections to be copied and pasted at an appropriate offset. Even when copying entire sections, the layout of the PCB still required a considerable amount of time: nearly 60 hours have been spent to complete the PCB layout.

4.8 Production

The prototype production has been outsourced to a local company in Trondheim, NOCA AS. The design files, including the necessary Gerber files, BOM and assembly drawings were handed over, and NOCA handled the PCB production, component acquisition and soldering of the final board.

Figure 4.8 shows an image of the final produced NUTS Backplane, with some labels to indicate the position of components. (The two connectors without labels are also slave connectors).

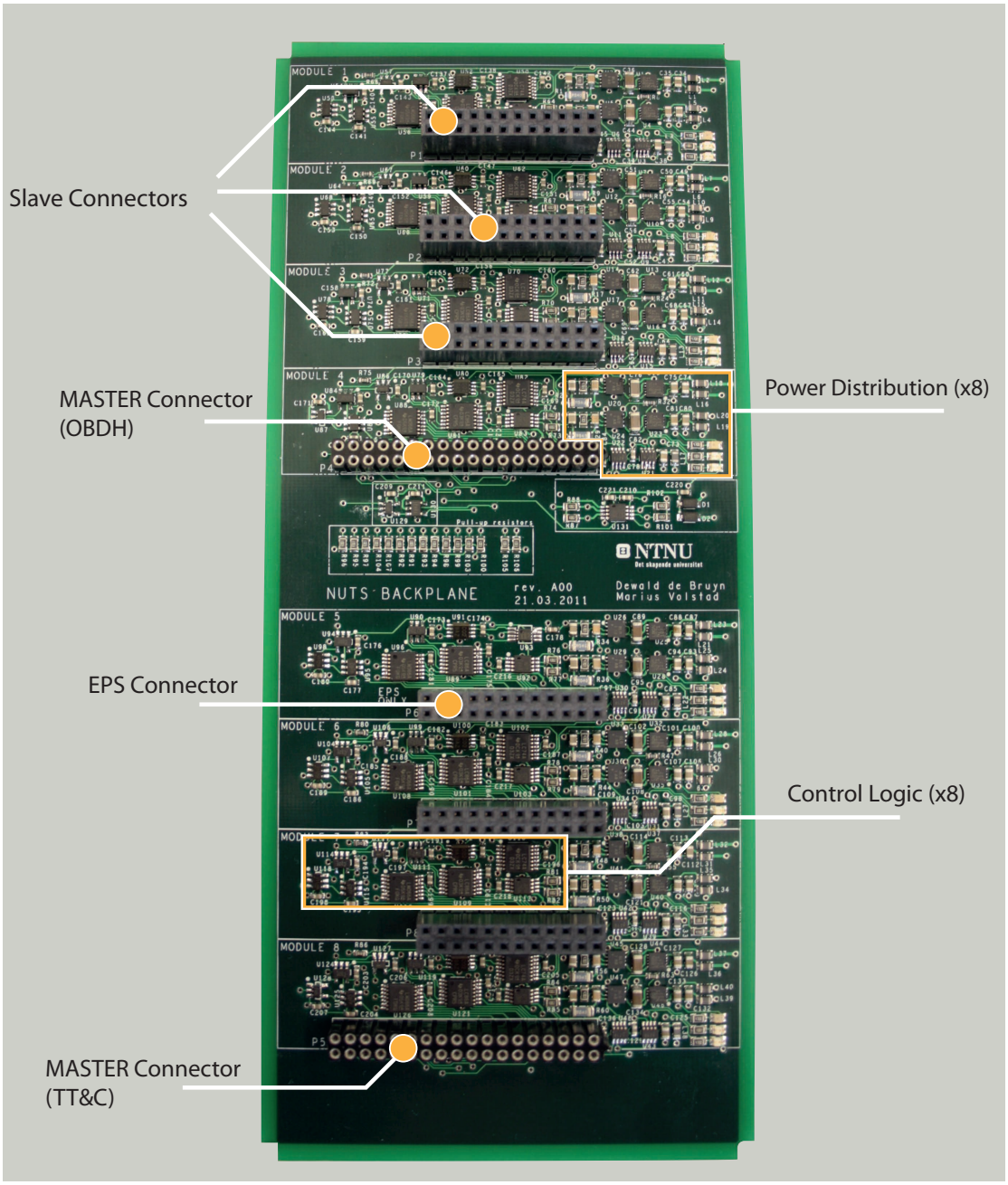


Figure 4.8: Final Backplane Board

Chapter 5

Backplane Testing & Verfication

An experimental prototype of the backplane power distribution unit was constructed before the design was incorporated into the backplane. This chapter presents the test setup used to verify the functionality of the circuit. An initial test plan with some simple hardware tests for the finished backplane is also outlined.

5.1 Power Distribution Prototype

During the early design phase of the backplane, it was decided to build a small prototype of one of the eight power distribution units, to verify that the different components work well together. Designing and producing a small prototype also helped estimate the size of the implementation, to help determine the feasibility of the solution.

The resulting prototype is a tiny 2-layer circuit board measuring only 26 mm × 19 mm. The PCB was manufactured using a milling machine at the workshop of the Institute of Technical Cybernetics, and the components were soldered by hand.

5.1.1 Test Setup

A test setup which allows the prototype to be tested over a wide load range was created. The test setup consists of an EFM32 microcontroller, a programmable current sink and the prototype PCB. The setup uses an EFM32 Gecko Development Kit (Gecko DK), from Energy Micro AS. The kit features a prototyping expansion card which plugs into the main board, and can be used to construct simple prototypes that work together with the EFM32. The programmable load and the power distribution module prototype were placed on such an expansion card and plugged into the Gecko DK. The EFM32 communicates with a PC running a

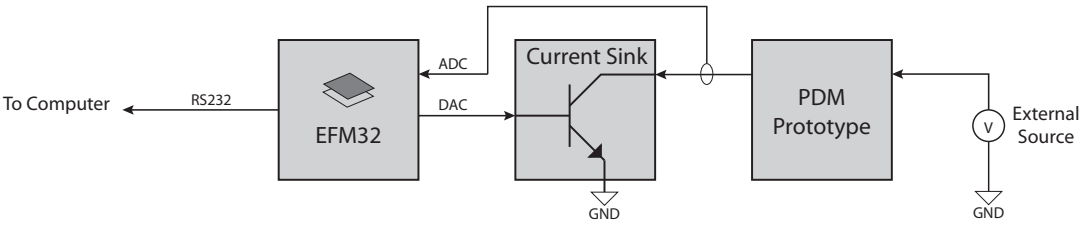


Figure 5.1: Power Distribution Module Test Setup

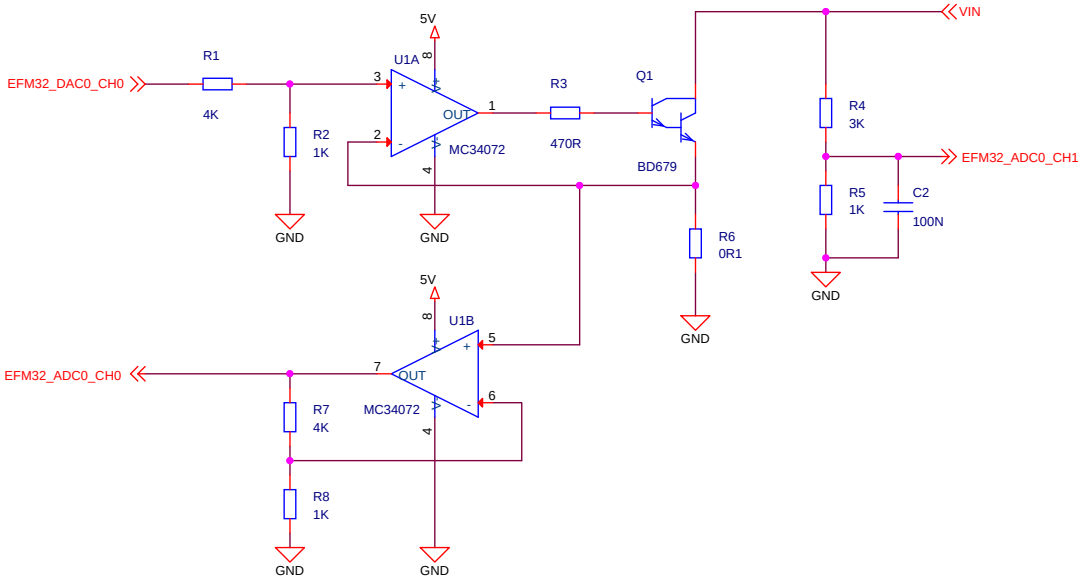


Figure 5.2: Programmable Load Circuit

LabView program to create stimuli and log the response of the circuit under test.

5.1.2 Programmable Load

The circuit used to create a programmable load is shown in figure 5.2. The amplifier U1A turns on the Darlington transistor Q1 so that the current through the shunt resistor R6 becomes equal to the reference voltage, which is provided by the DAC and a voltage divider (R1 & R2). Assuming an ideal operational amplifier gives the current I_{LOAD} through R6:

$$I_{LOAD} = \frac{V_{Iref}}{R6} = \frac{1}{R6}\left(V_{DAC}\frac{R2}{R1+R2}\right) = 10 \cdot 0.2 \cdot V_{DAC}\,(\mathrm{A}) \qquad (5.1)$$

Which means that the magnitude of the load current through R6 is equal to twice the DAC voltage. When using the internal reference of 1.25 V for the DAC, this

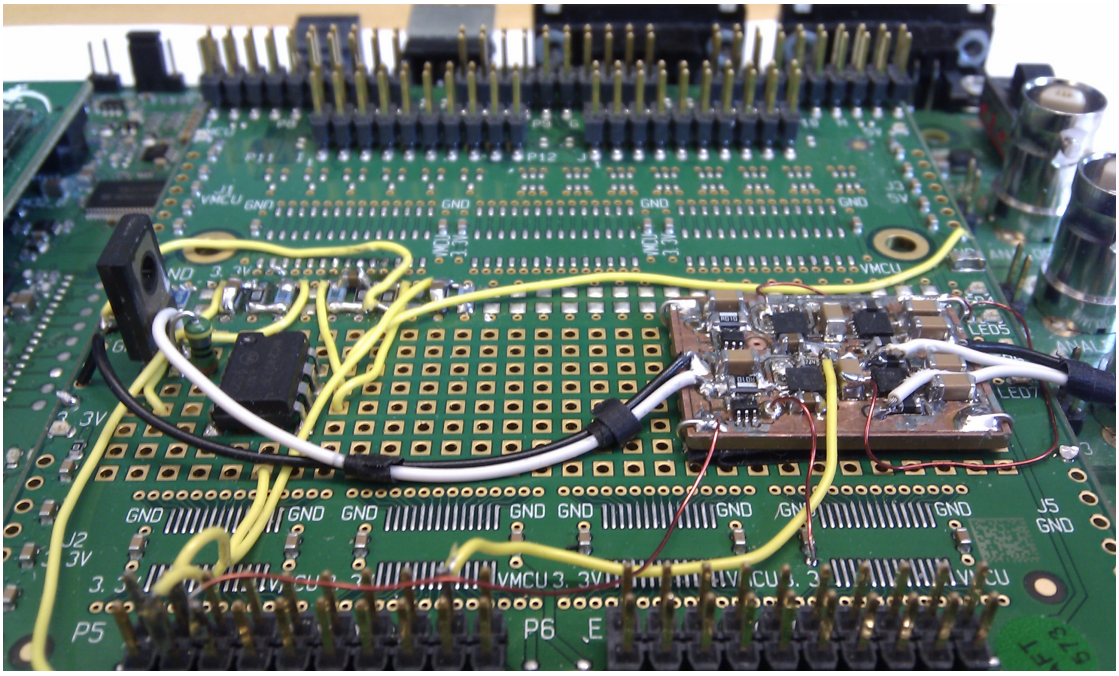


Figure 5.3: Test setup on the Gecko DK

gives a $0 - 2.5\,\mathrm{A}$ current range.

The other op-amp in the package is used in a non-inverting amplifier with a gain of $(1 + R7/R8) = 5$, to amplify the few millivolts over R6 to a useful range for the EFM32's analog to digital converter (ADC). This approach to measuring the load current was later discarded in favor of the built-in current measurement of the power distribution prototype, which is much more accurate.

Finally, R4 and R5 create a voltage divider so that the input voltage can be measured with the ADC.

Figure 5.3 shows the test setup on the Gecko DK expansion card. The current sink is to the left of the card, and the power distribution unit circuit board to the right.

5.1.3 Software

To allow LabView to control the test setup with ease, a small serial terminal is implemented in the EFM32. The terminal accepts text commands over an RS232 cable from the PC. Two main commands are issued by the LabView program:

1. `read <addr>`
2. `write <addr> <data>`

The `read` command returns the contents of the specified memory address, and the `write` command attempts to write `data` into the specified memory address. Since the all the IO in the EFM32 are memory-mapped, this allows the LabView program to directly write into the peripheral registers through the terminal interface.

In addition to handling `read` and `write` commands, the software also sets up the ADC to continuously sample, and the DMA is used to transfer the ADC results to a known memory location (which is read using `read`). This creates a powerful setup which allows most of the processing to be done in LabView, and the microcontroller simply becomes an IO-extension over RS232.

Several experimental LabView programs were written. For example, one program contains sliders and buttons to control the test setup, and plots the measured current and voltage values in a graph. Another test program was created which sets up a $0 - 1\,\mathrm{A}$ linear current ramp, and stores the sampled voltage and current values. The LabView program components are described in more detail in appendix C.1.

5.2 Backplane Initial Hardware Testing

A few basic tests were planned for when the backplane came back from production:

1. Visual inspection.
2. Initial power-up.
3. Voltage measurements.
4. Check response to an over-current condition.
5. Test communication through the I^2C bus repeaters.
6. Logic Tests.

5.2.1 Visual Inspection

It is always a good idea to perform an extensive visual inspection of the circuit board before power is applied to any part of it. This will help identify components that might possibly have been mounted with the wrong orientation. This can always happen due to human error such as misinterpretation of the assembly drawings. If this is indeed the case, applying power to the circuit board will likely cause damage, and it can be difficult to determine exactly which components have been damaged afterwards. Another thing to look out for is unwanted short circuits due to improper soldering.

5.2.2 Initial Power-up

The initial power-up of a new electronic prototype should always be done carefully in controlled conditions. A laboratory power supply with adjustable current limiting and current and voltage indication is required. Limiting the current from the power supply will limit any damage that might occur to components should there be an unforeseen design error that causes a short circuit.

Procedure: For the backplane, start with the power supply voltage at 0 V and connect it between a GND pin and either the 3V3_A or the 3V3_B pins of the EPS connector. Set the current limit to a low value (100 mA should work). Slowly ramp up the supply voltage to 3.3 V while continuously observing the current draw.

Repeat with the 5 V bus. If no direct power supply errors exist, all the green indicator LEDs should light up when both 5 V and 3.3 V is present.

5.2.3 Voltage Measurements

Both supply voltages at each module slot should be measured. This will give an indication of the low-load voltage drop across the power distribution units. All the 3.3 V voltages should be above 3.25 V, and all 5 V points should be above 4.95 V (less than 50 mV drop).

5.2.4 Over-Current Response

Each module supply voltage should be short circuited to test that the over-current protection circuits are working properly. This can be done by using a multimeter in a current measuring configuration, and connecting the meter across the supply voltages. In each case, the power supply to the other connectors should remain unaffected, while the red LED corresponding to the affected module should light up.

This should also be repeated with the backplane power supply `3V3_BP`.

5.2.5 I^2C communication

Two test devices should be connected to the module connectors such that they have two I^2C bus repeaters between them. It should be tested that the devices can communicate with each other at 400 KHz without problems. An oscilloscope should be used to verify the waveforms. This will verify the I^2C bus repeater configuration, together with the backplane I^2C supply voltage and pull-ups.

5.2.6 Logic Tests

Extensive tests should be performed to verify the backplane logic configuration. Preferably a test spanning the entire state-space of the backplane should be performed, but this can be tedious. The backplane reset monitor and watchdog circuit should also be verified for it's functionality.

Chapter 6

Backplane Results

The results obtained from the testing of the backplane power distribution unit prototype are first presented here. The initial hardware tests performed on the backplane prototype is also presented.

6.1 Power Distribution Prototype

Figure 6.1 shows the output voltage and current of the power distribution unit prototype under a $0 - 1\,\mathrm{A}$ current ramp test. The test was set up to slowly ramp up the current over a time of 4 seconds. The current limit programming resistor was set to $220\,\mathrm{k\Omega}$, which equals to a current limit of $635\,\mathrm{mA}$ according to equation 4.3.

As can be seen in the figure, the voltage decreases slightly with a small slope until the current reaches about $400\,\mathrm{mA}$, then starts to drop more rapidly as the load current comes close to the current-limit set-point. As soon as the load current reaches the set-point, the device enters the auto-retry mode.

Once in auto-retry mode, short current spikes can be seen about every 0.5 seconds. The duration is only a few milliseconds, and the amplitude is always limited to the current-limit value ($635\,\mathrm{mA}$).

It might seem that the output voltage of the power distribution does not return completely to $0\,\mathrm{V}$ during the retry time, but this is due to the measurement setup. Consider the circuit in figure 5.2: if the voltage at VIN is $0\,\mathrm{V}$, the op-amp still sources current through the base-emitter junction of the transistor, causing a small voltage (about $0.5\,\mathrm{V}$) to remain across R6 and Q1.

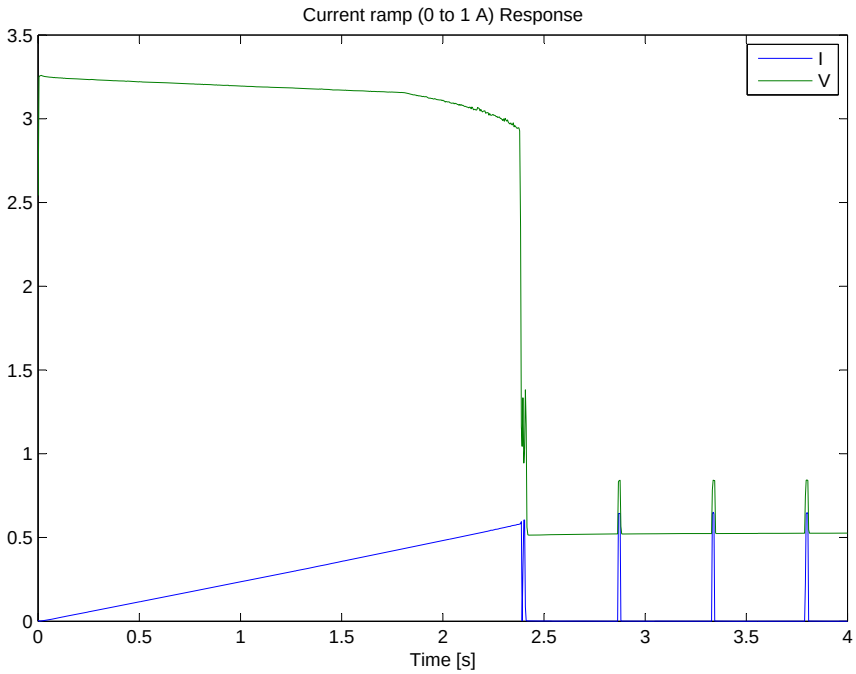


Figure 6.1: Response to a linear current ramp over 4 seconds

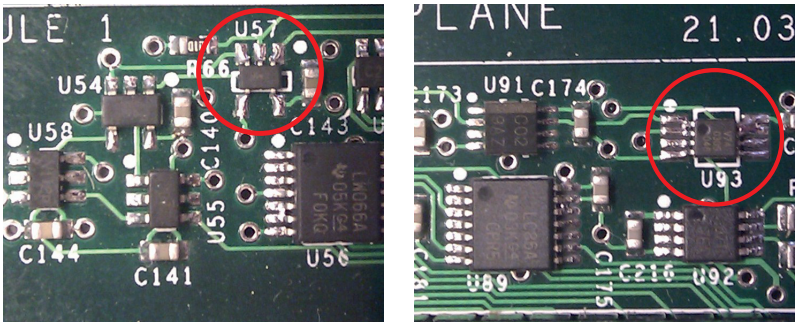


Figure 6.2: Components in wrong packages

6.2 Backplane Hardware Tests

6.2.1 Visual Inspection

Visual inspection revealed no accounts of components placed with the wrong orientation, nor of components in the wrong place. Also, no short-circuits were discovered that was not already there in the PCB traces (adjacent pads connected together).

However, several of the mounted components had the wrong package version. These package versions were smaller than the specified components' packages, and do not entirely fit on in the footprint on the PCB. This can clearly be seen in figure 6.2. In the left image, U57 should have the same footprint as U54 (SOT-23 package), but is clearly smaller (SC-70 package) and does not quite fit correctly. The same goes for U93, which should have the same package as U92 (TSSOP8 package), but is significantly smaller (VSSOP8 package).

Additionally, none of the connectors on the backplane are exactly as specified. The slave connectors are all 3 mm higher than specified, and the master connectors are of a completely different type.

6.2.2 Initial Power-up

Initial power-up proceeded smoothly without any problems. All green indicator LEDs light up when both 3.3 V and 5 V is applied.

6.2.3 Voltage Measurements

With 5.035 V applied to the 5V_A rail, and 3.302 V applied to the 3V3_A rail, the resulting module voltages are given in table 6.1.

Module	3.3 V	5 V
Module 1	3.265	4.999
Module 2	3.265	5.000
Module 3	3.261	4.999
Module 4	3.265	5.000
Backplane	3.259	5.001
Module 6	3.267	5.000
Module 7	3.268	5.000
Module 8	3.264	4.997

Table 6.1: Measured bus voltages

6.2.4 Over-Current Response

When short-circuiting a module supply voltage, the backplane behaves as expected. The green power indicator LED corresponding to the short-circuited rail turns off, and the red POWER_FLAG LED turns on. The other modules' supplies remain unaffected.

However, when attempting to short-circuit the backplane voltage, `3V3_BP`, the backplane over-current protection engages, but all the other modules' power switches are turned *off*. This is not the intended response by design, as it is required that system modules remain powered even if the backplane itself loses power.

Investigation into this issue revealed that the flip-flops (74LVC74A), used to control the power switches, do not feature partial power-down operation (I_{OFF} not supported). This causes current to leak through the pull-ups that are supposed to keep the switches on, lowering the logic level and turning them off. Fortunately, this could be fixed by a small BOM change, by changing the 74LVC74A devices to 74LV74A.

To test this, one of the 74LVC74A devices was replaced with a 74LV74A. The the module corresponding to the replaced device did not lose power when the test was repeated.

6.2.5 I^2C communication

The I^2C communication did not work as expected at first. It appeared that the bus repeaters became stuck at a low level after a high-to-low clock transition. To try to remedy this, the pull-up resistors on the backplane I^2C lines were changed from 10 KΩ to 2.7 KΩ. Apparently this stopped the repeaters from getting stuck, but there were still some anomalies on the clock line just after a low-to-high transition. Decreasing the ohmic value of the pull-up resistors on the module-side of the

repeaters fixed this problem, and allowed I^2C communication at 400 KHz.

6.2.6 Logic Tests

Some strange behavior was detected from the reset circuit, but it was later discovered that a missing pull-up resistor on the `BP_RESET` line could be the cause.

It was also discovered that a design error caused the `ADDR[2..0]` lines to be pulled to ground instead of being pulled up to 3.3 V. This breaks with the open-drain interface philosophy, and was fixed through a small wire-mod, turning the pull-down resistors into pull-ups istead.

A brief list of functions that have been verified:

- Module Addressing: Connected some of the `ID` lines of a module slot to GND, and verified that `ACK#` was received when this address was matched by doing the same with the `ADDR` lines.
- JTAG programming: Verified that debug signals pass through to the addressed module. Tested programming an AVR32 device with a JTAGICE mkII probe through the backplane.
- Power/bus enable/disable: Verified that module power and bus repeaters can be turned on or off.
- RESET: Verified that a *system reset* resets all modules except the module responsible for the reset signal. Verified that individual modules can be reset.

Chapter 7

Backplane Discussion

7.1 Design Evaluation

The design for the NUTS Backplane done in this project provides a good starting point for designing and constructing the small student satellite which will one day become NUTS-1. The backplane is designed to be robust and simple to use, but also provides some advanced safety features.

7.1.1 Backplane vs. Stacking Approach

The NUTS design distinguishes itself from many other CubeSat designs by building the system on a backplane instead of stacking cards on top of each other. This is believed to have some advantages. One of the main reasons to use a backplane lies in the ease of development. When a module in the middle of a multi-board stack has to be removed or debugged, the whole stack has to be taken apart, while with a backplane design the interesting module can be removed much easier. This can be very useful if it is suspected that a certain module is causing problems for the rest of the system.

The connectors used in the backplane design can be smaller than in a stacked design, since only the necessary signals to each module needs to be present in the connector. In a stacked design, each connector has to pass through all the signals to all the modules, even those that are not directly useful for the module in the middle. Implementing power supply switching thus becomes more complicated: the electronics that have been integrated into the NUTS Backplane would otherwise have to be implemented on a separate module card, and the separate power buses for each module would have to pass through all modules' connectors. This will indeed make the implementation of the safety features found in the NUTS system very difficult.

A stacked design does have some advantages in terms of mechanical stability though. The stacked design is stable by itself, where a backplane approach requires extra stability in form of fixing module cards to the satellite frame. Furthermore, choosing a stacked design that conforms to the PC104 standards found throughout the CubeSat community allows commercial components to be integrated into the satellite. The choice of basing the satellite on a backplane will in many ways prevent commercial modules from being considered, as they would not fit.

The advantages provided by being able to integrate much of the bus and power control logic into the backplane itself is considered to outweigh the disadvantages of using a backplane approach. The backplane approach has been around since the days of the NCUBE satellites, and the University of Oslo's CubeStar project is also based around a backplane approach, with many similarities to the NUTS design.

7.1.2 Power System Architecture

The NUTS backplane is based around using centralized power conversion and regulation. This differs from traditional satellite approaches, where all the electronics is supplied with a relatively high voltage bus, and the modules have local voltage regulation. Dual redundant power supply buses are implemented to prevent the entire system from failing should a voltage regulator fail.

The combination of these dual supply rails, in such a way that failure of one regulator does not affect the other, adds significant complexity to power distribution. In many ways, using simple Schottky diodes would have been safer/more reliable, but this comes with a much higher energy cost. Since power is the most precious commodity in a nanosatellite, the presented solution is considered a good trade-off.

The power switches present on the backplane are essential to allow the system to gracefully power down in case of low battery voltage. This feature is highly recommended in CubeSats. However, the backplane only provides the *possibility* to do this. Good system and software design is necessary to make use of these features. It is important that future students working on the NUTS project understand that the intention is to leave sub-systems powered, and to only use the power switches in anomalous situations. It is preferred to command sub-systems to enter a low-power standby state to save instead of powering them off.

The power supply protection circuits implemented have been chosen to handle an over-current situation as efficiently as possible. The average continued power during an over-current situation is limited, but should the module not recover after a reasonable number of retries, one of the system masters should permanently disable power to the module.

7.1.3 Bus Architecture

The choice to use a simple I^2C bus with bus repeaters has it's main arguments in simplicity. The I^2C bus has been used in CubeSats many times, and is considered sufficient in many ways. The bandwidth of the bus is somewhat limited, but considered large enough for the given application. This consideration will have to be evaluated by future users of the system, when the rest of the satellite begins to take form.

The I^2C repeaters are useful to prevent isolate badly behaved slaves from the system, but care should be taken when using the `BUS_EN#` feature. The data sheet for the PCA9517 states that the enable pin should not change state during an I^2C bus operation, as doing so could hang the system. It is therefore very important that this functionality is thoroughly tested, and that due precautions are taken. For example, it could be a good idea to hold a badly behaved slave in reset before disabling the bus repeater.

7.1.4 Control Logic

The implementation of control logic and memory elements to control the power and bus switches has some advantages. Firstly, addressing removes the need for many separate control lines from the master units (one for each power/bus enable switch). The memory elements (flip-flops) remove the need for a master module to continuously output a control signal, and also makes it easier to allow multiple masters.

There are some disadvantages associated with using memory elements, however. The main issue that can turn out to be a problem is that the state of some of the switches change unintentionally due to single event effects. The logic does not currently support reading the state of the power and bus enable switches, so this might be difficult to properly detect. A simple work-around is to program one of the master modules to periodically reassign the correct state to all power switches.

The logic supports having two master modules, but care should be taken in the design of these modules. It is very important that the master modules only pull control signals low, and not drive them high (open-drain logic). There is no inherent hardware arbitration to limit access to one master at a time, so it is possible for both masters to attempt to control the backplane at the same time. The open-drain logic approach is useful here, as a conflict between two masters will not cause large currents to flow which can damage the electronics.

Some software arbitration/checking should be implemented. A simple approach is to check that all signals are idle first, then assert the address lines. The master should then check if they have the correct value before proceeding. This will ensure that the other master is not attempting to do this at the same time. If

both masters attempt to access the bus at the same time, the one with the lowest address value (most bits low) will win the arbitration, and the other master must cancel it's operation immediately.

7.2 Evaluation of Results

7.2.1 Power Distribution Unit Prototype

The results from testing the power distribution unit early on proved to be very useful. The small size of each power distribution unit demonstrated by the prototype advocated it's implementation into the backplane itself, and the variable load tests performed gave a good indication of the expected response to over-current or short-circuit conditions.

The results from the current ramp shown in the previous section visually demonstrate the auto-retry function of the circuit. It should be noted, however, that the response shown was sampled at 250 Hz, and does not give much information about rapid transients. Additional testing with an oscilloscope is recommended to make sure that the initial current pulse due to a short-circuit condition is limited to a safe value.

The slight linear decrease in supply voltage as the current increases is as expected. This is mainly due to the $R_{DS(on)}$ of pass elements in the power path. However, the sharper slope after that occurs as the current comes closer to the trip point is somewhat unexpected, but is probably due to the current-limit switch. There also appears to be some oscillations around the point where the load current crosses the current-limit threshold. This can probably be attributed to lack of hysteresis around this point.

Due to these observations, it is important that the current limit for each module slot be dimensioned to fit the requirements of the sub-system module. The risk of physical damage to components during a short-circuit condition can be reduced by lowering the current limit, but operating too close to the current limit is not recommended. It is therefore recommended that the current limit for each module to be set to about 200 mA above the expected maximum current draw of the module.

7.2.2 Backplane Hardware

The overall quality of the produced backplane was satisfactory. Only a few components have the wrong package, but apart from that no other problems can be attributed to production.

Of the components that have the wrong footprint, only U92 is actually a BOM error, meaning that the wrong component part number has been specified. The other components were probably used because they were in stock at the company which assembled the prototype, and it was therefore cheaper and easier to use them. These components are electrically equivalent, and the prototype works despite these deviations.

However, these issues need to be fixed in subsequent versions. The main problem with components with wrong packages is the mechanical stability. These components increase the risk of failing due to high g-forces from shocks or vibrations during launch.

The connectors on the backplane also have to be sorted out. The reason different connectors were mounted is because the company assembling the prototype could not obtain the specified connectors through their normal channels. This again shows the importance of considering the availability of components when designing for rapid prototyping. One solution to this problem is to specify that the assembling company should not mount any connectors at all, and rather obtain and mount them later. Fixing the exact specifications of the connectors is important towards the rest of the project, as designers of sub-system modules have to design modules that can properly fit into the backplane.

7.3 Recommended Changes/Fixes

A few changes were made to the backplane after the prototype was received. These changes need to be incorporated into a design revision before more circuit boards can be produced. A short list includes:

1. Wire mod to change pull-down resistors on ADDR lines to pull-ups.
2. Reduction of all I^2C pull-up resistor values from 10 kΩ to 2.7 kΩ.
3. Addition of a pull-up resistor on the `BP_RESET#` line.

A general revision of pull-up resistor values is recommended. Stronger pull-ups means increased current consumption, but can help reduce problems due to line capacitance or electrical noise.

The use of `ADDR0` to trigger the watchdog circuit is quite unfortunate, and will impose some limitations when controlling the backplane. At the same time the `ACK#` signal is superfluous. If eight modules are used, no address remains unused, and the `ACK#` signal will always be asserted. This is still a somewhat useful feature during development, but does not increase the reliability of the backplane. It is therefore recommended to make a small hardware change where the `ACK#`

functionality is trimmed, and that the pin on the master connector instead be used as a dedicated watchdog signal.

It can be a good idea to wait with a hardware revision until additional testing has been performed. As the backplane is used in further development, more issues are likely to arise, and it would be beneficial to collect a list of changes before committing another revision. This will limit the number separate revisions, and allow greater control over the further development of the NUTS Backplane.

Chapter 8

Backplane Conclusion & Further Work

The NUTS Backplane design has been completed, and a fully functional prototype has been produced. The design has in many ways been done with the whole satellite mission in mind, and many important architectural choices have been made. The requirements that have been drawn up are designed to provide a minimum of required functionality and safety features without creating an over-complex design. The backplane design is based on non-programmable hardware only, and does not implement any software-based components.

The prototype has been tested to some extent to verify the required functionality. A separate prototype implementation of the power distribution module used on the backplane has been produced to verify the functionality of the chosen components.

The fact that the NUTS project is based around a backplane design is what discerns it from other CubeSat projects. The choice to integrate all power distribution and bus control electronics into the backplane itself leaves most of the valuable PCB real estate on insertet modules free to use by the module developers. Furthermore, the safety features implemented in the backplane design can contribute considerably to overall system reliability, without adding enormous complexity to the rest of the system. Some of these features include redundant power supply buses, over-current protection and the ability to cycle power or isolate modules from the rest of the system. The backplane design is a good trade-off between simplicity, reliability and functionality, while still being a very low cost approach to satellite design.

Even though the implemented functions are simple in themselves, the fact eight module slots are placed on the backplane with similar functionality has resulted in a rather large design. This meant that a considerable amount of time and resources have been spent on the schematics and PCB layout. Good design practices

and prior practical experience from the industry have been essential in producing a working prototype. The board was manufactured and assembled by an external company, which turned out to be a good choice since it allowed working on other parts of the project during the production time, as well as giving much better results than would be obtained by soldering everything by hand. Choosing low cost, highly available components has been essential in producing a working prototype in such a short time frame.

Implementing and testing of a prototype of the power distribution electronics early on verified the possibility to implement redundant power rails, and it verified some of the component choices. This allowed the design of the backplane to proceed confidently. The separate verification of the power distribution module was very beneficial, as making changes to the large integrated design would have been both difficult and expensive, and might in the worst case have resulted in a non-functional backplane.

Testing of the final backplane has only been done on a very limited scale due to time limitations. Preliminary tests show expected behavior in terms of functionality and efficiency.

The NUTS Backplane is an essential part of the satellite system, and it is very important to fix the functional specifications at an early time in the project. It is therefore recommended that future students should be very careful in making architectural changes in to the backplane, as doing so might hold back the rest of the project until the new changes can be verified. Although major architectural modifications are advised against, there exists a list of changes that need to be implemented before the next prototype series can be produced. These changes mostly include some minor bug fixes and updating of component values.

Further work should be focused on thoroughly testing the backplane in all possible configurations and situations. Other outstanding work is the need to design a test rig to easily and effectively calibrate the power monitoring circuits on the backplane. This will become more important as the final satellite is being put together and software development progresses.

Part III

NUTS EPS Module

Chapter 9

EPS Introduction

Together with the NUTS Backplane, the Electrical Power System (EPS) module forms the power management solution for the entire satellite. Where the backplane has the responsibility to distribute power and to protect modules from power anomalies, the EPS module is responsible for providing that power. This includes efficient conversion of the power radiated from the sun, battery charging and monitoring, and bus voltage regulation.

In this chapter, some design requirements for the NUTS EPS module are presented, together with a short comparison of some different potential architectures for the EPS module. The background for implementing a battery charge regulator (BCR) is also presented. The design and testing of a BCR prototype will be the main focus of the following chapters.

9.1 Purpose

The purpose of the EPS module is:

- To supply $2 \times 3.3\,\mathrm{V}$ power rails to the backplane.
- To supply $2 \times 5.0\,\mathrm{V}$ power rails to the backplane.
- To charge the batteries with power from solar cells.
- To protect the batteries from over-charge and over-discharge.
- To implement and provide connection points for the required remove-before-flight pin and deployment switches.
- To provide telemetry data about battery state-of-charge and available power from solar panels.

9.2 Requirements

9.2.1 Bus Voltage Regulation

One of the main purposes of the EPS module is to convert the varying battery voltage into 3.3 V and 5.0 V for the rest of the system. To achieve this, the EPS should contain a total of four step-down DC/DC converters; $2 \times 3.3\,\mathrm{V}$ output and $2 \times 5.0\,\mathrm{V}$ output, for the four power rails on the backplane.

Some safety features is required on the power supplies, such as over-voltage protection (OVP), which is necessary to prevent power supply failures from damaging other system components. Short circuit protection is not directly required, as this is already implemented elsewhere in the system, but the regulators should be able to survive large current peaks over short periods. This requirement is necessary to ensure that the voltage regulators do not fail before the over-current protection on the backplane has time to engage.

9.2.2 Battery Management

The choice of energy storage system has been briefly discussed in the introductory part, and from here on it will be assumed that the energy storage system will consist of multiple parallel strings of two $LiFePO_4$ cells in series. This will give a nominal battery voltage of $2 \times 3.3\,\mathrm{V} = 6.6\,\mathrm{V}$. In their most discharged state, the battery voltage can be below 6.0 V, and the charge termination voltage is 7.2 V.

With all Li-Ion batteries, special considerations must be taken to ensure proper charging. The most common charging algorithm is constant-current constant-voltage (CCCV) charging, where the batteries are charged at a constant current until their voltage reaches the battery specific end-of-charge voltage, after which the charge current gradually drops while the battery voltage is held constant. Charging usually terminates when the charge current drops below a few percent of the nominal charge current.

The operational lifetime of the satellite mission is directly linked to the lifetime of the energy storage system. Improper battery handling can cause damage to the batteries, or reduce their capacity, and could easily be a major point of failure for the entire satellite. Both discharging of Li-Ion batteries below certain safe limits, and charging the batteries above safe limits can cause irreversible damage to the batteries.

The requirements needed to be fulfilled by the EPS module in terms of battery management are:

1. A proper charging strategy must be implemented (CCCV).

2. Over-voltage protection for individual cells.

3. Under-voltage protection for individual cells.

4. Over-current protection according to the maximum discharge rate specified by the batteries.

Additionally, an optional feature that could be implemented to help prolong battery lifetime is *charge balancing*, which keeps all the cells in a series string at the same potential. This is normally not strictly required for battery packs with only two cells in series, but is highly recommended when three or more cells are connected in series.

9.2.3 Solar Panels

The amount of current drawn from the solar panels should be controlled in some manner. The operational efficiency of the solar panels depend on where on the I-V curve they are used, and drawing too little current or too much current can be very inefficient. If possible, maximum power point tracking should be implemented to get the most power out of the solar panels.

9.2.4 Power Monitoring & Control

It is necessary for the EPS unit to provide, in a digital form, data about the available power, the battery state-of-charge, and estimates of how long the satellite can continue to operate at certain power levels. The basic design strategy is that it is not the EPS's responsibility to control the power consumption of the rest of the satellite, only to provide power. It should be up to the system masters, the OBC and/or the TT&C modules, to force systems to enter low power states, or to disable modules that consume too much power. In order for those modules to make good system management choices, data about the power status has to be provided by the EPS module.

However, should the batteries reach a critical depth-of-discharge (DOD), the EPS must be allowed to power down the entire satellite until the battery voltage has recovered, to prevent losing the entire mission due to battery failure. The scheme for protecting batteries from over-discharge then takes a hierarchical form:

1. System masters (OBC & TT&C) control power consumption.

2. EPS powers down the satellite in order to recover from dangerous DOD.

3. Battery protection circuits engage and disconnect batteries entirely to prevent failure.

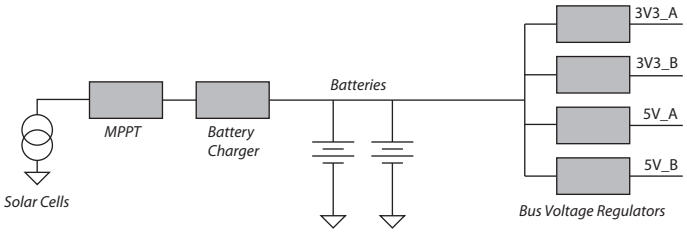


Figure 9.1: EPS Architecture

On the highest level, the system masters control the entire system to efficiently utilize the available energy. This is the default, and should in most cases be sufficient to keep the satellite running. On the next level, the EPS unit can override the masters and turn off power to the rest of the satellite if for some reason battery voltage is too low. The system can then wait for power from solar cells in order to recover. On the lowest level, battery protection circuits local to each battery cell disconnects them to prevent the cells from reaching dangerously low levels and failing.

9.2.5 RBF Pin & Deployment Switches

The remove-before-flight pin and deployment switches should interface to the EPS module in order to implement the functionality required by the CubeSat standard [6]. Additionally, the EPS module needs some means of charging the batteries while the satellite is in the P-POD launcher, so that batteries are fully charged upon launch.

9.3 EPS Architecture

As briefly discussed in section 2.2, a few different power system topologies can be implemented in a satellite, though not all of them are suitable for the NUTS project. Figure 9.1 shows the basic elements in the NUTS EPS module. In the figure, battery protection circuits are not shown. The *MPPT* block is responsible for keeping the solar cells close to their most efficient operating point. The *Battery Charger* block is responsible for implementing a CCCV charging strategy, and the four independent *Bus Voltage Regulators* provide the regulated supply buses.

9.3.1 Battery Charge Regulator

Previous work on a battery charging solution for the NUTS project by Lars Opkvitne featured a fully functional Li-Ion battery charger [17]. Such a battery

charger has many features such as battery temperature monitoring and safety timers, and is in reality a rather complex device. The charger suggested used a step-down converter charge two or three cells in series, and requires an input voltage slightly higher than the charge termination voltage.

Achieving a high enough voltage for the battery charger requires all 4 solar cells on the long sides of the satellite to be connected in series. This would give about $4 \times 2.63\,\mathrm{V} = 10.52\,\mathrm{V}$ open circuit, and about 9.4 V at the maximum power point. But this leaves the two cells on the top of the satellite in a somewhat awkward situation, as the voltage from only two cells in series is not enough to charge the batteries by themselves.

In this work some simplifications to the battery charging solution are suggested. These simplifications allow the battery charger to be combined with maximum power point tracking in a single regulator. These simplifications are:

1. Remove constant current control from the charger.

2. Use a standard boost converter instead of a specialized charge IC.

The reason for the constant-current part of a CCCV charge strategy is to protect the batteries from charging too quickly and becoming damaged. The safe charging current for slow charging of $LiFePO_4$ batteries is usually 1C. This means that the charge current for a 2.2 Ah battery pack should not be more than 2.2 A in the constant-current phase.

In the NUTS satellite constant current control is both unnecessary and impractical. Consider the expected maximum power from the solar panels presented in table 2.1: 7.2 W. Even with no losses between the solar panels and the batteries, and the solar panels operating at their maximum power point, the charge current can never not exceed $7.2\,\mathrm{W}/6.0\,\mathrm{V} = 1.2\,\mathrm{A}$ when the batteries are discharged. Higher battery voltages means even less available current. There is simply not enough power available to charge the batteries too quickly, and constant current limiting in the battery charger can be safely removed.

It is impractical because it will be difficult to keep the charge current constant as the irradiance on the solar panels vary through orbit. It is desirable to use *all* available power to power the system and charge the batteries, and constant current limiting would mean wasted power.

The proposed solution uses a boost regulator with input and output voltage tracking. The output voltage control is to enforce the constant-voltage part of the charge cycle, and the input voltage control can be used bias the solar panels to their most efficient operating point. A boost regulator has been chosen so that the two top solar panels can contribute equally to the charging of the batteries. The combined MPPT and charger unit is called a Battery Charge Regulator (BCR), and will be the focus of the next chapters.

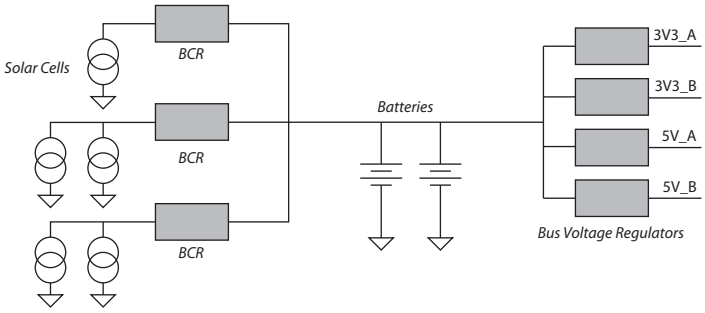


Figure 9.2: EPS with 3 BCR units

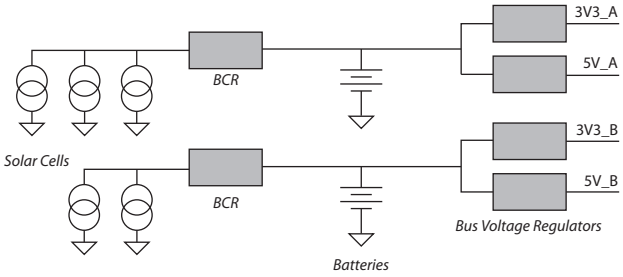


Figure 9.3: Separate battery charging

Several BCR regulators can be used in parallel. This is very useful since it allows maximum power point tracking of individual panels, which will have different MPPs due to differences in irradiance and temperature. Instead of using one BCR per side of the satellite, a good trade-off can be to use one BCR for solar cells on opposite sides of the satellite, since only one of these sides will be in the sun at any given time. With solar cells on five of six sides, this will require three BCR units, as shown in figure 9.2. A similar strategy is used in commercial CubeSat power solutions from Clyde Space, as presented in [7].

Using several BCR units also functions as a built-in redundancy: the loss of a single BCR unit or damage to a solar panel will not disable the entire system.

9.3.2 Redundant Power Sections

While charging the batteries in parallel works well, it might be desirable to split up the batteries and charge them separately for increased reliability, as shown in figure 9.3. This approach is elegantly supported by the NUTS Backplane, which already incorporates redundant power rails.

The greatest advantage of this approach is that all failure modes of the batteries and voltage regulators are handled. The most common failure mode for the

batteries is to fail open rather than fail short, and the architecture of figure 9.2 will survive this. However, if a short circuit should arise somewhere (for example at the input of a voltage regulator), this can disable the entire system.

However, this scheme also has some disadvantages compared to the simple direct connection. If a voltage regulator should fail, half of the battery capacity, and half of the solar power is also lost. A loss of a voltage regulator in the system of figure 9.2 does not entail any capacity reduction. The loss of half the operational power can be a serious problem for the satellite.

These factors have to be carefully considered and weighed up before a final decision about the power system architecture can be made.

Chapter 10

EPS Implementation

Based on the basic EPS architecture described in the previous section, a design for a battery charge regulator (BCR) is proposed. The design is based around using a commercially available step-up (boost) regulator IC, with extra components in the feedback loop. A microcontroller is added to the design so that a MPPT algorithm can be implemented, and so that telemetry data can be made available to the rest of the system over the common I^2C bus.

10.1 Battery Charge Regulator

A normal Li-Ion charger operating in CCCV mode can be simplified as a regulator which limits the current based on two factors: charge current and battery voltage. In a switch-mode charger, regulation is achieved by adjusting the duty cycle of the pulse-width modulation. The *most limiting* of the two factors controls the duty cycle. For example, if the battery voltage is far from the final charge voltage, the output voltage of the regulator is not a limiting factor. Instead, the duty cycle is adjusted so that the charge current does not exceed the programmed limit.

As discussed in the previous chapter, the need for a fixed charge current is deprecated in the NUTS project. Instead of using a fixed charge current, one can "choose" a current such that the solar panels are biased at their most efficient operating point. As shown in figure 2.3, the output voltage from the solar panels will vary based on the load current according to the I-V characteristic. The opposite is also true: the load current will vary based on the bias voltage, and it is this principle that will be used to control the battery charger. Thus, by adjusting the regulator duty cycle based on the input (solar panel) voltage, proper biasing of the solar cells can be implemented.

As the batteries reach the programmed charge voltage, output voltage regulation takes over, and the charge current is slowly reduced. This also means that

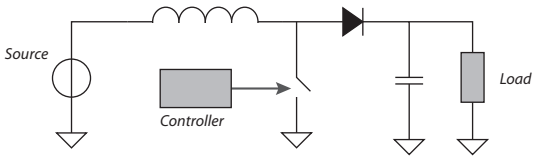


Figure 10.1: Basic Step-up Converter

the input voltage is slowly backed off (increases), until the input voltage finally reaches the cell open-circuit voltage V_{OC}. At this time the charge current is 0 A. In the NUTS satellite this will never happen, as the system load will always require some current even if the battery is fully charged.

By dynamically adjusting the set-point for the regulator input voltage during the *constant input voltage phase*, maximum power point tracking can be achieved. This is where the microcontroller is used. Instead of programming the entire regulation algorithm into the microcontroller, only a very simple tracking algorithm is programmed to adjust the set-point. Current regulation around this set-point, and transition into constant-voltage mode, is achieved by using a dedicated switch-mode control IC.

10.1.1 Step-up Converter

The basic boost, or step-up converter topology is shown in figure 10.1. The switch is usually a MOSFET controlled by a pulse-width modulated (PWM) signal. The difference between the input and the output voltage is determined by the duty cycle of the PWM signal.

The current through the inductor increases during the time the switch is closed. When the switch is opened, the only path for the current flow is through the flyback diode into the capacitor and the load. This results in the energy that that has been accumulated in the inductor during the On-state to be transfered to the capacitor.

One way to create a battery charge regulator is to use a microcontroller to generate the PWM signal that controls the power switch. The microcontroller would then combine maximum power point tracking and battery charging in a single algorithm and control the duty cycle of the PWM signal accordingly. This is the approach taken in the CubeStar project at the University of Oslo (though they use a step-down converter instead) [18].

Instead the NUTS BCR design is based around a commercial boost converter IC to control the MOSFET switch. This has several advantages over the microcontroller implementation:

1. No discrete PID controller needs to be implemented in the microcontroller,

as a boost converter IC has built-in compensation. The analog control also loop offers higher bandwidth and better control.

2. Higher switching frequency is achievable. A higher switching frequency means less ripple current and smaller inductors, which take up less of the valuable space and weight budget. As a comparison, consider a microcontroller timer running at 32 MHz and a resolution of 8-bits. The highest possible PWM frequency is $32\,\text{MHz}/2^8 = 125\,\text{kHz}$, and if higher resolution is required, the frequency has to be lowered even further. Modern switch-mode controllers operate at frequencies of 300 kHz to over 1 MHz.[1]

3. Higher accuracy is achievable, as no analog-to-digital conversion is necessary. This can be essential when charging Li-Ion batteries as they have high requirements on charge voltage accuracy. Using an analog circuit instead of a microcontroller also eases the needs for calibration.

4. Reliability. It is generally considered that microcontrollers in the NUTS project have a large chance to fail due to single event upsets (SEU), caused by the high levels of radiation in the space environment, that can corrupt program or data memory. It can be very dangerous if something as important as solar power conversion is directly dependent on software.

Some criteria when selecting a converter chip were:

- At least 2 A switch current. (Sufficiently higher than expected maximum from solar panels)
- High efficiency, preferably over 90%.
- Input voltage range up to 6 V (Higher than maximum solar panel V_{OC}).
- Output voltage range up to 8.4 V (To accommodate charging of different Li-Ion battery chemistries).

Two different devices were considered: the AP1609 from Diodes Inc, and the TPS61086 from Texas Instruments. The AP1609 is a high-efficiency step-up converter with a built-in N-channel MOSFET, working at around 300 kHz, and comes in a SO8 package. The TPS61086 is also a high-efficiency step-up converter with integrated MOSFET, but works at 1.2 MHz, and comes in a tiny 3×3 mm QFN package.

Although the higher switching frequency of the TPS61086 makes it attractive, the AP1609 has been chosen for the prototype instead. The prototype is intended

[1] However, very high switching frequencies tend to lead to higher switching losses in MOSFET devices.

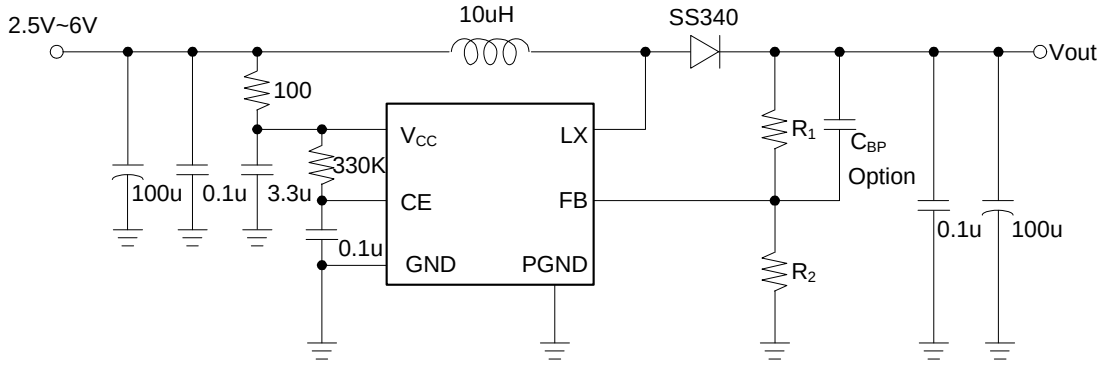


Figure 10.2: AP1609 Typical Application. Taken from the data sheet [13]

as a proof-of-concept design, and the final choice of switching regulator IC can be changed at a later stage. The main reason for choosing the AP1609 is the package: the SOIC package is much easier to handle and solder by hand than the QFN package.

10.1.2 Charge Termination Control

The AP1609 works by constantly comparing the voltage at it's feedback (FB) pin with an internal reference. In a standard application, as shown in figure 10.2, an external voltage divider programs the output voltage so that:

$$V_{fb} = V_{out}\left(\frac{R_2}{R_1 + R_2}\right) = V_{ref} \rightarrow V_{out} = V_{vref}\left(1 + \frac{R_1}{R_2}\right) \tag{10.1}$$

The high accuracy required in the end-of-charge voltage when charging Li-Ion batteries can be a problem. For example, the accuracy of the AP1609's internal reference is only specified to $\pm 2.5\%$. If charging a $LiFePO_4$ battery to 7.2 V, the actual charge termination voltage could be as high as 7.38 V due to innucracy in the regulator alone. For $LiFePO_4$ batteries this can be dangerously high, so most dedicated Li-Ion charge IC's specify 0.5% battery voltage accuracy.

To overcome this problem, an external reference with higher accuracy is used in the NUTS BCR. The idea comes from a Li-Ion Battery charge controller from National Semiconductor, the LM3420. The LM3420 is intended to be used together with a linear or switching regulator, to provide 0.5% accurate end-of-charge control. The device consists of an op-amp with an open-emitter output, a voltage reference and an integrated voltage divider. The LM3420 is intended to directly drive the feedback pin of a regulator.

Unfortunately, the LM3420 is not available in any versions that would suit the end-of-charge voltage of $LiFePO_4$ batteries, so the devices cannot be used in the

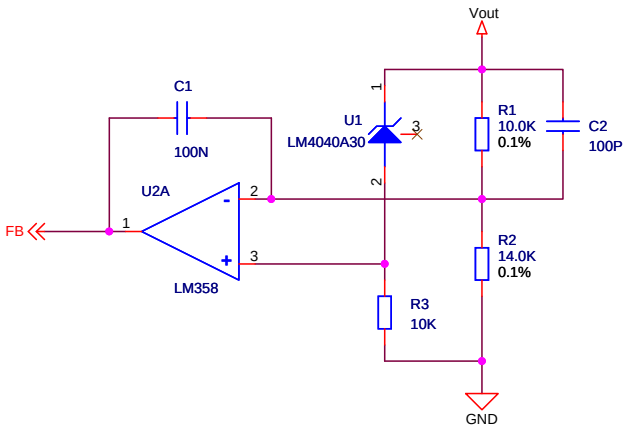


Figure 10.3: BCR End-of-Charge Control.

NUTS BCR design. Instead, the functionality is copied and implemented with discrete components, consisting of a standard op-amp, a precision reference and some precision resistors. The resulting circuit is shown in figure 10.3.

With the reference voltage of 3.0 V, the reference at the non-inverting input of the op-amp is always

$$V_+ = V_{out} - 3.0\,\mathrm{V} \tag{10.2}$$

and the voltage at the inverting input is

$$V_- = V_{out}\left(\frac{R_2}{R_1 + R_2}\right) \tag{10.3}$$

The op-amp regulates the voltage at the feedback pin, which again regulates V_{out} so that $V_+ = V_-$, giving the output voltage as:

$$V_{out} - 3.0\,\mathrm{V} = V_{out}\left(\frac{R_2}{R_1 + R_2}\right) \rightarrow V_{out} = 3.0\,\mathrm{V} \cdot \left(1 + \frac{R_2}{R_1}\right) \tag{10.4}$$

With the selected values of 10 kΩ and 14 kΩ, the battery voltage becomes exactly 7.2 V, as desired. By changing the feedback resistors, the output voltage can be adjusted to provide accurate end-of-charge control for any battery technology (or number of cells, as long as the total voltage is within the limits of the AP1609).

In the circuit, C1 and C2 provide additional frequency compensation.

10.1.3 Input Voltage Control

Controlling the input voltage of the AP1609 is done in basically the same way as controlling the output voltage, except that the sign of the control law is changed.

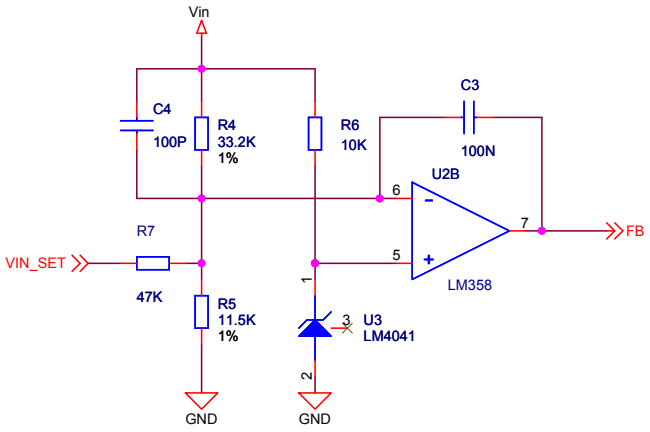


Figure 10.4: BCR Input Voltage Regulation

Instead of limiting the maximum input voltage, a limit needs to be imposed on the *minimum*. Increased current through the regulator means lower input voltage according to the I-V characteristic, so that current limiting is the same as imposing a *lower* limit on the input voltage.

The circuit for controlling the input voltage is shown in figure 10.4. The input voltage control for the NUTS BCR does not have the same accuracy requirements as the end-of-charge control, as the input voltage set-point is intended to be modified by a tracking algorithm. The non-inverting input is now always biased to the voltage reference:

$$V_+ = V_{ref} = 1.225\,\mathrm{V} \tag{10.5}$$

and disregarding the `VIN_SET` for now,

$$V_- = V_{in} \cdot \frac{R_5}{R_4 + R_5} \tag{10.6}$$

When $V_- = V_+$, the input voltage becomes:

$$V_{in} = V_{ref}\left(1 + \frac{R_4}{R_5}\right) \tag{10.7}$$

With $R_4 = 33.2\,\mathrm{k\Omega}$ and $R_5 = 11.5\,\mathrm{k\Omega}$, the input voltage set-point becomes about 4.76 V, which has been chosen because it equals the maximum power point voltage (V_{MPP}) of two solar cells in series.

By applying an analog voltage to the `VIN_SET` term, a microcontroller can control the input set-point. The input voltage set-point now becomes

$$V_{in} = V_{ref}\left(1 + \frac{R_4}{R_5}\right) + \frac{R_4}{R_7}\left(V_{ref} - V_{set}\right) \tag{10.8}$$

10.1.4 Combining both Control Laws

Both the input voltage controller and the output voltage controller cannot be directly connected to the feedback pin at the same time. Doing so will create a conflict between the two control laws.

For example, if the battery voltage is to be kept constant, the end-of-charge controller will source current to force the feedback pin high. This will cause a reduction in charge current, and likewise an increase in input voltage as the solar panels are backed off the MPP. This will again cause the input voltage regulator to try to lower the voltage at the feedback pin, effectively preventing output voltage control.

To prevent this from happening an addition law is enforced: *each controller can only drive the feedback pin high.* This will prevent one controller disrupting the other by pulling the feedback pin low. A very simple and effective way of accomplishing this to use diodes in the feedback path. The completed schematic diagram for the Battery Charge Regulator is shown in figure 10.5.

Not shown in figure 10.5 is the power supply to the op-amps. Another BAT54C diode is used to power the op-amps from either the solar panel (input) voltage or the battery voltage, whichever is the highest. This allows the BCR to start up even if the batteries are completely dead as long as solar power is available.

10.2 Analog Watchdog Circuit

The BCR is intentionally designed to work without software control or intervention. The greatest advantage of not relying on a microcontroller lies in the potential increased reliability. However, possibility for software control has been added to the BCR in order to accommodate a tracking algorithm for the solar panel maximum power point, which will give additional efficiency as the solar panel conditions change. The software is given control over the input voltage set-point, and thus also the current in the regulator, and can in this way also disable the regulator if needed.

Allowing a microcontroller to influence the BCR can have a negative effect on reliability. To illustrate this, consider a PWM signal is used to control the input voltage set-point. If the microcontroller hangs (for example due to a SEU in data memory), and this causes the PWM signal to remain stuck at either 0 V or 3.3 V, the BCR will be compromised. This will either force the operating point of the solar cells to the short-circuit current area, where the voltage would be too low for the regulator to properly function, or force the operating point to the open-circuit voltage area, where the current is zero.

In order to prevent the described scenario, a very simple watchdog circuit has

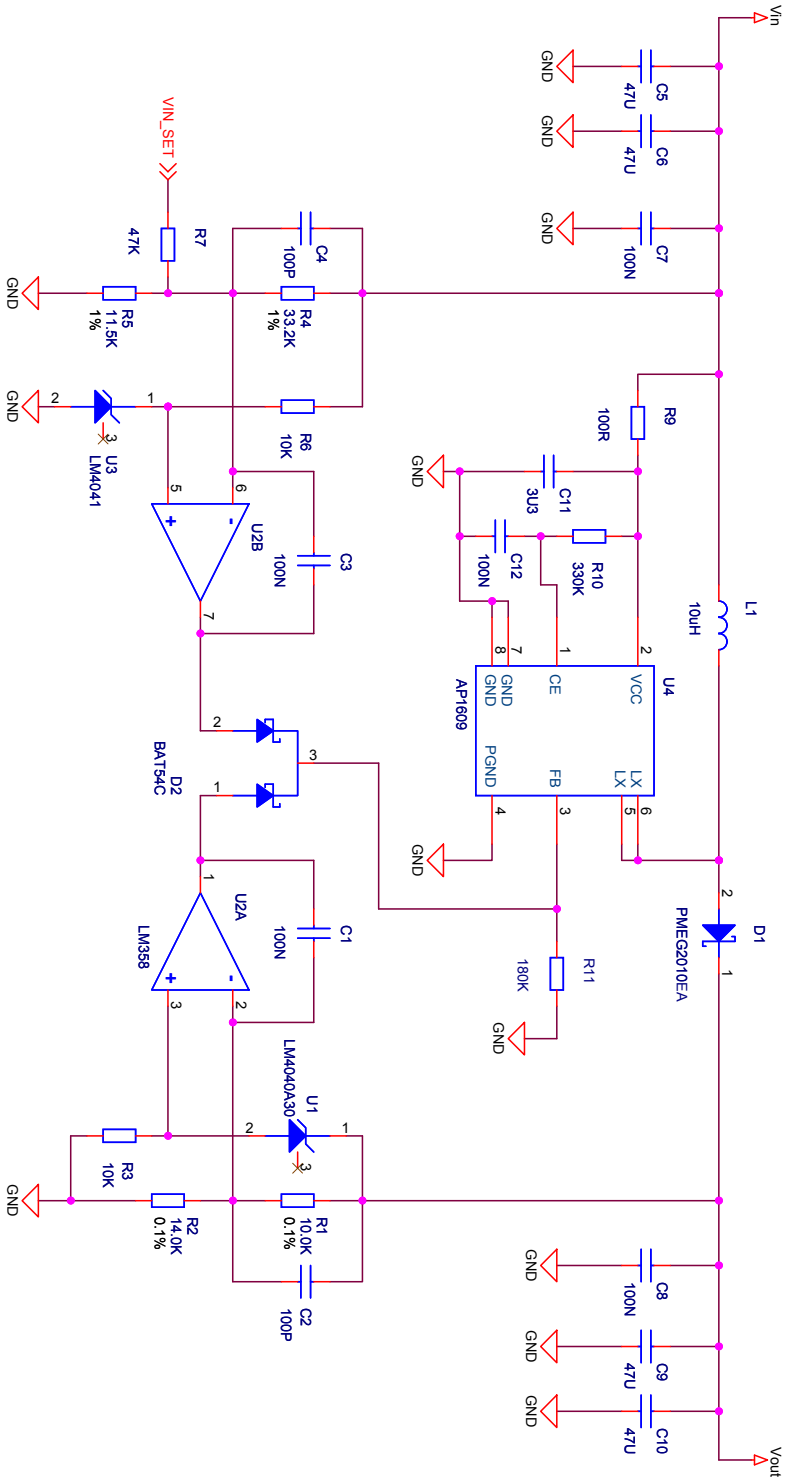


Figure 10.5: Battery Charge Regulator

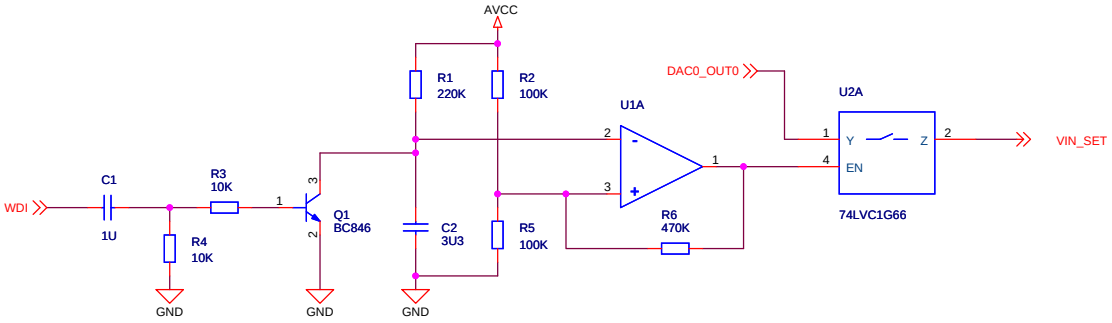


Figure 10.6: EPS Watchdog Circuit

been designed to prevent a fault in the microcontroller from compromising the BCRs. The basic idea is to use an analog switch to disconnect the microcontroller from the BCR if it is not responding. This will cause the BCR to revert back to the default hardware set-point, and continue to function, though without dynamic MPPT.

The watchdog circuit is shown in figure 10.6. It consists of a high-pass filter, a timing circuit, a comparator with hysteresis and an analog switch. The timing circuit consists of resistor R1, capacitor C2 and transistor Q1. Capacitor C2 is slowly charged through R1, and can be "reset" by turning on Q1. If the voltage over C2 rises above the threshold set by R2 and R5, the comparator disables the switch. R6 is adds some hysteresis to the comparator. The transistor Q1 is controlled by the `WDI` signal from the microcontroller, passed through a high-pass filter consisting of C1 and R4. The high-pass filter prevents a stuck-at fault from constantly resetting the timer.

The length of the watchdog timeout can be calculated. The voltage across the capacitor with t_0 defined as just after the last positive flank of WDI is given by:

$$V_{C_2}(t) = V\left(1 - e^{-t/(R_1 C2)}\right) \tag{10.9}$$

With R2 = R5 = 100 kΩ the threshold voltage is $V/2$ (disregarding the slight influence added by the hysteresis), and the circuit times out when the capacitor voltage reaches this:

$$V\left(1 - e^{-t/(R_1 C2)}\right) = \frac{V}{2} \rightarrow t = -R_1 C_2 \log \frac{1}{2} \tag{10.10}$$

This equals to about 1.5 seconds for the component values used on the prototype ($R_1 = 100\,\text{k}\Omega$ and $C_2 = 10\,\mu\text{F}$).

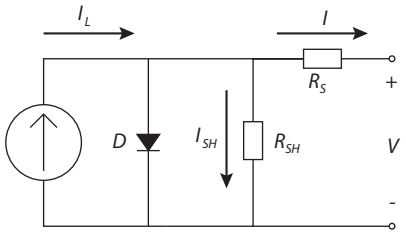


Figure 10.7: One-Diode Model

10.3 MPPT Algorithm

A tracking algorithm is necessary if the power from the solar cells is to be maximized. The BCR default set-point can be set according to the I-V characteristic at a certain temperature, but this will not keep the solar cells at their MPP for all temperature and irradiance conditions. In order for the solar cells to be used most efficiently all the time, the maximum power point must be tracked as the I-V characteristic changes.

A very simple algorithm for finding the maximum power point based on voltage and current measurements has been evaluated. The algorithm is based on the "perturb and observe" (P&O) concept, which is widely used as a simplified feed-forward control strategy for MPPT. [15] A simplified model of a solar panel has been created in MATLAB, and the algorithm was tested on this model using Simulink.

10.3.1 The One-Diode Model for PV cells

The simplest model to describe the behavior of a solar cell is the one-diode model, where a current source is in parallel with a diode. The diode determines the I-V characteristics of the cell. Also added to the model is a series resistance R_S and a shunt resistance R_{SH}, to represent internal losses due to current flow and leakage current to ground respectively. See figure 10.7.

The I-V characteristic of the solar array is given by

$$I = I_{SC} - I_0 \left(\exp \left[\frac{q\,(V + R_S I)}{nkT_k} \right] - 1 \right) - \frac{V + R_S I}{R_{SH}} \tag{10.11}$$

where V and I represent the output voltage and current, q is the electronic charge, I_{SC} is the light-generated current, I_0 is the reverse saturation current, n is a dimensionless diode factor, k is the Boltzmann constant, and T_k is the temperature in °K. [15]

For simplicity, the cell temperature was fixed to 25 °C, and the other parameters in the model where experimentally chosen to approximately mimic the I-V

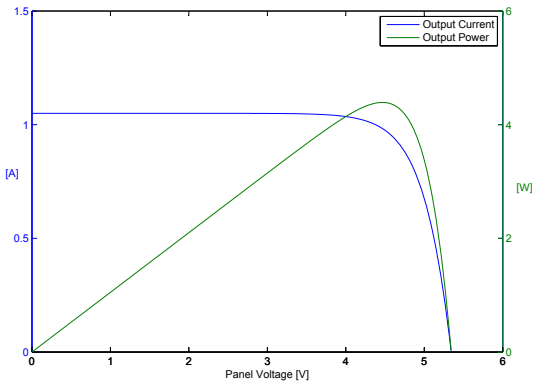


Figure 10.8: Simplified model I-V characteristic

characteristics of the 30% Triple Junction GaAs solar cell from Azur Space, given in [3]. A look-up table was generated with the simplified I-V characteristic for a panel consisting of four cells: two parallel strings of two cells each, equivalent to one long side-panel of the NUTS satellite. Figure 10.8 shows the current and power vs. panel voltage for the simplified model.

10.3.2 The Perturb and Observe Algorithm

The very simple MPPT algorithm is illustrated in figure 10.9. The algorithm works by continuously taking small fixed-size steps (perturb), and measuring the result (observe). During each iteration, voltage and current measurements are used to calculate the solar panel power, and is compared with measurements from the previous iteration. If the result of the step is an increase in power from the previous iteration, the algorithm continues stepping with the same slope. If however the result from the step is a decrease in power, the slope is inverted. This is basically a hill climbing algorithm, and is feasible because the power from the I-V characteristic has only one (global) maximum.

The MPPT algorithm is very simple, and requires only a few lines of code to implement. The tunable parameters are the step size and the sampling/stepping frequency. The fixed step-size of the algorithm will cause the controller to oscillate around the maximum power point as it is continuously stepping up and down. By choosing a smaller step-size, the amplitude of these oscillations decrease, but the convergence time increases. Choosing a higher sampling frequency will cause faster convergence, but the BCR must be allowed to settle to the new input voltage set-point between each step.

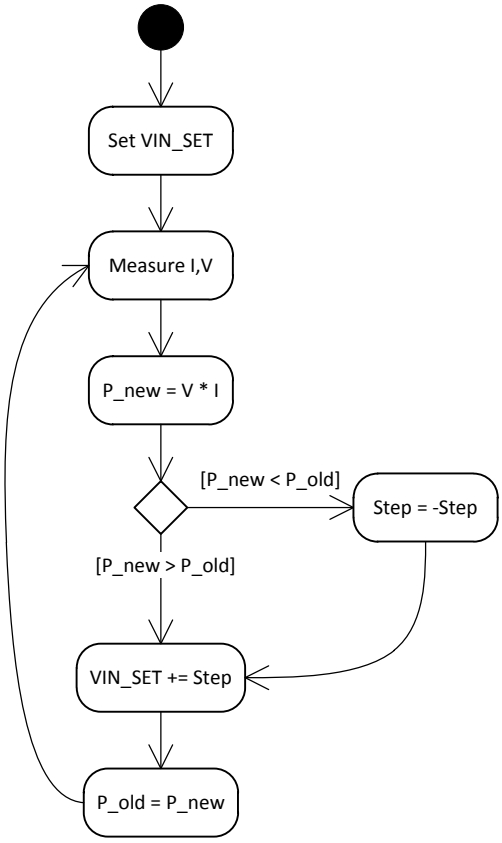


Figure 10.9: Simple MPPT Algorithm

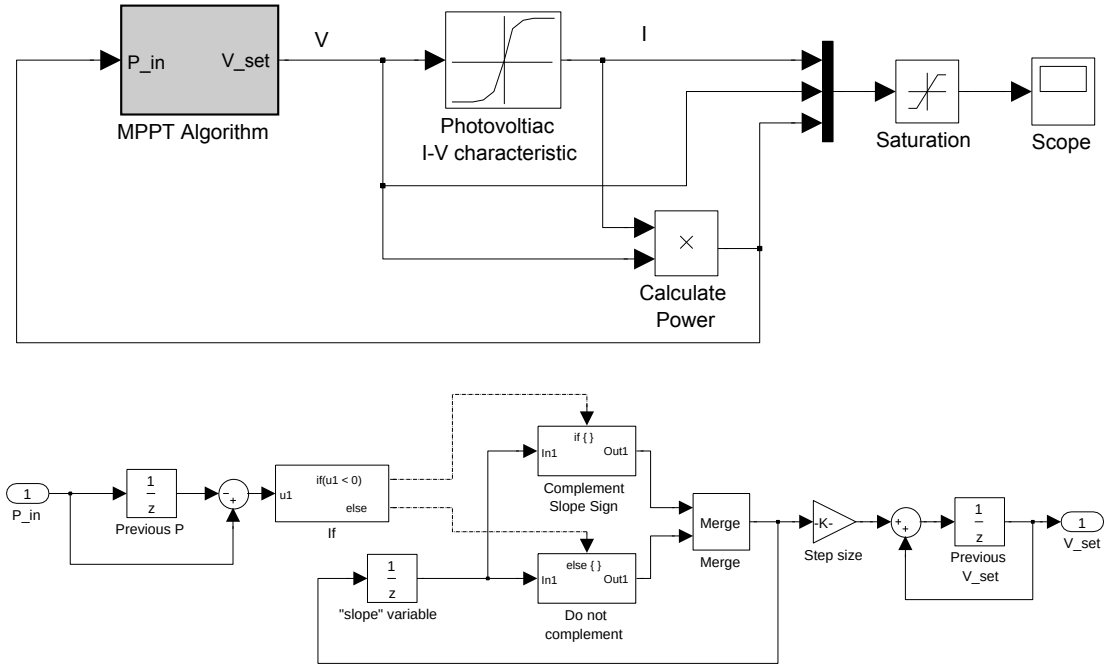


Figure 10.10: Simulink Test of MPPT Algorithm

10.3.3 MATLAB Implementation

A simple test of the algorithm together with the PV model described above has been implemented in Simulink. The panel I-V characteristics is added as a lookup table in Simulink, which interpolates between the values in the table. The MPPT algorithm is built with discrete blocks, and supplies the voltage input to the lookup table. See figure 10.10.

Chapter 11

EPS Testing & Verification

A proof-of-concept prototype of the BCR circuit has been produced to test the feasibility of the solution. The prototype unit consists of the BCR circuit, the Analog Watchdog circuit, an EFM32 microcontroller with voltage and current measuring, and a serial-to-USB converter for measurement data transfer to a PC. Figure 11.1 shows a block diagram of the BCR prototype setup.

11.1 Microcontroller

An EFM32G210F128 microcontroller from Energy Micro AS. is used on the BCR prototype to perform measurements, and to test control of the BCR input voltage set-point. The EFM32 features a powerful 32-bit ARM Cortex-M3 processor, and is optimized for ultra low power consumption through the use of advanced sleep modes and autonomous peripherals.

The ultra low power consumption of the EFM32 make it ideal for use in the EPS module, because the EPS microcontroller should be allowed stay active even when

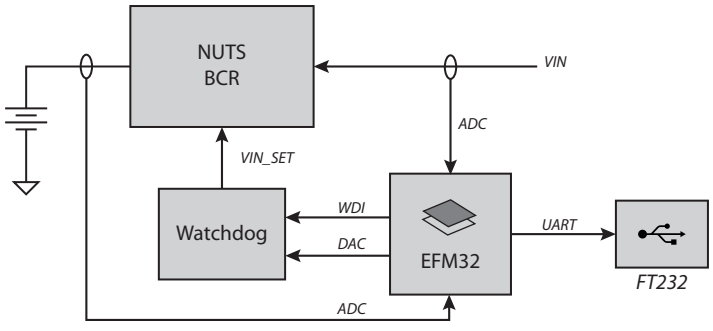


Figure 11.1: BCR Prototype Block Diagram

the rest of the system is shut down to recover from deeply discharged batteries. Peripherals used on the BCR prototype include a 12-bit analog to digital converter (ADC), a 12-bit digital to analog converter (DAC), a USART for communicating with a PC, and an I^2C interface which can be used to interface to the system common bus.

The DAC of the EFM32 is connected to the `VIN_SET` input of the BCR, through the watchdog circuit's analog switch, to test control over the input voltage set-point. The DAC is used with an internal 2.5 V reference, allowing the input voltage set-point to be controlled between about 3.9 V and 5.6 V, according to equation 10.8.

11.1.1 Voltage & Current Measurements

Voltage and current measuring capability was designed into the BCR prototype to allow testing without the need for external measurement equipment. This is especially useful since a complete charging cycle can be quite long, and this scheme allows collected data to be stored on a PC for future analysis.

Voltage measurements of the input and output voltages of the BCR are available using the EFM32's ADC. A simple resistor divider brings the voltages into range of the ADC's internal 1.25 V reference.

Current measurement is done with a 10 mΩ current shunt (high side) between the input and the BCR and between the BCR and the battery. Two INA214 differential current sense amplifiers from Texas Instruments are used to amplify the voltage drop across the resistors to useful values for the ADC to read.

11.1.2 Communication

An FT232 serial-to-USB chip was placed directly on the BCR prototype module, together with a USB-mini-B connector. This eliminates the need for a large DSUB-9 RS232 connector, RS232 level shifting, or external RS233-to-USB cable.

The I^2C bus lines are also brought out to a connector, so that communication of telemetry data to the rest of the system can be evaluated at a later stage.

11.1.3 Software

The software on the EFM32 was set up to continuously sample the ADC at a frequency of 20 Hz. The raw data from voltage and current measurements are converted to actual values in volts and amperes, based on programmed calibration values. The resulting floating point values are sent over the USART using a simple `printf()` statement.

(a) $LiFePO_4$ Battery Pack (b) Protection circuit revealed

Figure 11.2: Battery Pack used in testing of the BCR Prototype

A safety timer feature has also been added to the EFM32 software. After 4 hours, the EFM32 forces the BCR input voltage set-point high, so that charging ceases.

On the PC, a simle LabView program is used to capture the measured values, display them in a real-time graph, and finally log them to a file for later analysis.

11.2 Batteries

The batteries used to verify proper battery charging were two $LiFePO_4$ battery packs provided by Gylling Teknikk AS. Each battery pack contains four A123 APR18650M1A cells. Each cell has a nominal voltage of 3.3 V, a capacity of 1.1 Ah, measures 18×650 mm and weighs 39 g. The series-parallel connection result in a nominal battery voltage of 6.6 V and a capacity of 2.2 Ah, which is roughly equal to about 14.5 Wh.

The battery packs come with an integrated protection circuit, protecting the batteries from over- and under-voltage, and from over-current. It additionally features cell balancing to keep the cells at the same voltage.

11.3 Power Source

The solar cells that will be used in the NUTS project are quite expensive, at around 200 USD each, and can be difficult to obtain. The complete satellite will require 18 cells in total (with a total value around 20000 NOK).

Because of this there were no solar cells available yet, and an alternative approach had to be taken to test the BCR prototype. Ideally, a programmable current

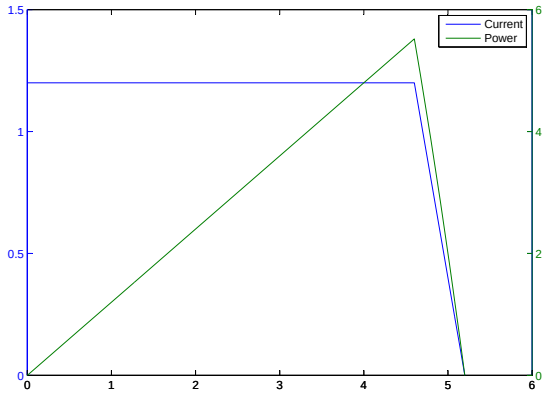


Figure 11.3: I-V characteristic of a current limited PSU with series resistor

source could be used, where the instrument can be programmed with the I-V characteristics of the solar cells. Unfortunately this is also expensive and somewhat specialized equipment, and was not available during the time of writing.

A simple approach has been used to testing the BCR. A laboratory power supply together with a small series resistor of $0.5\,\Omega$ was connected to the input of the BCR. The voltage over the resistor increases with increasing current, mimicking the rightmost part of the solar cell I-V characteristic. By adjusting the voltage of the power supply, the power can be controlled, as the BCR is supposed to keep it's input voltage at a constant level:

$$I_{in} = \frac{V_{supply} - V_{in}}{0.5\,\Omega} \tag{11.1}$$

With $V_{supply} = 5.2\,\mathrm{V}$, the current drawn when $V_{in} = 4.7\,\mathrm{V}$ is around $1\,\mathrm{A}$. The "open circuit voltage" of the supply, when the BCR is not drawing any current, is equal to V_{supply}. By using the current limiting feature of the power supply, an upper "short circuit current" limit can be imposed.

The resulting characteristic is shown in figure 11.3. Though not as elegant as the PV model in figure 10.8, this will suffice for testing the BCR's input voltage regulation, but not for testing the MPPT algorithm.

11.4 Efficiency Measurement Setup

The efficiency of the BCR circuit can be calculated as

$$\eta = \frac{P_{out}}{P_{in}} = \frac{V_{out} I_{out}}{V_{in} I_{in}} \tag{11.2}$$

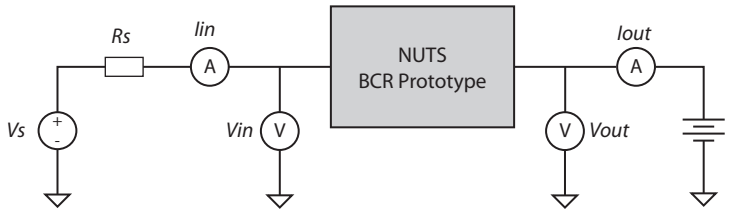


Figure 11.4: Efficiency Measurement Setup

It is obvious that small inaccuracies in any of the four terms will lead to wrong estimates of the efficiency, so a more accurate test setup than the integrated measurements was devised. This setup is shown in figure 11.4, and uses four digital multimeters that can be read at the same time.

Chapter 12

EPS Results

12.1 BCR Prototype

The resulting BCR prototype is shown in figure 12.1. The prototype PCB was produced with a milling machine, and all the components were soldered by hand.

The initial version displayed some stability problems in the form of audible noise from the inductor. Adjusting the compensation components greatly improved the frequency stability.

Calibrating the integrated measurement circuits proved to be very difficult. This means that the measurements are not very accurate, but they still suffice to verify the operation of the BCR.

12.1.1 Battery Charging

Figure 12.2(a) shows the battery voltage and current when charging from around 6.5 V to 7.2 V. Since the input power to the BCR is constant during input voltage control, the charge current decreases slightly as the battery voltage increases. As soon as the battery voltage reaches the end-of-charge voltage, the end-of-charge control takes over and keeps the battery voltage constant and forces the charge current down.

The input voltage and current to the BCR can be seen in figure 12.2(b). The input voltage control causes the input current to remain constant at 1 A during the input voltage control phase. As soon as the BCR enters the constant output voltage mode, the input voltage is allowed to increase as the need for charge current goes down.

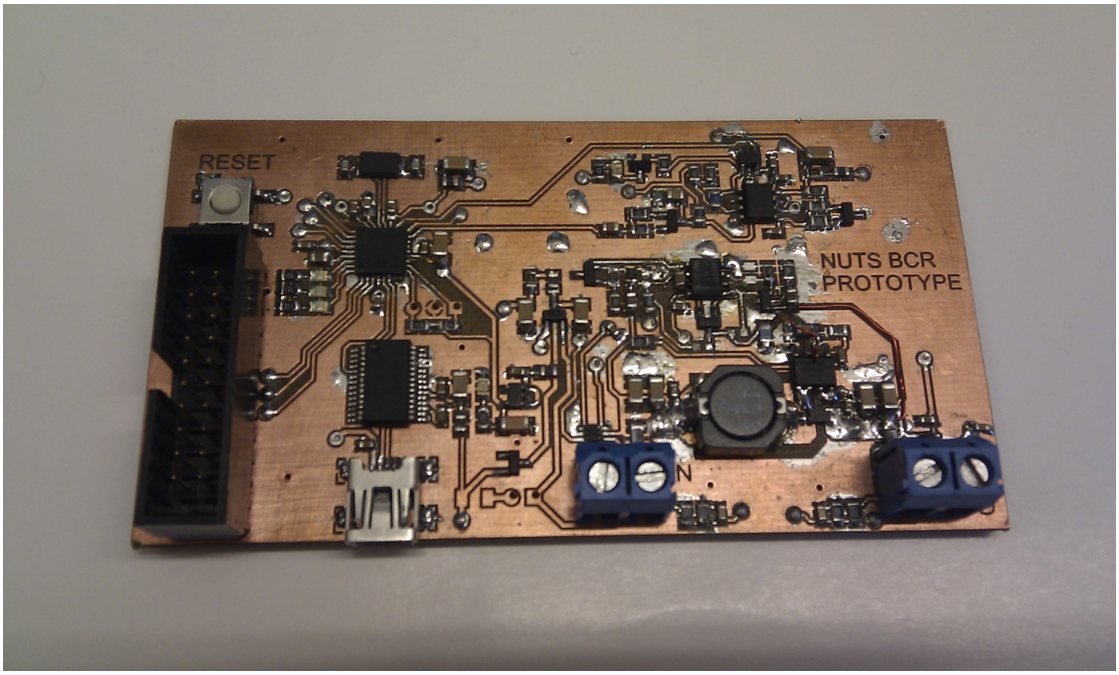


Figure 12.1: BCR Prototype

12.1.2 End-of-charge Voltage Accuracy

The end-of-charge voltage for the prototype has been measured to 7.19 V. This is equal to a voltage accuracy of

$$1 - \frac{7.19\,\mathrm{V}}{7.20\,\mathrm{V}} = 0.00138 < 0.2\% \tag{12.1}$$

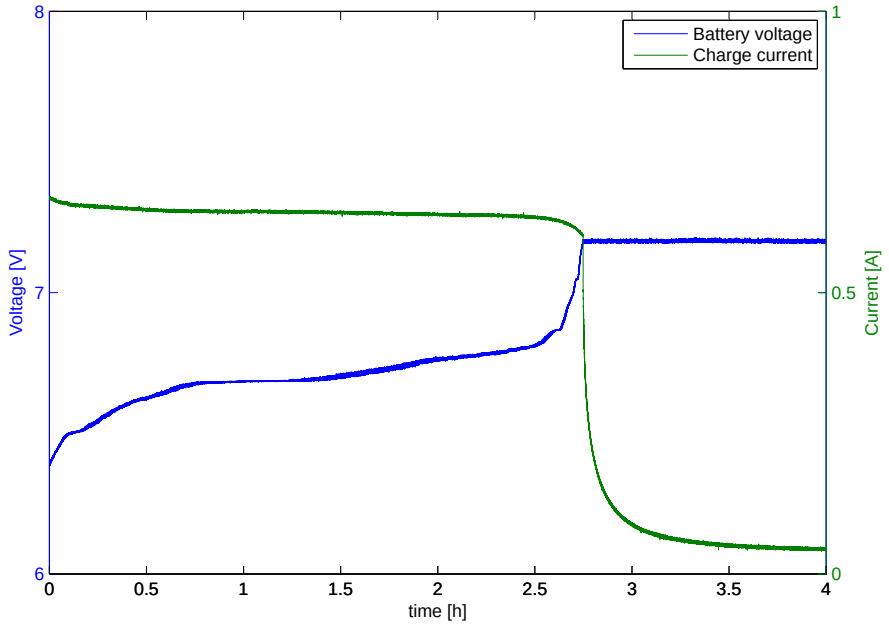
12.1.3 Efficiency

The BCR efficiency when the battery voltage is at a nominal of 6.6 V, the input voltage at 4.76 V and the input current set to 1 A has been measured to 91%. Periodic measurements during charging indicate that the overall efficiency remains above 90% for the entire constant input voltage phase.

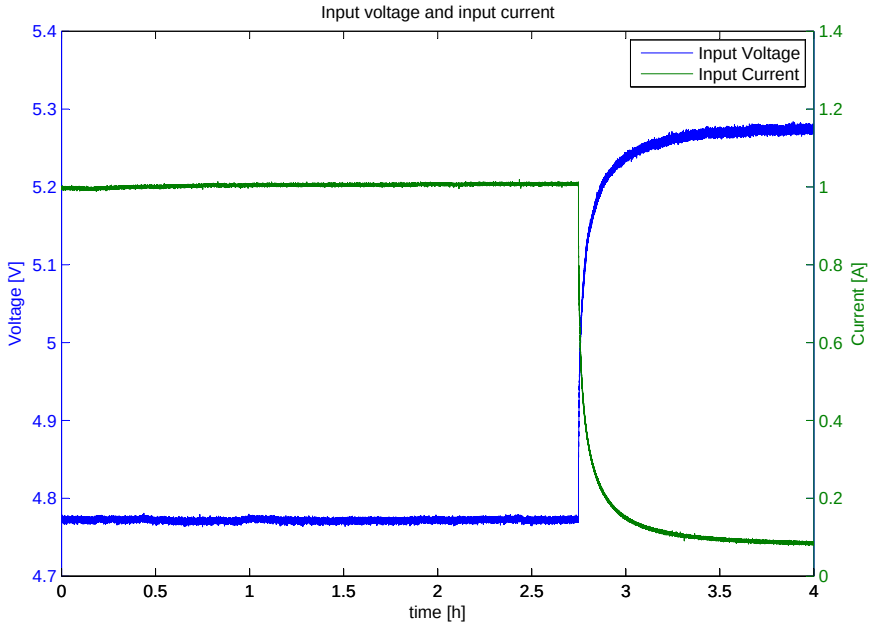
As the input current decreases when the battery end-of-charge voltage has been reached, the efficiency of the converter also drops drastically.

12.1.4 Input Voltage Set-point Biasing

Simple tests such as writing values into the DAC output register show that the input voltage set-point can be adjusted by the EFM32.



(a) Battery Charge Current and Voltage



(b) BCR Input Voltage and Current

Figure 12.2: BCR Results

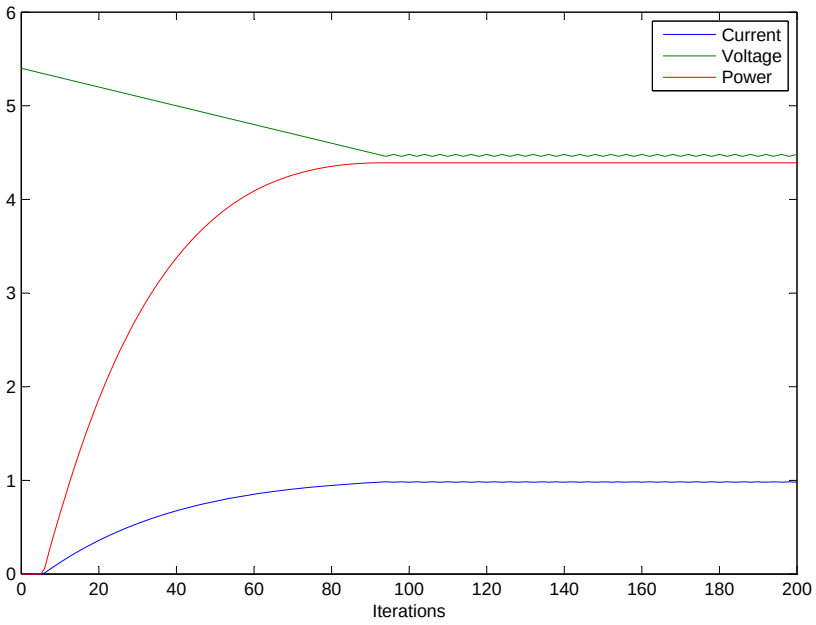
12.1.5 Analog Watchdog Circuit

The watchdog circuits operates as intended. If pulses on the WDI signal cease for more than 1.5 seconds, the analog switch turns off, preventing the EFM32 from changing the input voltage set-point.

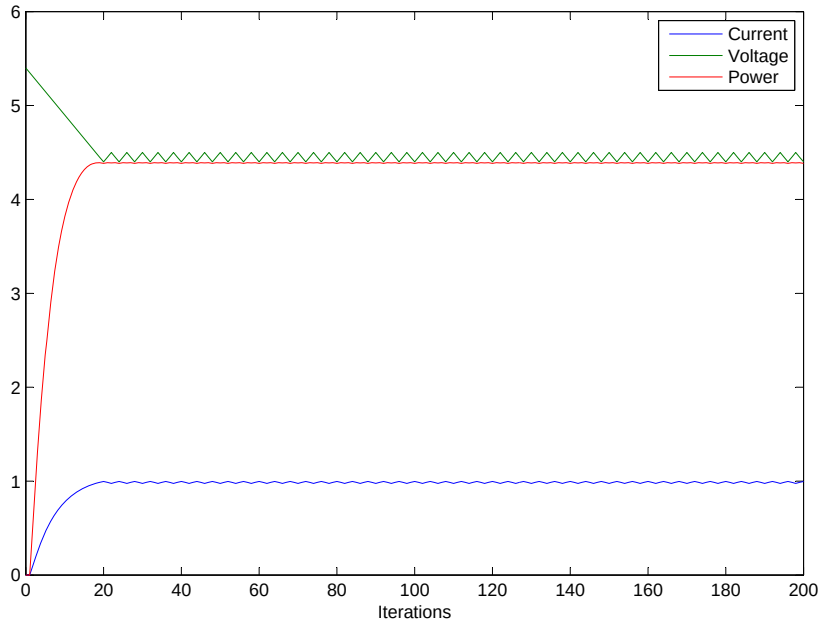
12.2 MPPT Algorithm

The result of the MATLAB simulation of the simple MPPT algorithm is given in figure 12.3. In both cases the algorithm starts at the open-circuit voltage of the solar panels, then gradually decreases the panel voltage, increasing the current until the maximum power point is reached.

The effect of varying the step size is demonstrated: the larger step size causes the algorithm to converge to the MPP much faster, but causes a significant voltage ripple as soon as the MPP is reached.



(a) Step size = 10 mV



(b) Step size = 50 mV

Figure 12.3: Simulation of MPPT Algorithm

Chapter 13

EPS Discussion

13.1 Design Evaluation

The proposed design for the EPS module uses a simple battery-voltage bus approach, where the input to the four system voltage regulators always follow the battery voltage. The solar cells are used at their maximum power point, and all the power not used for the system load goes towards charging the batteries. As the batteries reach their end-of-charge voltage, the input power from the solar cells is reduced by moving their operating point towards the open-circuit voltage.

More traditional approaches with MPPT constantly keep the solar panels at their MPP, and uses current shunts to remove the excess power. The shunts generate heat, which can be problematic to remove from the system. With the elegant dynamic approach presented in this work, this problem is eliminated.

The EPS architecture presented is well suited to a nanosatellite application. In contrary to large commercial satellites, a nanosatellite has very little power production, and efficiency is of key concern. In contrast to the fully-regulated bus approach, which uses separate charge, discharge and perhaps also MPPT regulators, the NUTS approach uses only one regulator in the power path, potentially offering much more efficient use of the available power.

Separate voltage regulators for the 3.3 V and 5 V buses allow the rest of the satellite to operate consistently despite changes in battery voltage, which can vary significantly. The presented power management approach allows each component in the power path to be utilized at it's highest efficiency: the solar panels through MPPT, the battery through proper charge management, and finally the system through separate regulators for the bus voltages. These three "domains" are isolated in terms of voltage, and the net result is that stable 3.3 V and 5 V buses are available as long as the batteries are not completely discharged, independent of whether solar power is available or not.

The BCR circuit used on the EPS is designed to be very robust and reliable, by largely using relatively simple analog components. The BCR is designed to operate without software by default, and provides a fixed voltage bias point for the solar panels, which can easily be adjusted by modifying some resistor values. The addition of a very simple tracking algorithm in the EPS microcontroller augments the BCR functionality, providing even higher efficiency by keeping the solar panels at their MPP over temperature and irradiance variations. The watchdog circuit has been added to prevent a failed microcontroller from becoming a single-point-of-failure for the entire satellite.

The tracking algorithm itself is extremely simple, and requires only a few lines of C code to be implemented. The main reason for it's simplicity is the fact that the actual tracking of input and output voltages of the regulator is done by analog hardware. From a control systems perspective, this is cascaded control with the MPPT algorithm implementing implementing feed-forward control for the outer loop, and the BCR itself performing closed-loop control of the inner loop. Using an analog inner control loop provides much higher bandwidth and resolution than could be achieved with discrete control in a small low-cost microcontroller. The design of the inner control loop has also been greatly simplified by basing the design on a standard boost converter chip.

The design of the BCR allows several units to be used in parallel on the EPS module. Either three separate regulators (opposite side panels plus top panel), or five regulators (one regulator for each panel) could be used, and gives great flexibility if the solar cell configuration should change at a later stage in the project. The number of BCR's will probably be limited by the physical size of the implementation, as everything has to fit on a $9 \times 9\,\mathrm{cm}$ circuit board. Most components for the prototype version has been chosen to be simple to solder by hand, and are thus not size-optimized. A final version of the BCR could be significantly smaller, allowing up to five regulators to be implemented.

13.2 Evaluation of Results

The initial prototype required some tuning before it operated reliably. Some problems were observed with the stability of input tracking, and a high pitched noise could be heard emitting from the inductor, indicating unwanted oscillations well below the switching frequency. Adding capacitors for compensation improved significantly on this. In figures 10.3 and 10.4, the capacitors C2 and C4 add an additional 90° phase lift at higher frequencies, effectively increasing the phase margin of the loop. The values of the compensation components were found experimentally, but proper analysis of the phase-margin should be done to prove stability. Unfortunately, not much information about the gain and compensation

of the AP1609 is given, so analyzing the whole control loop becomes quite difficult.

Integrating hardware for measuring the characteristics of the BCR into the prototype board seemed like a good idea at first, but proved to be more problematic than intended. The main difficulty was calibrating the ADC gain and offset parameters, and a significant amount of time was spent in trying to properly calibrate the measurements using a manual procedure. Several times an accidental short circuit somewhere on the PCB caused a failure of the microcontroller, and each time it was replaced the new one had to be recalibrated.

The measured efficiency of the BCR at 91% is very good. This makes the custom NUTS BCR comparable with commercial CubeSat EPS solutions, such as the one from Clyde-Space presented in [7], which specifies the overall efficiency at "about 90%".

The battery end-of-charge voltage tracking also displays excellent accuracy, with better than 0.2% regulation achieved in the lab setup. This good, as charging the batteries to a slightly too high voltage can cause lasting damage, and not charging the batteries to their maximum level means less power will be available in eclipse. The high accuracy of the BCR end-of-charge control allows efficient use of the battery capacity, while not exceeding safe levels.

The simulations of the MPPT algorithm show that it works, but not much can be said about performance until it has been integrated into the BCR prototype and tested with a proper power source. Unfortunately not enough time or resources (solar cells/power source) were available to complete proper verification of the power point tracking. However, the greatest result from the MPPT algorithm is it's apparent simplicity. In software, as in hardware, simple implementations can lead to higher reliability, as there are less elements that can fail. The simplicity of the MPPT tracking algorithm suggests it as good candidate for implementation in the final EPS.

Adjusting the input voltage set-point of the BCRs has also been tested only to a small extent. Since a resistor was used with a voltage source, the set-point has only been moved along a linear characteristic. Proper verification using real solar cells or a solar cell emulator should be done to verify stability.

13.3 Recommendations for Further Work

A lot of work remains to finish the EPS module. The work done so far has been mostly at a proof-of-concept stage, and mainly shows the feasibility of the proposed solution. All the different components have yet to be implemented in a form that can be plugged into the system backplane. This includes connectors for solar cells and batteries, separation switch and remove-before-flight pin, BCRs, design of 3.3 V and 5 V conditioning, and battery protection. All these features have yet to

be integrated to form a complete EPS module.

As well as integrating all the components into a complete EPS module, more testing is also needed on the presented solutions. It is possible that the BCR regulator needs further analysis and tuning to guarantee stability. If different, smaller, components are chosen for the BCR this tuning will have to be done again for the new components.

The MPPT algorithm needs to be implemented together with the BCR and tested with real solar cells to measure the actual overall efficiency and MPPT accuracy.

Chapter 14

EPS Conclusion

The EPS module is an essential part of the NUTS system in that it provides electrical power to the rest of the satellite. The primary tasks of the EPS module are to efficiently condition the available power from the solar cells, to safely charge the batteries, and to provide two regulated 3.3 V and two regulated 5 V power rails. Secondary tasks include voltage and current measurements, calculation of available solar power and determination of battery state-of-charge.

As part of the EPS, a prototype of a battery charge regulator (BCR) has been developed and tested. The BCR integrates solar panel maximum power point tracking with battery charging in a single stage. The BCR is based on a commercial step-up converter with a custom feedback loop adding input voltage tracking in addition to battery end-of-charge control. Several units can be used in parallel for redundancy and individual tracking on separate panels. The input voltage set-point can be controlled from the EPS microcontroller, allowing a simple tracking algorithm to be implemented to track the solar cell's most efficient operating point over changing temperature and irradiance conditions. To prevent a failed microcontroller from disabling the BCR, an analog watchdog circuit has been implemented to force the input voltage set-point back to a default value upon time-out. The constructed BCR prototype includes the measurement circuitry used to monitor regulator input and output voltage and current.

The prototype BCR has been produced and tested in the lab. A four cell $LiFePO_4$ battery pack has successfully been charged by the BCR prototype, which shows an efficiency of 91 %, and end-of-charge voltage control with 0.2% accuracy.

Input voltage set-point adjustment using a microcontroller together with the analog watchdog circuit has also been tested and verified. A simple algorithm used for tracking the maximum power point of the solar cells has been demonstrated and simulated using MATLAB. The extreme simplicity of the tracking algorithm makes it a very good candidate for reliable implementation in the final EPS.

Although the BCR prototype is still very experimental, it demonstrates how

a maximum power point tracker and battery charger can be combined, forming a relatively simple, but highly efficient solution. The high efficiency achieved is of key importance, as the available power in a nanosatellite is very limited. The overall simplicity of the solution, together with an added watchdog timer for safety, combined with the fact that most of the hardware is based on purely analog components result in a highly robust and reliable design.

Part IV

Summary

Chapter 15

Conclusion

During the course of this project a power management system for a small student satellite has been designed. The work done forms part of the NTNU Test Satellite (NUTS), which is scheduled for launch some time during 2014. Even though the project is still at an early stage, a lot of progress has been made in the course of one semester.

The problem of Power Management and Distribution for a small student satellite is highly diverse, encompassing many different aspects, and has been split into two parts: Power Distribution and Power Conditioning. An ambitious attempt at solving both problems in a very limited time frame has been pursued to a great deal of success.

A complete system backplane with integrated power distribution and protection has been designed for the NUTS project. The backplane forms an important part of the satellite, and many high level architectural design choices have been made that will affect the course of the whole project. In addition to implementing power distribution in the backplane, other vital system functions such as a common communication bus, backplane control logic, and possibilities for sub-module reset and recovery has been designed and implemented. The result is a fully functional backplane prototype.

The system backplane provides a robust and reliable platform for integrating different satellite modules, and form the very basis of the NUTS satellite. The NUTS backplane provides some advanced safety features not usually found in comparable commercial CubeSat solutions. Some of these safety features include dual redundant power supply rails and short-circuit protection for all system modules. At the same time, simplicity and robustness is ensured by basing the design on discrete logic components, eliminating software vulnerability from the backplane.

The architecture of the backplane itself supports two master modules working as a self-checking pair, where both modules have the ability to reset, disable and perform recovery on the rest of the satellite. All control logic as well as power dis-

tribution and bus isolation for all modules has been integrated into the backplane hardware, saving valuable space on inserted modules.

The design is well thought through, and appears to be a good trade-off between simplicity and reliability. Simple solutions has been chosen wherever possible, and separate verification and testing has been done to verify the integrity of the design choices. The backplane approach is what puts the NUTS project apart from other projects concerned with building CubeSats, and the fully functional backplane prototype produced during this work is essential in driving the project ahead.

In addition to the design of a backplane prototype, a proposed design for an Electrical Power System (EPS) module has also been presented. The EPS handles charging batteries from solar cells and power conditioning for the rest of the system, and also provides battery protection and important power telemetry data to the rest of the system. As part of the proposed EPS design, a proof-of-concept prototype of a battery charge regulator (BCR) has been constructed and tested.

The BCR effectively combines maximum power point tracking with proper battery charge control in a single high efficiency conversion stage. It allows the solar panels to be utilized at their most efficient operating point, and it allows all power not used by the rest of the satellite to be used to charge the batteries.

Control of the BCR is divided into an analog part and a digital part. The analog part completely handles input and output voltage tracking by using standard components in a slightly non-standard configuration. The digital part implements an extremely simple hill-climbing algorithm to steer the solar cell operating point towards the maximum power point. A watchdog timing circuit has been devised to protect the analog part from a failing microcontroller.

The BCR displays excellent performance in terms of efficiency and end-of-charge control. With over 90% efficiency, little power is wasted in power conversion, and little heat is generated. The end-of-charge voltage is controlled to within 0.2% accuracy, allowing the battery capacity to be fully utilized while always staying below the safe charge voltage limits.

Although the EPS module is still very much experimental, and a lot of work remains before a finished EPS is ready, the work done here presents some very good preliminary results. It shows how a simple charge regulator can be custom built from relatively standard components, while still achieving high efficiency and excellent control, and being a very low-cost solution.

Bibliography

[1] Clyde space overview. http://www.clyde-space.com/about_us.

[2] Gomspace company introduction. http://www.gomspace.com/index.php?p=profile.

[3] AZUR SPACE Solar Power GmbH. *30% Triple Junction GaAs Solar Cell*.

[4] Peter Berlin. *Satellite Platform Design*. Department of Space Science, University of Luleå, Kiruna Sweden, fifth edition, 2007.

[5] Roger Birkeland, Elisabeth Blom, and Erik Narverud. Design of a small student satellite. Master's thesis, NTNU, 2006.

[6] California Polytechnic State University. *CubeSat Design Specification*, 2009.

[7] Craig S. Clark. An advanced eelctrical power system for cubesats. *European Small Satellite Services Symposium*, 2010.

[8] Linear Technology Corporation. *LTC4413: Dual 2.6A Ideal Diodes*, 2011.

[9] Leonard David. Cubesats: Tiny spacecraft, huge payoffs. http://www.space.com/308-cubesats-tiny-spacecraft-huge-payoffs.html.

[10] Edgar Elden. Power management system for a pico satellite. Master's thesis, NTNU, 2008.

[11] Ingulf Helland. Systems platform for nano and pico satellites. Master's thesis, Department of Electronics and Telecommunications, NTNU, 2008.

[12] A123 Systems Inc. Development of battery packs for space applications. *NASA Aerospace Battery Workshop*, 2007.

[13] Diodes Inc. *PWM/PFM Dual Mode Step-Up DC/DC Converter*, 2009.

[14] Texas Instruments Incorporated. *INA219: Zero-Drift, Bi-Directional CURRENT/POWER MONITOR with I2C Interface*, 2010.

[15] Ying-Tung Hsiao Joe-Air Jiang, Tsong-Liang Huang and Chia-Hong Chen. Maximum power tracking for photovoltaic power systems. *Tamkang Journal of Science and Engineering*, 2005).

[16] Surrey Satellite Technlogy Ltd. Satellite classification. `http://centaur.sstl.co.uk/SSHP/sshp_classify.html`.

[17] Lars O. Opkvitne. Effektstyring ncube. Master's thesis, Department of Electronics and Telecommunications, NTNU, 2008.

[18] Martin Oredsson. Electrical power system for the cubestar nanosatellite. Master's thesis, University of Oslo, 2010.

[19] Maxim Integrated Products. *MAX14523: 250mA to 1.5A, Adjustable Current-Limit Switches*, 2011.

[20] Spectrolab. *29.5% NeXt Triple Junction (XTJ) Solar Cells*, 2010.

[21] Agilent Technologies. *AN 1293 - Sequential Shunt Regulation*, 1997.

Part V

Appendix

Appendix A

NUTS Backplane PCB

A.1 Assembly Drawing

The assembly drawing shows the position and orientation of each component on the PCB. This is very useful to refer to a component cannot be located or identified based on the silk print on the PCB alone. The assembly drawing for the top side is given in figure A.1, and the assembly drawing for the bottom side components is given in figure A.2.

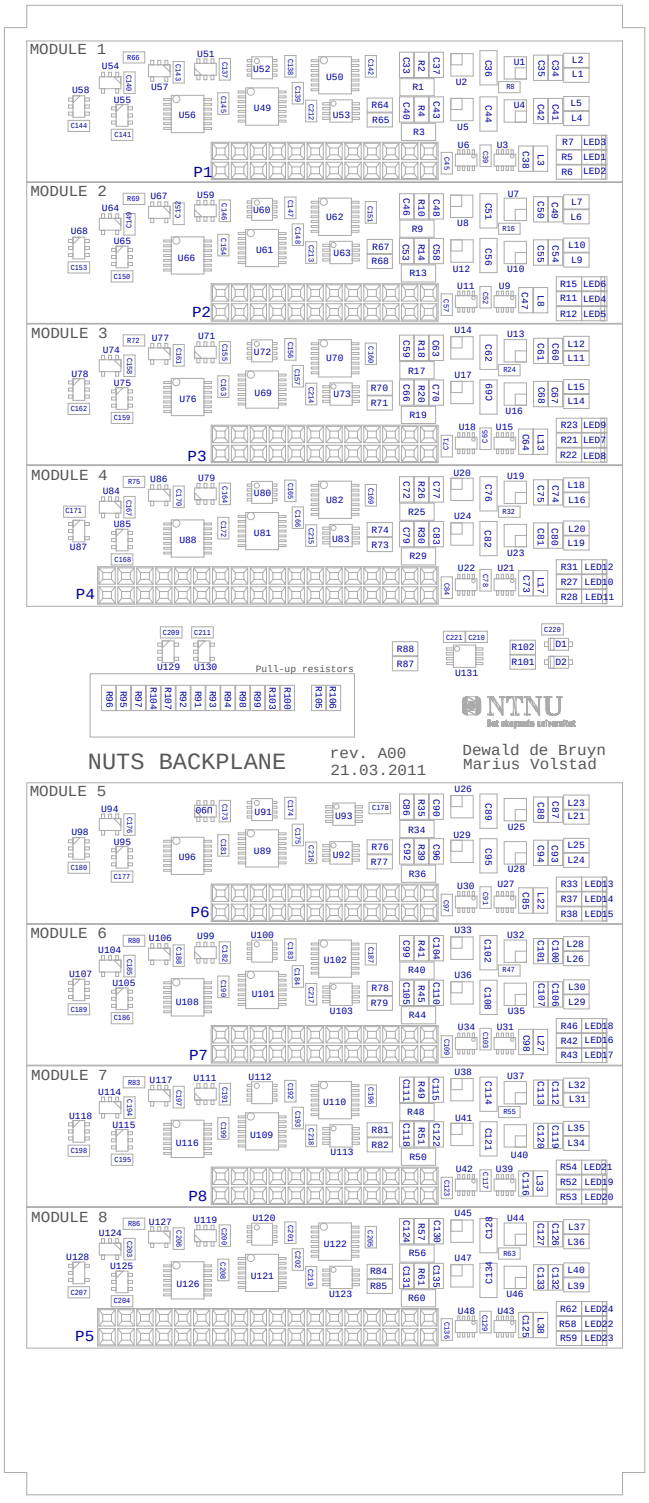


Figure A.1: Backplane Assembly Drawing (top)

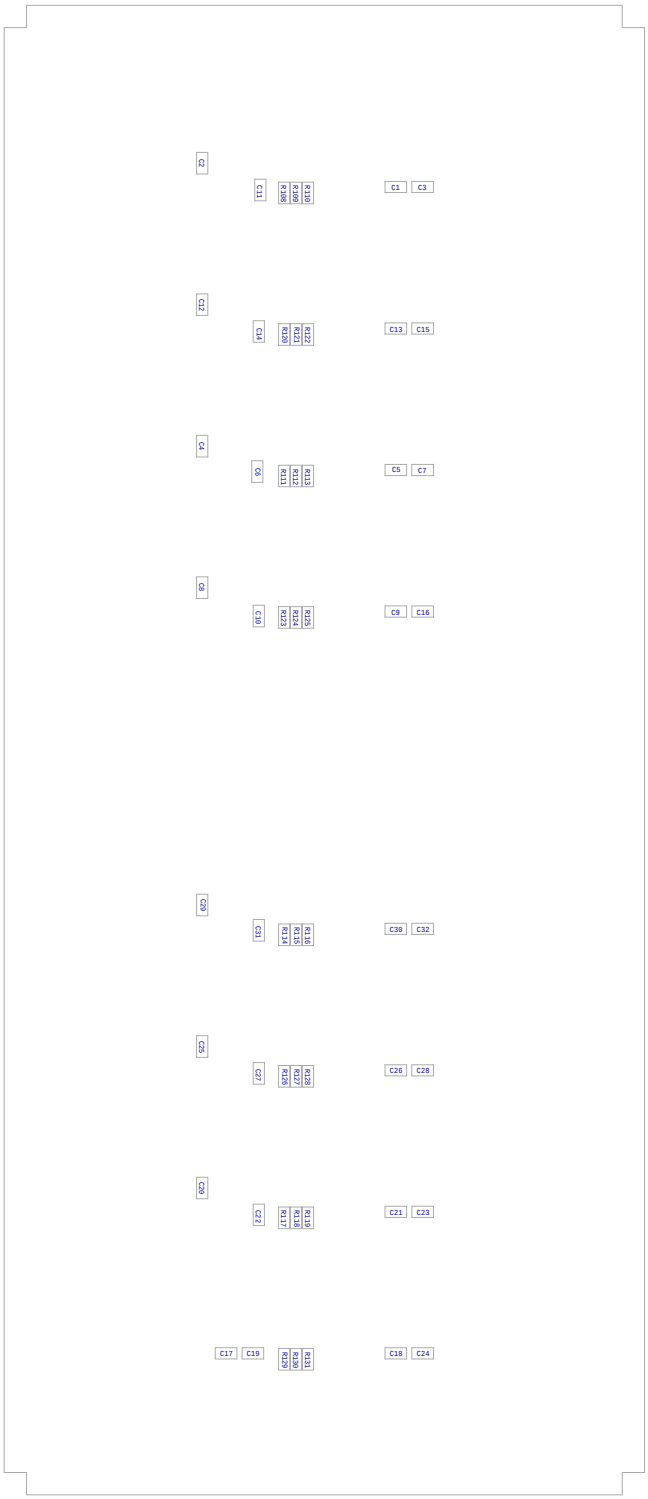


Figure A.2: Backplane Assembly Drawing (bottom)

A.2 Bill of Materials

The Bill of Materials, or BOM, is an excel file that lists all the components and their part numbers. A print-out of the BOM is shown in figure A.3.

NUTS Backplane Revised: Tuesday, March 29, 2011
Revision: A01

Item	Quantity	Reference	Part	Manufacturer	Manufacturers Part Number	Footprint	Tolerance	Rating	Technology
1	128	C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,C25,C100N	100N	Murata	GRM188	T_0603	25V	X7R	
2	56	C33,C34,C35,C38,C40,C41,C42,C46,C47,C49,C50,C53,C54,C55,C59,C60,C61,C64,C66,C67,C68,C72,C7.10U	10U	Murata	GRM21	T_0805	10V	X7R	
3	16	C36,C44,C51,C56,C62,C69,C76,C82,C89,C95,C102,C108,C114,C121,C128,C134	22U	Murata	GRM31	T_1206	10V	X7R	
4	16	C37,C43,C48,C58,C63,C70,C77,C83,C90,C96,C104,C110,C115,C122,C130,C135	1U	Murata	GRM21	T_0805	10V	X7R	
5	4	C209,C210,C211,C220	100N	Murata	GRM188	T_0603	25V	X7R	
6	2	D1,D2	MBRS020L			SOD123			
7	16	LED1,LED2,LED4,LED5,LED7,LED8,LED10,LED11,LED14,LED15,LED16,LED17,LED19,LED20,LED22,LED2GREEN	KingBright	APHCM2012CGCK	T_0805				
8	8	LED3,LED6,LED9,LED12,LED13,LED18,LED21,LED24	RED	KingBright	APHCM2012SURCK	T_0805			
9	32	L1,L2,L4,L5,L6,L7,L9,L10,L11,L12,L14,L15,L16,L18,L19,L20,L21,L23,L24,L25,L26,L28,L29,L30,L31,L32,L	BLM22PG221S	Murata	BLM21PG221S	T_0805			
10	8	L3,L8,L13,L17,L22,L27,L33,L38	BLM21BD102S	Murata	BLM21BD102S	T_0805			
11	6	P1,P2,P3,P6,P7,P8	RECEPTACLE_2X12	Tyco Electronics	6-534206-2	TYCO_AMPMODU_24			
12	2	P4,P5	RECEPTACLE_2X18	Tyco Electronics	6-534206-8	TYCO_AMPMODU_36			
13	16	R1,R3,R9,R13,R17,R19,R25,R29,R34,R36,R40,R44,R48,R50,R56,R60	0R04			T_1206	1 %		
14	16	R2,R4,R10,R14,R18,R20,R26,R30,R35,R39,R41,R45,R49,R51,R57,R61	470K			T_0805	1 %		
15	8	R5,R11,R21,R27,R37,R42,R52,R58	1K			T_0805			
16	16	R6,R7,R12,R15,R22,R23,R28,R31,R33,R38,R43,R46,R53,R54,R59,R62	470R			T_0805			
17	38	R8,R16,R24,R32,R47,R55,R63,R66,R69,R72,R75,R80,R83,R86,R108,R109,R110,R111,R112,R113,R114 100K				T_0603			
18	20	R64,R65,R67,R68,R70,R71,R73,R74,R76,R77,R78,R79,R81,R82,R84,R85,R87,R88,R101,R102	10K			T_0805			
19	15	R91,R92,R93,R94,R95,R96,R97,R98,R99,R100,R103,R104,R105,R106,R107	100K			T_0805			
20	16	U1,U4,U7,U10,U13,U16,U19,U23,U25,U28,U32,U35,U37,U40,U44,U46	LTC4413	LinearTechnology	LTC4413EDD	DFN10			
21	16	U2,U5,U8,U12,U14,U17,U20,U24,U26,U29,U33,U36,U38,U41,U45,U47	MAX14523	Maxim	MAX14523A	TDFN8			
22	16	U3,U6,U9,U11,U15,U18,U21,U22,U27,U30,U31,U34,U39,U42,U43,U48	INA219			SOT23-8			
23	8	U49,U61,U69,U81,U89,U101,U109,U121	74LVC86A	Texas Instruments	SN74LVC86APW	TSSOP14			
24	7	U50,U62,U70,U82,U102,U110,U122	74LVC74A	Texas Instruments	SN74LVC74APWR	TSSOP14			
25	8	U51,U59,U71,U79,U90,U99,U111,U119	74LVC1G332	Texas Instruments	SN74LVC1G332DBVR	SOT23-6			
26	8	U52,U60,U72,U80,U91,U100,U112,U120	74LVC2G02	Texas Instruments	SN74LVC2G02DCTR	SM8			
27	9	U53,U63,U73,U83,U92,U103,U113,U123,U131	PCA9517	Texas Instruments	PCA9517DGKR	TSSOP8			
28	8	U54,U64,U74,U84,U94,U104,U114,U124	74LVC1G32	Texas Instruments	SN74LVC1G32DBVR	SOT23-5			
29	8	U55,U65,U75,U85,U95,U105,U115,U125	74LVC2G07	Texas Instruments	SN74LVC2G07DBVR	SOT23-6			
30	8	U56,U66,U76,U88,U96,U108,U116,U126	74LV4066	Texas Instruments	SN74LV4066APWR	TSSOP14			
31	7	U57,U67,U77,U86,U106,U117,U127	74LVC1G125	Texas Instruments	SN74LVC1G125DBVR	SOT23-5			
32	6	U58,U68,U78,U98,U107,U118	74LVC1G11	Texas Instruments	SN74LVC1G11DBVR	SOT23-6			
33	3	U87,U128,U129	74LVC1G08	Texas Instruments	SN74LVC1G08DBVR	SOT23-5			
34	1	U93	74LVC1G74	NXP	74LVC1G74DC	TSSOP8			
35	1	U130	TPS3823	Texas Instruments	TPS3823-33DBVT	SOT23-5			

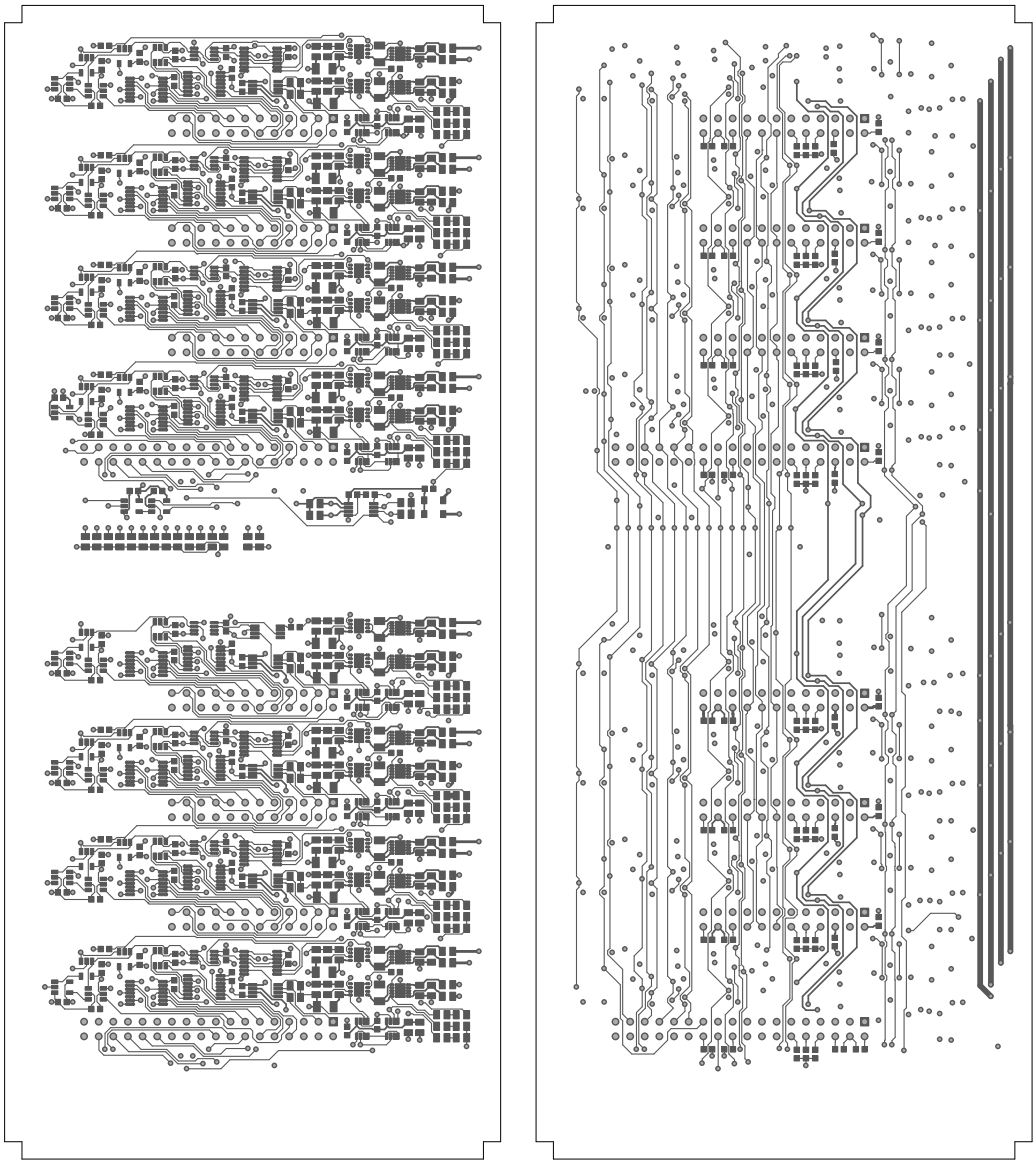
Figure A.3: Backplane Bill of Materials (print-out)

A.3 Gerber Files

Rendered images of the NUTS Backplane gerber files are presented in figures A.4 and A.5. The layer stack-up, from top to bottom is:

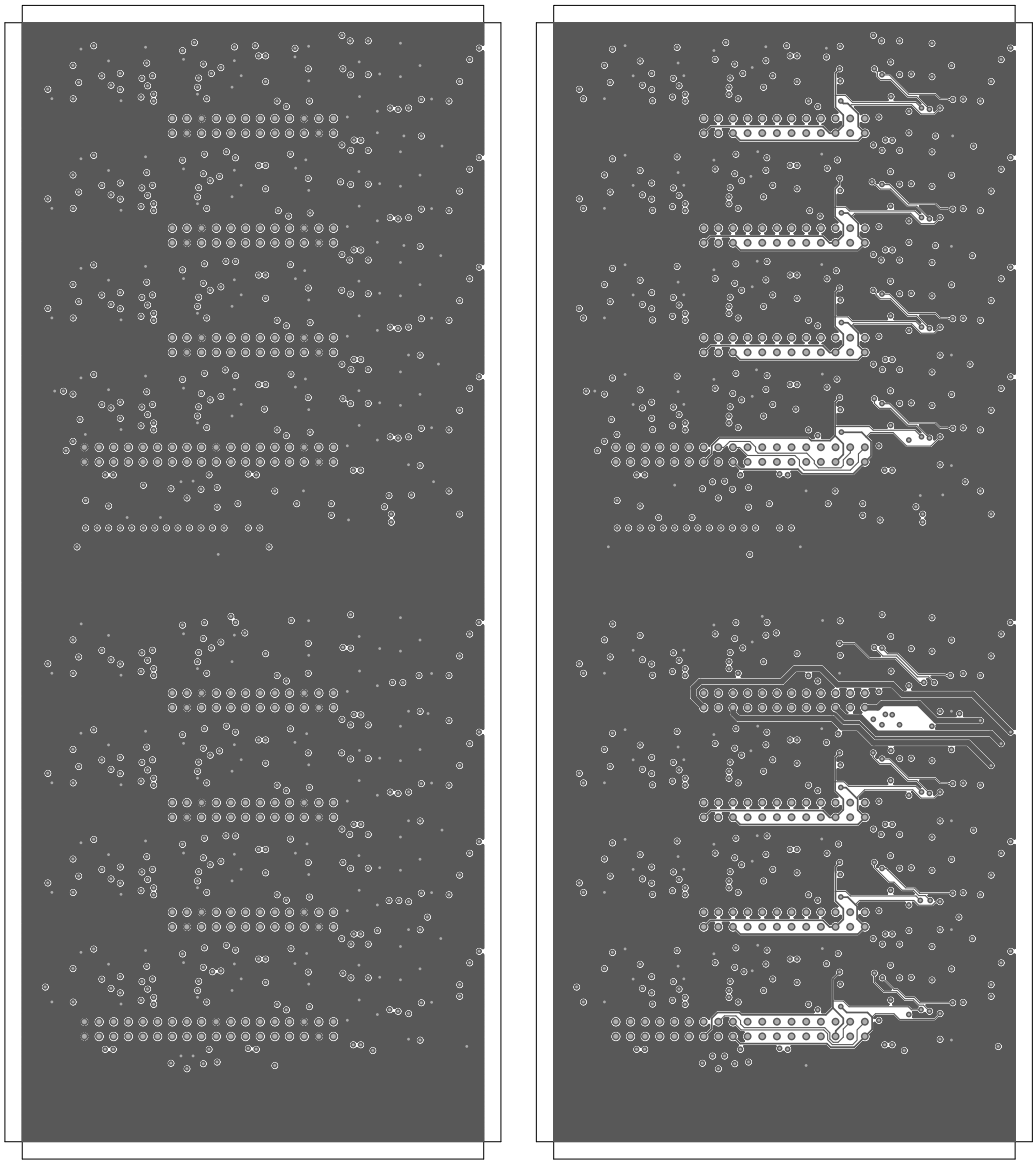
Layer no.	Name	Rendered image	Description
1	Top	Figure A.4(a)	Components & signals
2	Gnd	Figure A.5(a)	Ground plane
3	Pwr	Figure A.5(b)	Power plane + power signals
4	Bottom	Figure A.4(b)	Signals

Full Gerber RS274X files of the backplane PCB used for production can be found on the accompanying CD-ROM. The format is: metric 4.5, leading zero suppression.



(a) TOP Layer (b) BOTTOM Layer

Figure A.4: Backplane TOP & BOTTOM layers



(a) GND Layer (Layer 2)

(b) PWR Layer (Layer 3)

Figure A.5: Backplane GND & PWR layers

Appendix B

Schematics

B.1 Backplane Schematics

The NUTS Backplane schematic consists of 11 pages, including a titlepage. The version presented here is the actual version used to produce the initial prototype, and does not contain any changes or fixes done since.

B.2 BCR Prototype Schematics

Following the Backplane schematics, the schematic drawings for the BCR prototype are given.

NTNU

Det skapende universitet

NUTS Backplane

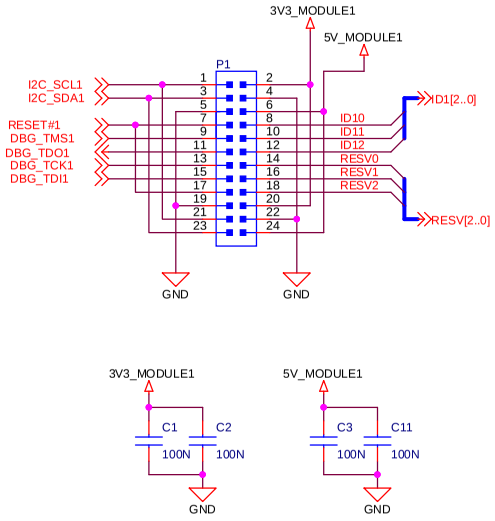
Revision History

Revision	Comment
PA00	Initial Pre-Release Version
PA01	Fixed errors from initial review
A00	First Release Version
A01	Changed current limit set resistor values.

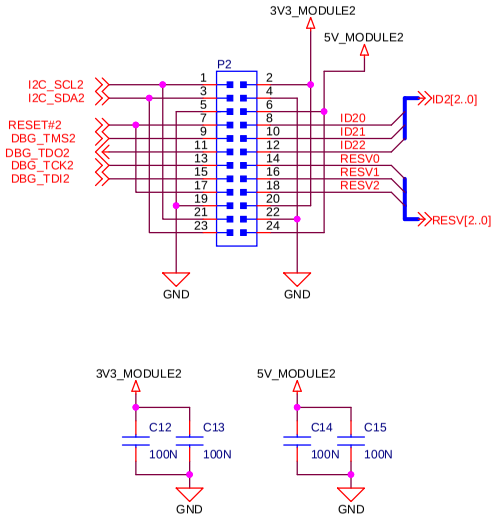
/
TOP

		Schematic Title: **NUTS Backplane**	
		Page Title: **Front Page**	
Designed: <OrgAddr1>	Approved: <OrgAddr2>	Document number: **<Doc>**	Revision: A01
Size: A3	BOM Doc No: <Cage Code>		
Design Created Date: Friday, February 18, 2011		Sheet Created Date: Friday, February 18, 2011	Sheet Modified Date: Tuesday, March 29, 2011	Sheet: 1 of 11

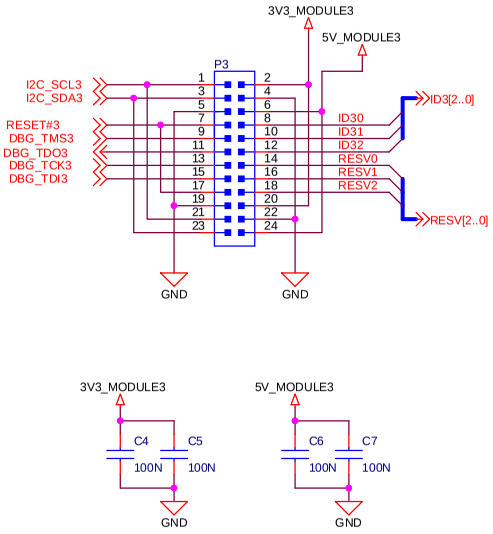
Module Connector 1



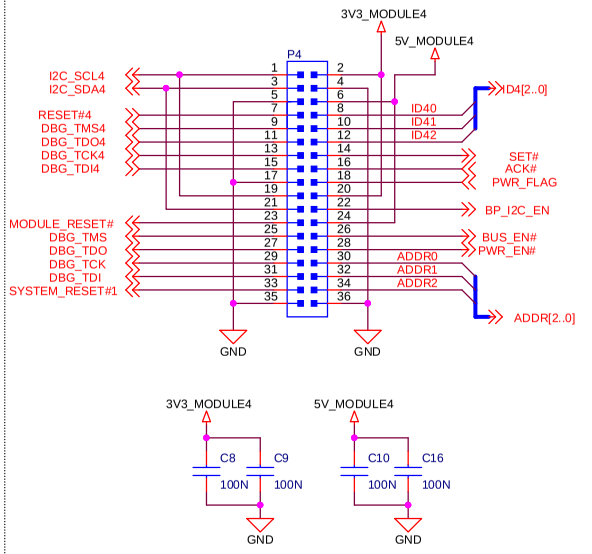
Module Connector 2



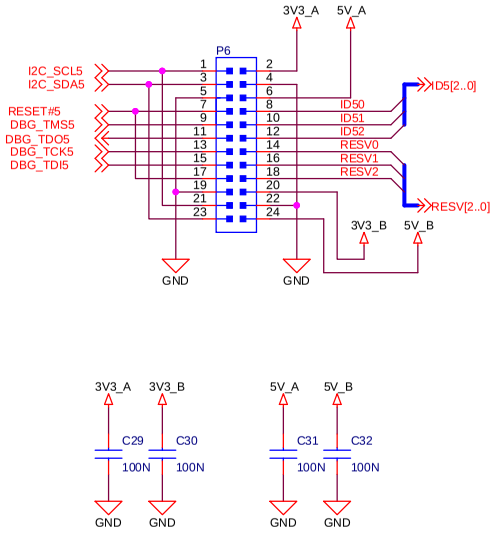
Module Connector 3



Module Connector 4
On Board Data Handling

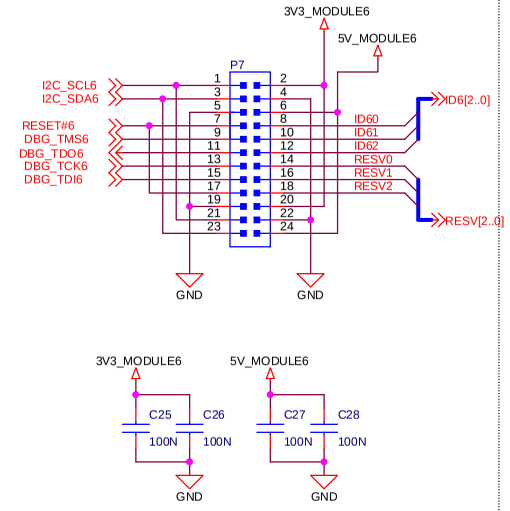


Module Connector 5
Electrical Power System

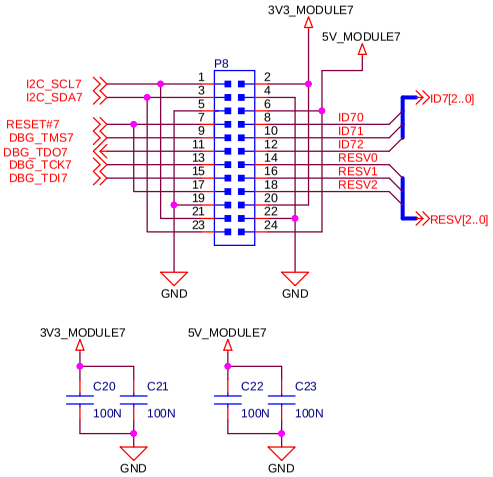


The EPS supplies the 3V3_A, 3V3_B, 5V_A and 5V_B rails. The module 5 power distribution circuit is used to provide protection to the backplane. Module 5 power cannot be turned off.

Module Connector 6

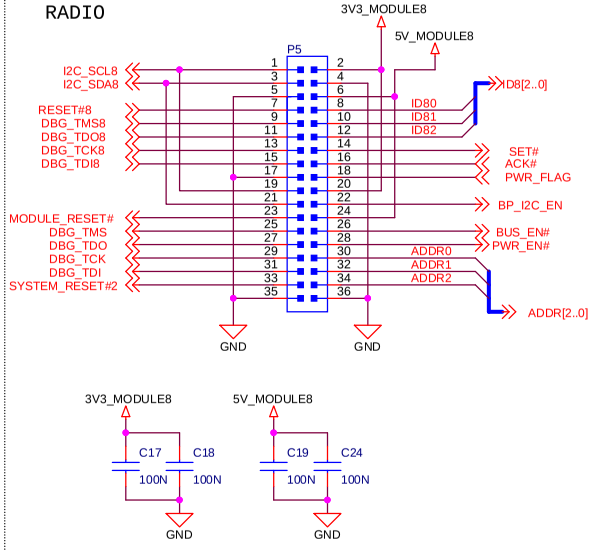


Module Connector 7

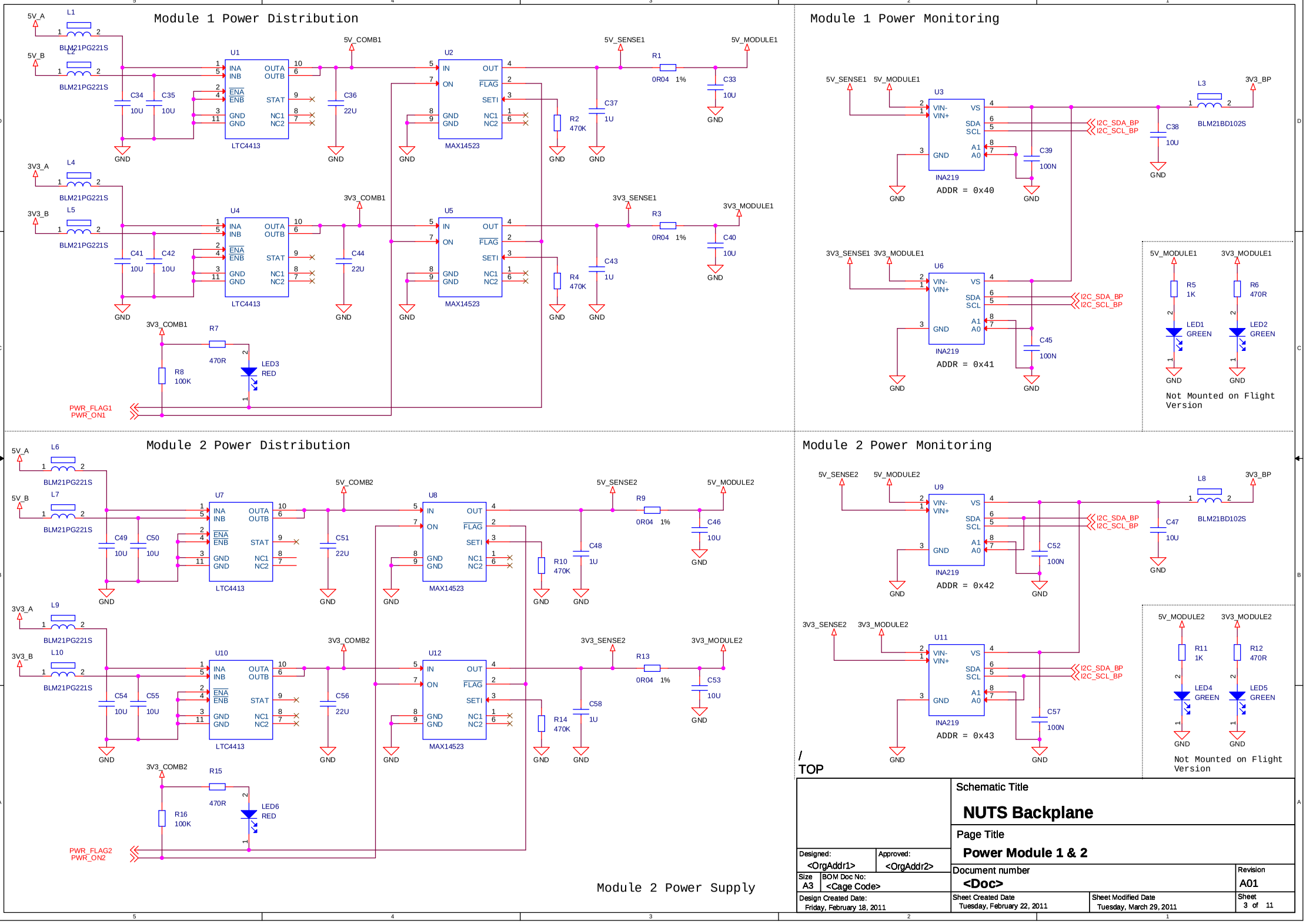


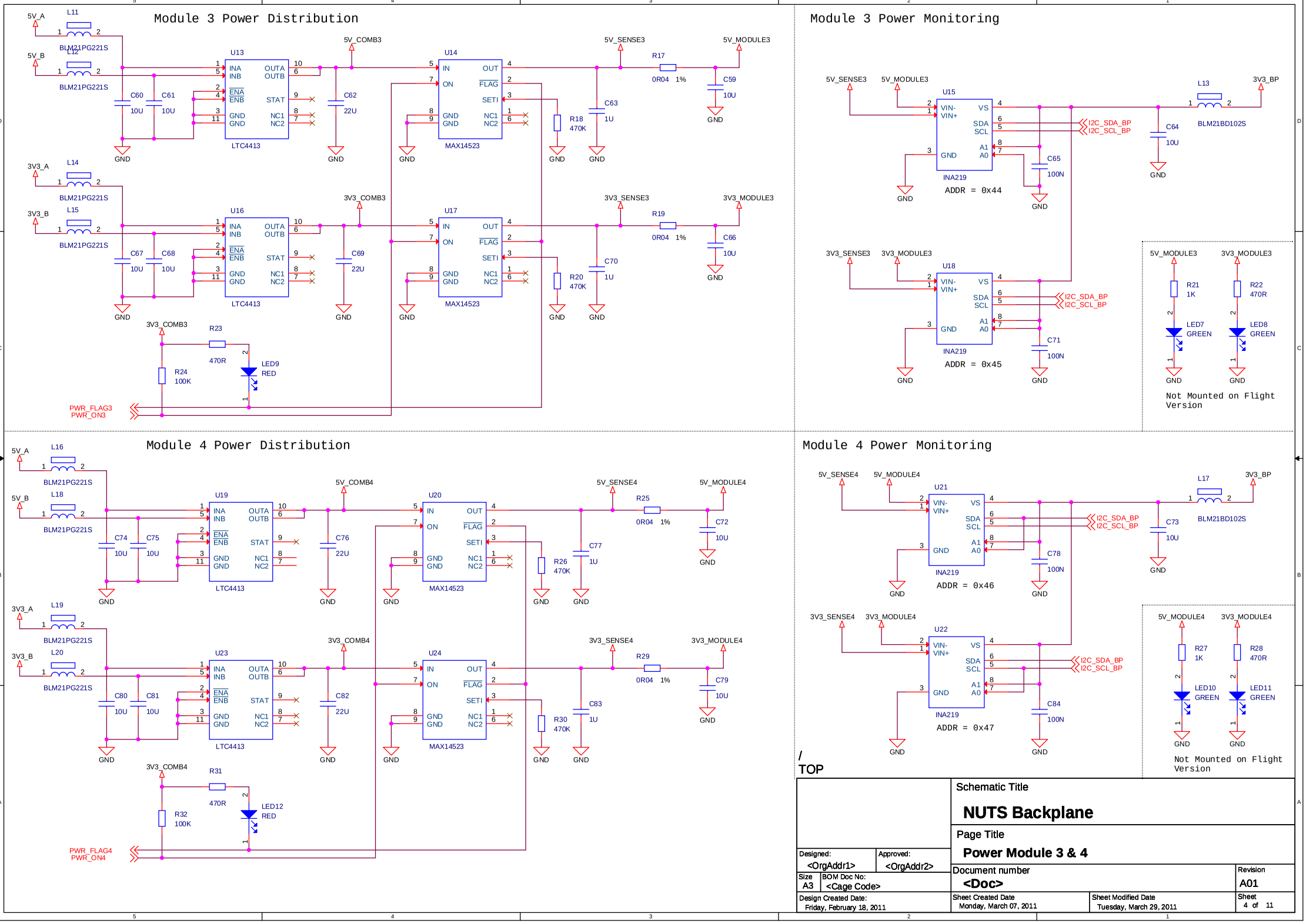
/
TOP

Module Connector 8
RADIO



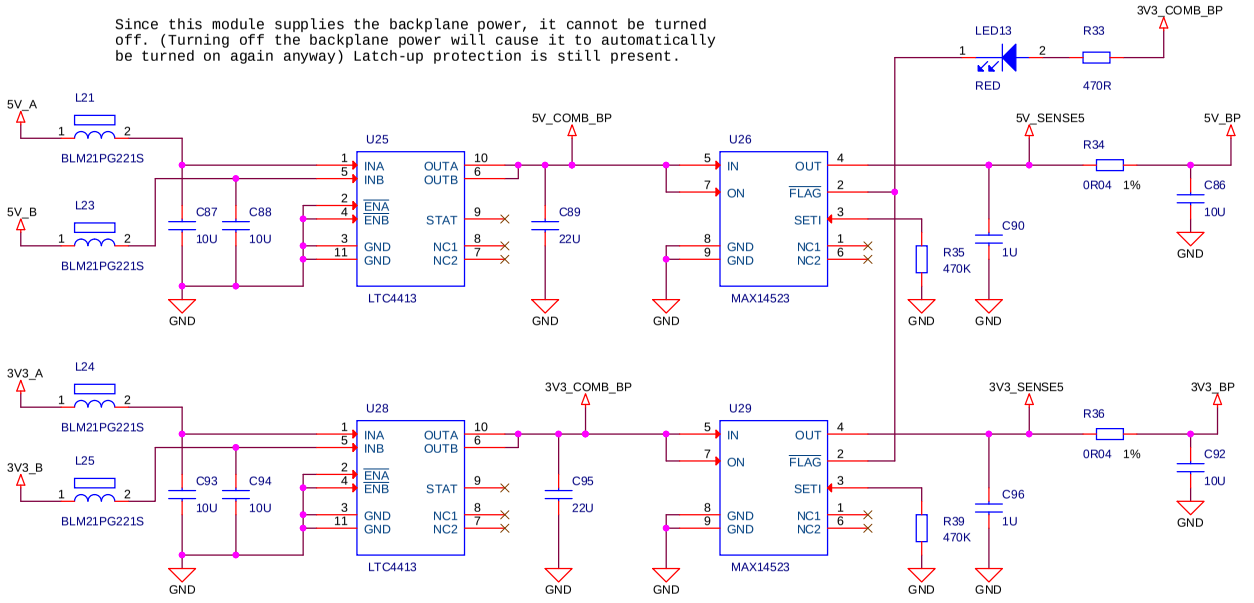
Schematic Title	**NUTS Backplane**
Page Title	**Module Connectors**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number **<Doc>** / Revision A01
Size A3 / BOM Doc No: <Cage Code>	
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Friday, February 18, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet 2 of 11


Module 1 Power Distribution
5V_A
5V_B
L1
L2
BLM21PG221S
U1
LTC4413
INA
INB
OUTA
OUTB
ENA
ENB
STAT
GND
NC1
NC2
C34
C35
10U
C36
22U
5V_COMB1
U2
MAX14523
IN
OUT
ON
FLAG
SETI
R2
470K
C37
1U
5V_SENSE1
R1
0R04 1%
5V_MODULE1
C33
3V3_A
3V3_B
L4
L5
C41
C42
C44
3V3_COMB1
U4
U5
R4
C43
3V3_SENSE1
R3
C40
3V3_MODULE1
3V3_COMB1
R7
470R
LED3
RED
R8
100K
PWR_FLAG1
PWR_ON1
Module 1 Power Monitoring
5V_SENSE1
5V_MODULE1
U3
INA219
VIN-
VIN+
VS
SDA
SCL
A1
A0
ADDR = 0x40
I2C_SDA_BP
I2C_SCL_BP
L3
3V3_BP
BLM21BD102S
C38
C39
100N
3V3_SENSE1
3V3_MODULE1
U6
ADDR = 0x41
C45
5V_MODULE1
3V3_MODULE1
R5
1K
R6
470R
LED1
GREEN
LED2
GREEN
Not Mounted on Flight Version
Module 2 Power Distribution
5V_A
5V_B
L6
L7
U7
U8
5V_COMB2
5V_SENSE2
5V_MODULE2
R9
C46
C49
C50
C51
R10
C48
3V3_A
3V3_B
L9
L10
U10
U12
3V3_COMB2
3V3_SENSE2
3V3_MODULE2
R13
C53
C54
C55
C56
R14
C58
3V3_COMB2
R15
470R
LED6
RED
R16
100K
PWR_FLAG2
PWR_ON2
Module 2 Power Supply
Module 2 Power Monitoring
5V_SENSE2
5V_MODULE2
U9
ADDR = 0x42
L8
C47
C52
3V3_SENSE2
3V3_MODULE2
U11
ADDR = 0x43
C57
5V_MODULE2
3V3_MODULE2
R11
1K
R12
470R
LED4
GREEN
LED5
GREEN
Not Mounted on Flight Version
/
TOP
Schematic Title
NUTS Backplane
Page Title
Power Module 1 & 2
Designed:
<OrgAddr1>
Approved:
<OrgAddr2>
Size
A3
BOM Doc No:
<Cage Code>
Document number
<Doc>
Revision
A01
Design Created Date:
Friday, February 18, 2011
Sheet Created Date
Tuesday, February 22, 2011
Sheet Modified Date
Tuesday, March 29, 2011
Sheet
3 of 11

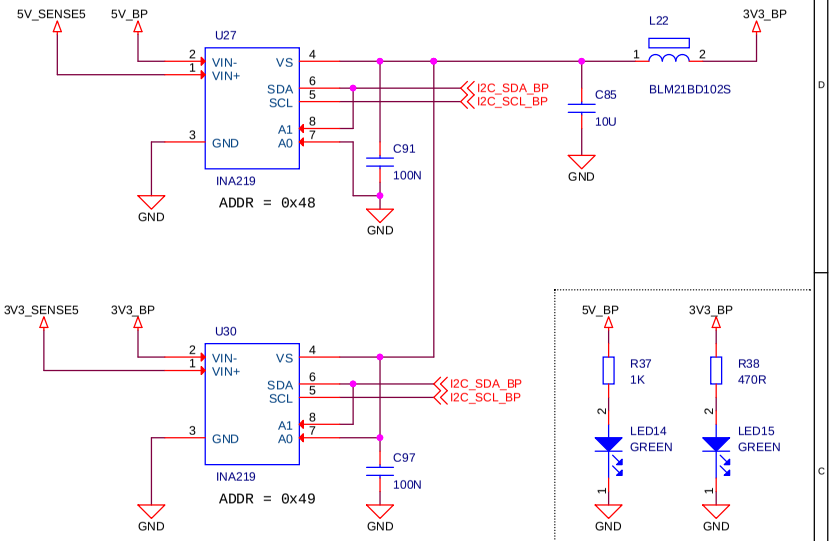

Module 3 Power Distribution
5V_A
5V_B
L11
L12
BLM21PG221S
U13
LTC4413
INA
INB
OUTA
OUTB
ENA
ENB
STAT
GND
NC1
NC2
C60
C61
10U
C62
22U
5V_COMB3
U14
MAX14523
IN
ON
OUT
FLAG
SETI
R18
470K
C63
1U
5V_SENSE3
R17
0R04 1%
5V_MODULE3
C59
3V3_A
3V3_B
L14
L15
U16
U17
C67
C68
C69
3V3_COMB3
R19
R20
C70
3V3_SENSE3
C66
3V3_MODULE3
3V3_COMB3
R23
470R
LED9
RED
R24
100K
PWR_FLAG3
PWR_ON3
Module 3 Power Monitoring
5V_SENSE3
5V_MODULE3
U15
INA219
VIN-
VIN+
VS
SDA
SCL
A1
A0
ADDR = 0x44
I2C_SDA_BP
I2C_SCL_BP
L13
3V3_BP
BLM21BD102S
C64
C65
100N
3V3_SENSE3
3V3_MODULE3
U18
ADDR = 0x45
C71
5V_MODULE3
3V3_MODULE3
R21
1K
R22
470R
LED7
GREEN
LED8
GREEN
Not Mounted on Flight Version
Module 4 Power Distribution
5V_A
5V_B
L16
L18
U19
U20
C74
C75
C76
5V_COMB4
R25
R26
C77
5V_SENSE4
C72
5V_MODULE4
3V3_A
3V3_B
L19
L20
U23
U24
C80
C81
C82
3V3_COMB4
R29
R30
C83
3V3_SENSE4
C79
3V3_MODULE4
3V3_COMB4
R31
470R
LED12
RED
R32
100K
PWR_FLAG4
PWR_ON4
Module 4 Power Monitoring
5V_SENSE4
5V_MODULE4
U21
ADDR = 0x46
L17
3V3_BP
C73
C78
100N
3V3_SENSE4
3V3_MODULE4
U22
ADDR = 0x47
C84
5V_MODULE4
3V3_MODULE4
R27
1K
R28
470R
LED10
GREEN
LED11
GREEN
Not Mounted on Flight Version
/
TOP
Schematic Title
NUTS Backplane
Page Title
Power Module 3 & 4
Designed:
<OrgAddr1>
Approved:
<OrgAddr2>
Size
A3
BOM Doc No:
<Cage Code>
Document number
<Doc>
Revision
A01
Design Created Date:
Friday, February 18, 2011
Sheet Created Date
Monday, March 07, 2011
Sheet Modified Date
Tuesday, March 29, 2011
Sheet
4 of 11

Backplane Power Distribution

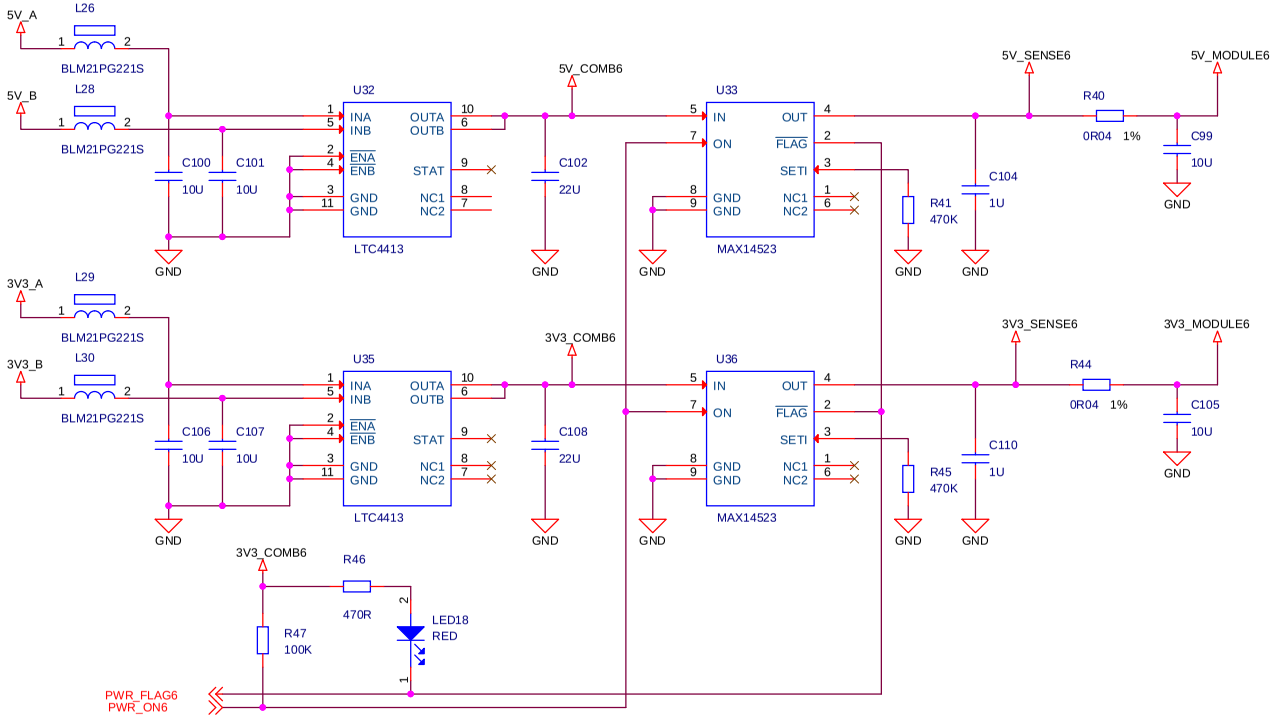
Since this module supplies the backplane power, it cannot be turned off. (Turning off the backplane power will cause it to automatically be turned on again anyway) Latch-up protection is still present.



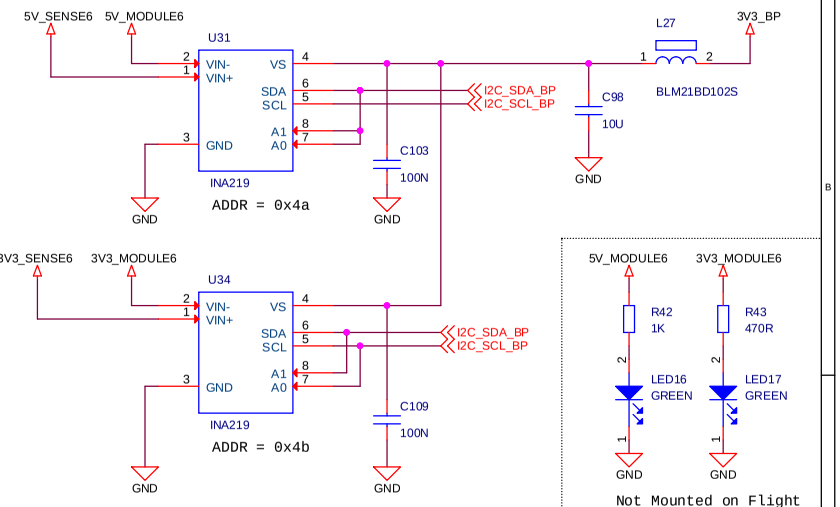
Backplane Power Monitoring



Module 6 Power Distribution

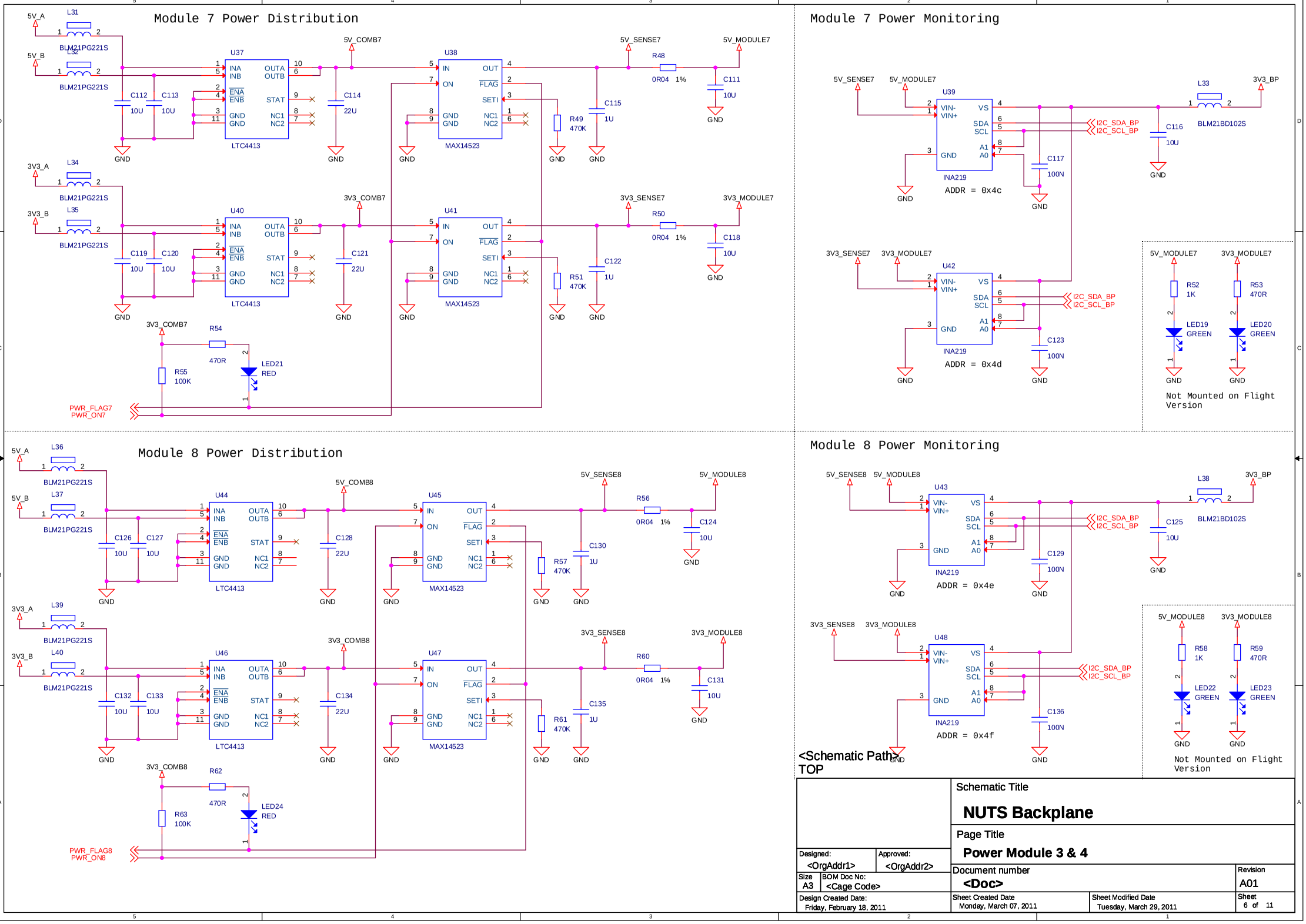


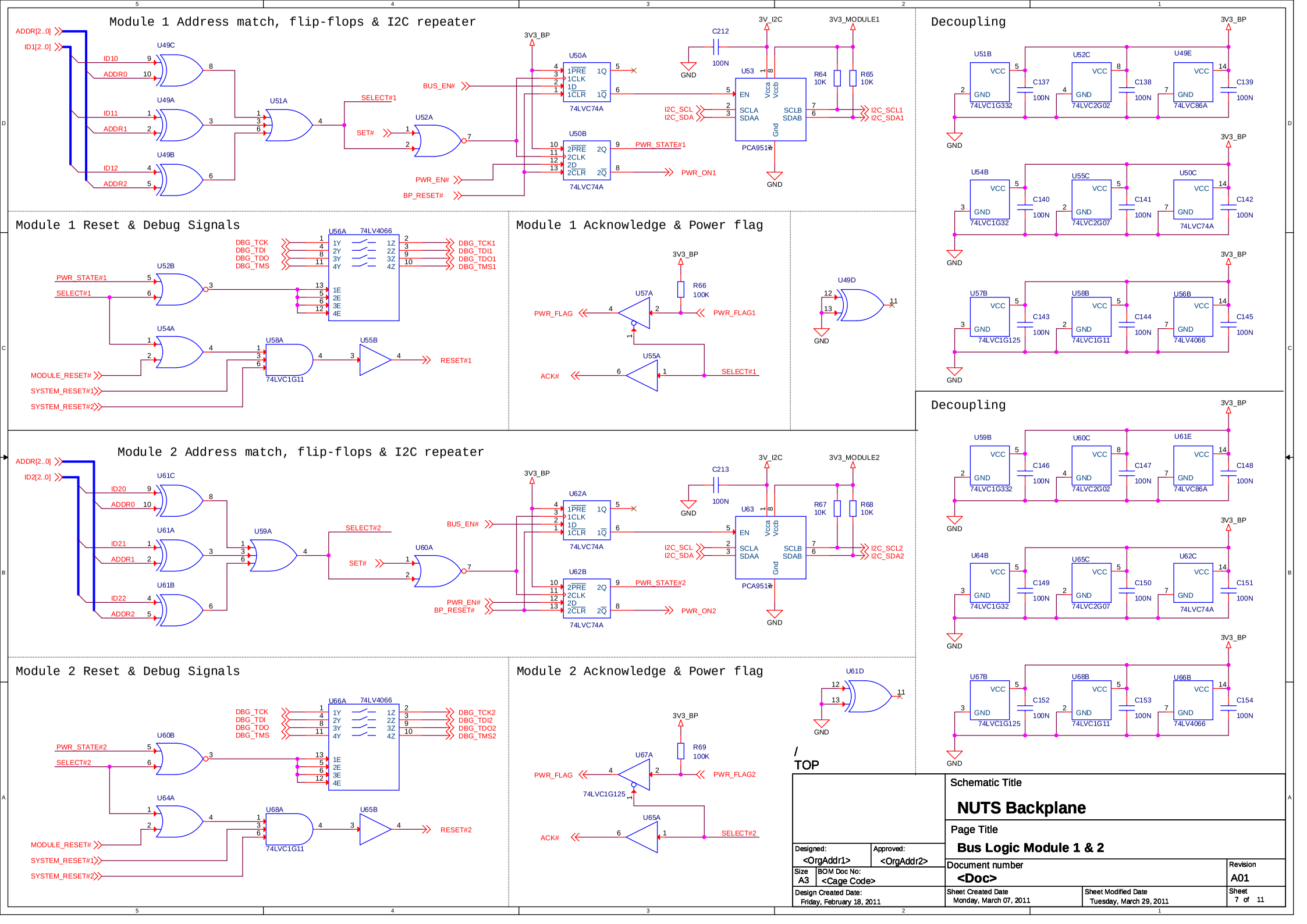
Module 6 Power Monitoring



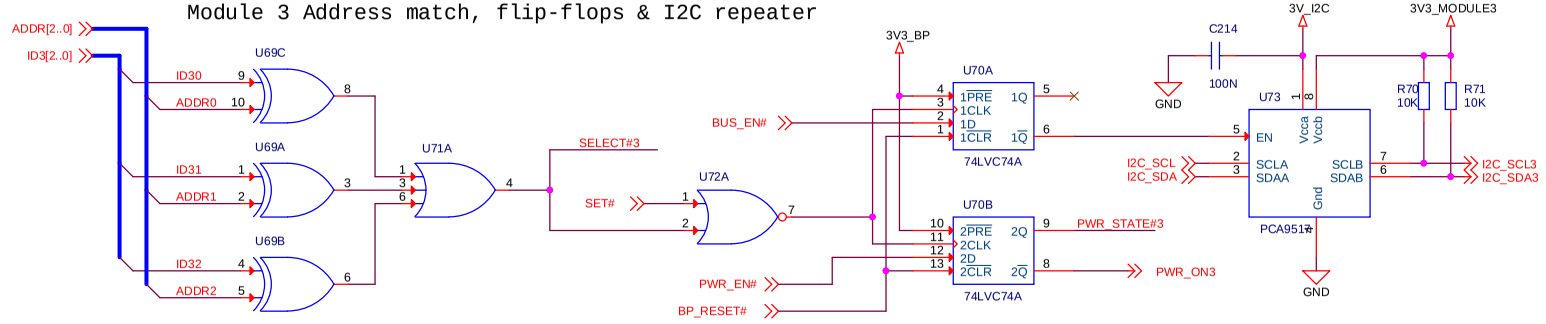
/
TOP

Schematic Title	**NUTS Backplane**
Page Title	**Power Module 3 & 4**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number **<Doc>** / Revision **A01**
Size A3 / BOM Doc No: <Cage Code>	
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Monday, March 07, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet 5 of 11

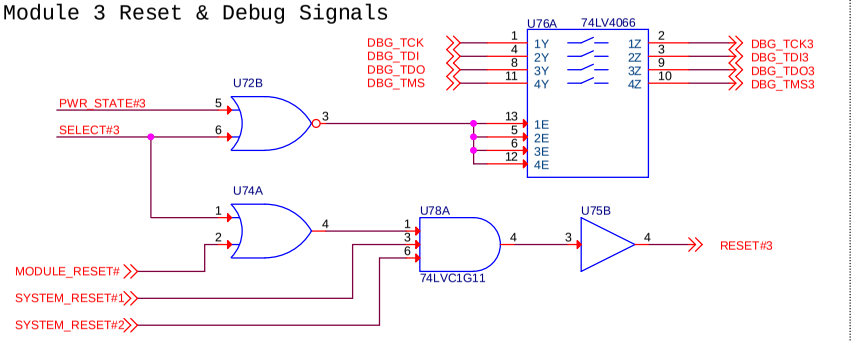

Module 7 Power Distribution
5V_A
5V_B
L31
L32
BLM21PG221S
U37
LTC4413
INA
INB
OUTA
OUTB
ENA
ENB
STAT
GND
NC1
NC2
C112
C113
10U
C114
22U
5V_COMB7
U38
MAX14523
IN
OUT
ON
FLAG
SETI
C115
1U
R49
470K
R48
0R04 1%
5V_SENSE7
5V_MODULE7
C111
3V3_A
3V3_B
L34
L35
U40
U41
C119
C120
C121
C122
3V3_COMB7
R50
3V3_SENSE7
3V3_MODULE7
C118
R51
R54
470R
LED21
RED
R55
100K
PWR_FLAG7
PWR_ON7
Module 8 Power Distribution
L36
L37
U44
U45
C126
C127
C128
5V_COMB8
R56
5V_SENSE8
5V_MODULE8
C124
R57
C130
L39
L40
U46
U47
C132
C133
C134
3V3_COMB8
R60
3V3_SENSE8
3V3_MODULE8
C131
R61
C135
R62
LED24
R63
PWR_FLAG8
PWR_ON8
Module 7 Power Monitoring
U39
INA219
VIN-
VIN+
VS
SDA
SCL
A1
A0
ADDR = 0x4c
I2C_SDA_BP
I2C_SCL_BP
L33
3V3_BP
BLM21BD102S
C116
C117
100N
U42
ADDR = 0x4d
C123
R52
1K
R53
470R
LED19
GREEN
LED20
Not Mounted on Flight Version
Module 8 Power Monitoring
U43
ADDR = 0x4e
L38
C125
C129
U48
ADDR = 0x4f
C136
R58
R59
LED22
LED23
<Schematic Path>
TOP
Schematic Title
NUTS Backplane
Page Title
Power Module 3 & 4
Designed:
<OrgAddr1>
Approved:
<OrgAddr2>
Size
A3
BOM Doc No:
<Cage Code>
Document number
<Doc>
Revision
A01
Design Created Date:
Friday, February 18, 2011
Sheet Created Date
Monday, March 07, 2011
Sheet Modified Date
Tuesday, March 29, 2011
Sheet
6 of 11


Module 1 Address match, flip-flops & I2C repeater
ADDR[2..0]
ID1[2..0]
ID10
ADDR0
ID11
ADDR1
ID12
ADDR2
U49C
U49A
U49B
U51A
SELECT#1
SET#
U52A
BUS_EN#
PWR_EN#
BP_RESET#
3V3_BP
U50A
U50B
74LVC74A
PWR_STATE#1
PWR_ON1
C212
100N
GND
3V_I2C
3V3_MODULE1
U53
PCA9517
EN
SCLA
SDAA
SCLB
SDAB
Vcca
Vccb
Gnd
I2C_SCL
I2C_SDA
R64
10K
R65
10K
I2C_SCL1
I2C_SDA1
Module 1 Reset & Debug Signals
U56A
74LV4066
DBG_TCK
DBG_TDI
DBG_TDO
DBG_TMS
DBG_TCK1
DBG_TDI1
DBG_TDO1
DBG_TMS1
U52B
PWR_STATE#1
SELECT#1
U54A
U58A
U55B
74LVC1G11
RESET#1
MODULE_RESET#
SYSTEM_RESET#1
SYSTEM_RESET#2
Module 1 Acknowledge & Power flag
U57A
R66
100K
PWR_FLAG
PWR_FLAG1
U55A
ACK#
SELECT#1
U49D
Decoupling
U51B
U52C
U49E
U54B
U55C
U50C
U57B
U58B
U56B
VCC
GND
C137
C138
C139
C140
C141
C142
C143
C144
C145
100N
74LVC1G332
74LVC2G02
74LVC86A
74LVC1G32
74LVC2G07
74LVC74A
74LVC1G125
74LVC1G11
74LV4066
Module 2 Address match, flip-flops & I2C repeater
ADDR[2..0]
ID2[2..0]
ID20
ADDR0
ID21
ADDR1
ID22
ADDR2
U61C
U61A
U61B
U59A
SELECT#2
SET#
U60A
BUS_EN#
PWR_EN#
BP_RESET#
3V3_BP
U62A
U62B
74LVC74A
PWR_STATE#2
PWR_ON2
C213
100N
GND
3V_I2C
3V3_MODULE2
U63
PCA9517
I2C_SCL
I2C_SDA
R67
10K
R68
10K
I2C_SCL2
I2C_SDA2
Module 2 Reset & Debug Signals
U66A
74LV4066
DBG_TCK
DBG_TDI
DBG_TDO
DBG_TMS
DBG_TCK2
DBG_TDI2
DBG_TDO2
DBG_TMS2
U60B
PWR_STATE#2
SELECT#2
U64A
U68A
U65B
74LVC1G11
RESET#2
MODULE_RESET#
SYSTEM_RESET#1
SYSTEM_RESET#2
Module 2 Acknowledge & Power flag
U67A
R69
100K
PWR_FLAG
PWR_FLAG2
74LVC1G125
U65A
ACK#
SELECT#2
U61D
Decoupling
U59B
U60C
U61E
U64B
U65C
U62C
U67B
U68B
U66B
VCC
GND
C146
C147
C148
C149
C150
C151
C152
C153
C154
100N
74LVC1G332
74LVC2G02
74LVC86A
74LVC1G32
74LVC2G07
74LVC74A
74LVC1G125
74LVC1G11
74LV4066
/
TOP
Schematic Title
NUTS Backplane
Page Title
Bus Logic Module 1 & 2
Designed:
<OrgAddr1>
Approved:
<OrgAddr2>
Document number
<Doc>
Revision
A01
Size
A3
BOM Doc No:
<Cage Code>
Design Created Date:
Friday, February 18, 2011
Sheet Created Date
Monday, March 07, 2011
Sheet Modified Date
Tuesday, March 29, 2011
Sheet
7 of 11

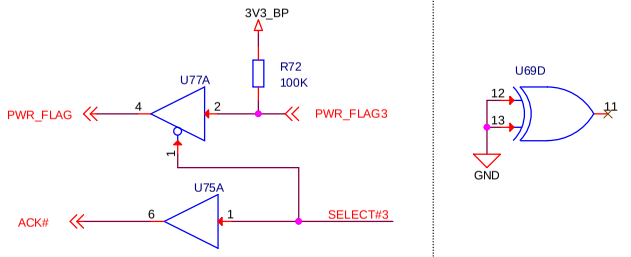
Module 3 Address match, flip-flops & I2C repeater



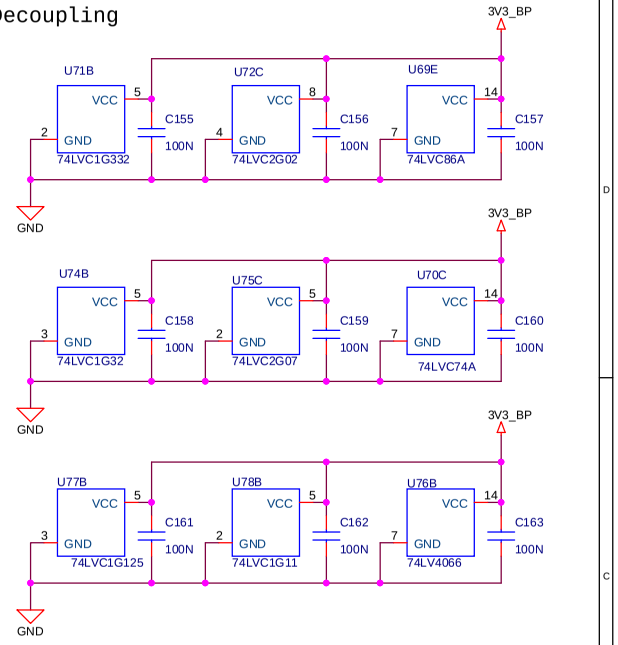
Module 3 Reset & Debug Signals



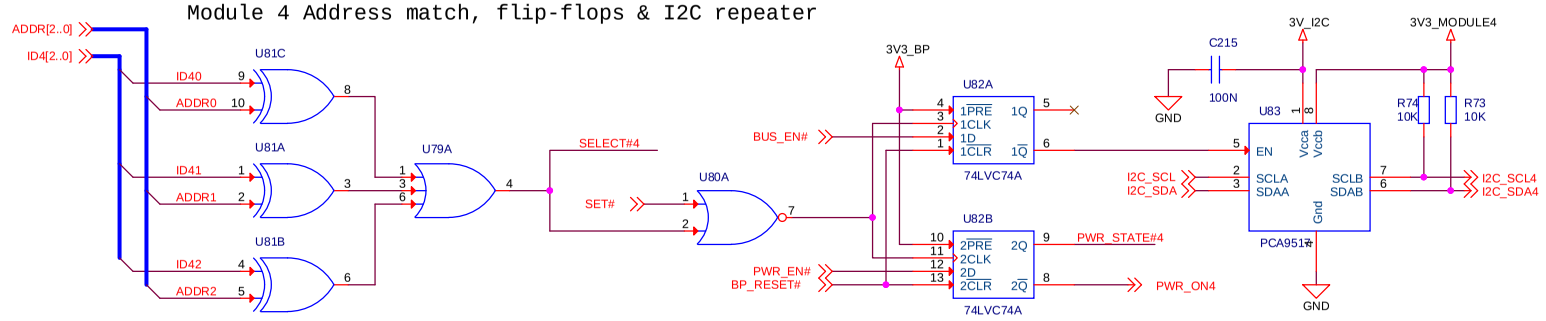
Module 3 Acknowledge & Power flag



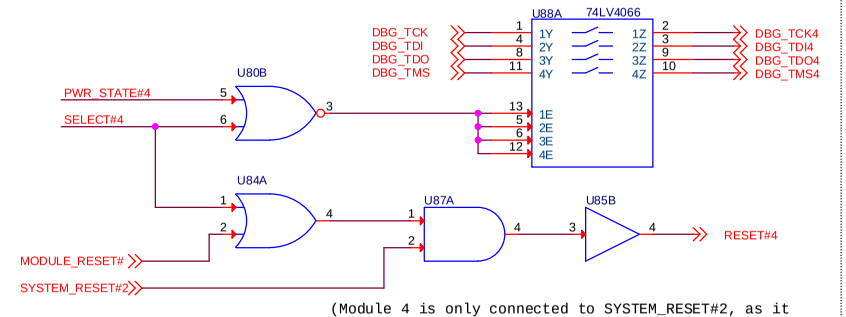
Decoupling



Module 4 Address match, flip-flops & I2C repeater

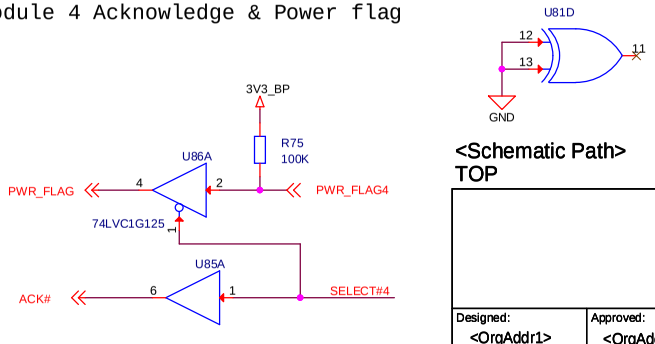


Module 4 Reset & Debug Signals



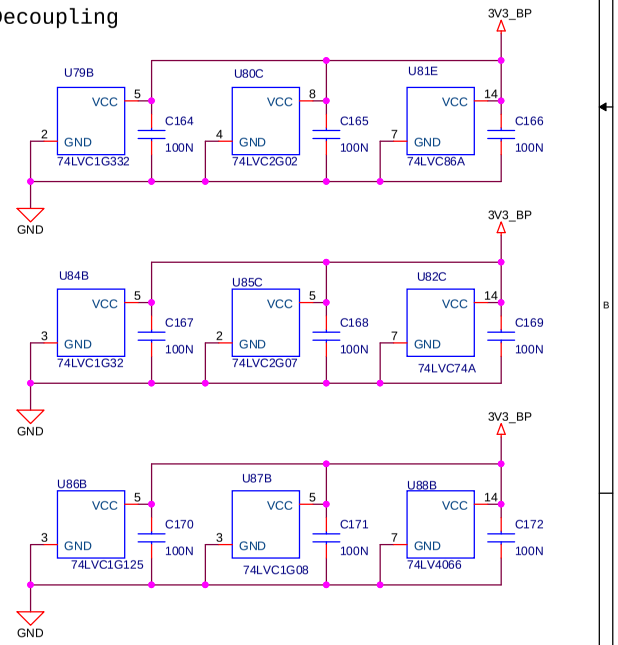
(Module 4 is only connected to SYSTEM_RESET#2, as it supplies the SYSTEM_RESET#1 signal, and connot be allowed to reset itself in this way).

Module 4 Acknowledge & Power flag



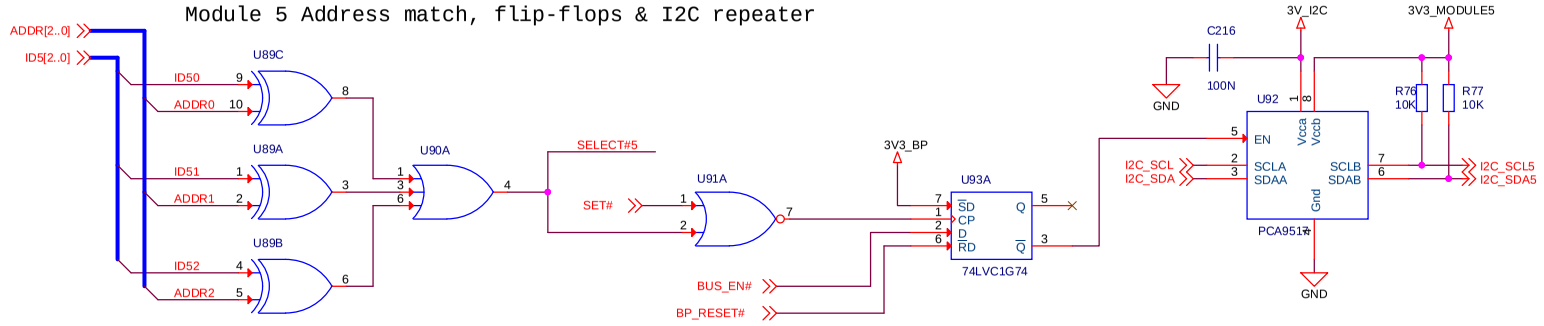
<Schematic Path>
TOP

Decoupling

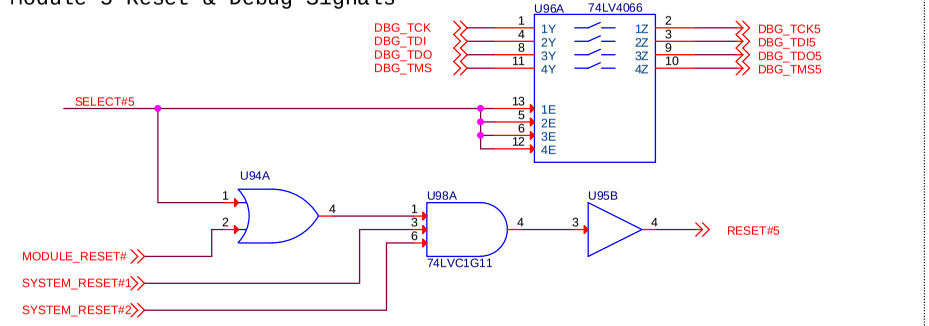


Schematic Title	**NUTS Backplane**
Page Title	**Bus Logic Module 1 & 2**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number: **<Doc>** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Monday, March 14, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet: 8 of 11

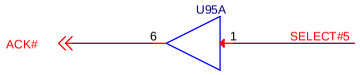
Module 5 Address match, flip-flops & I2C repeater



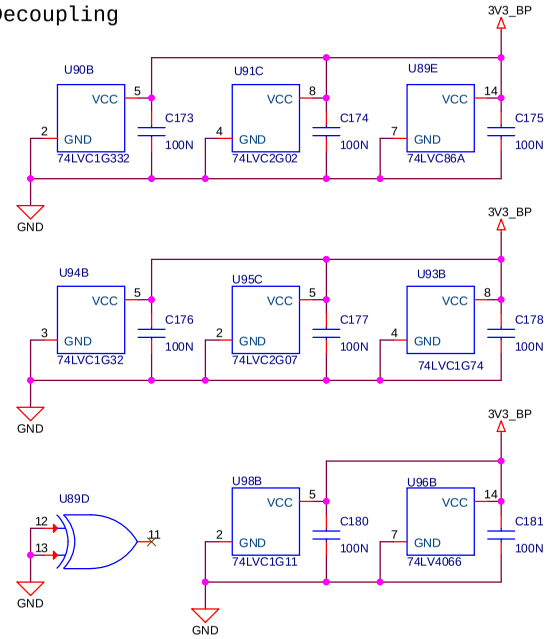
Module 5 Reset & Debug Signals



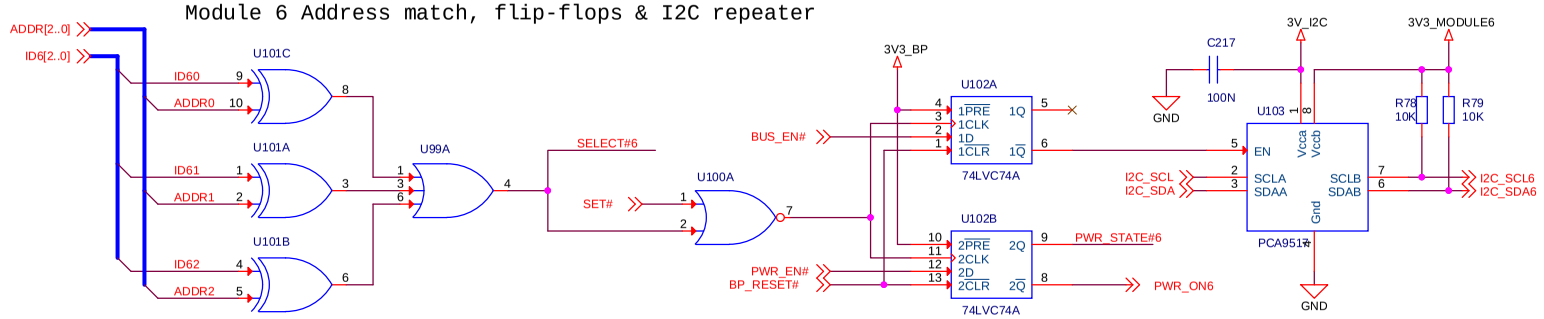
Module 5 Acknowledge



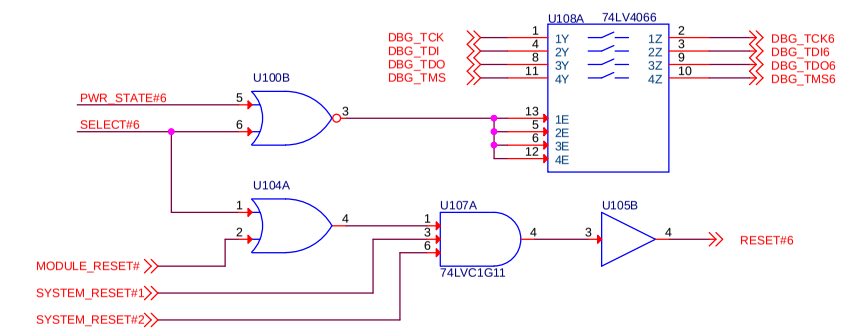
Decoupling



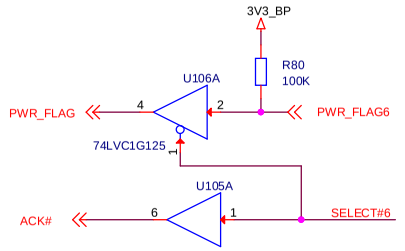
Module 6 Address match, flip-flops & I2C repeater

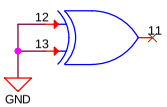


Module 6 Reset & Debug Signals



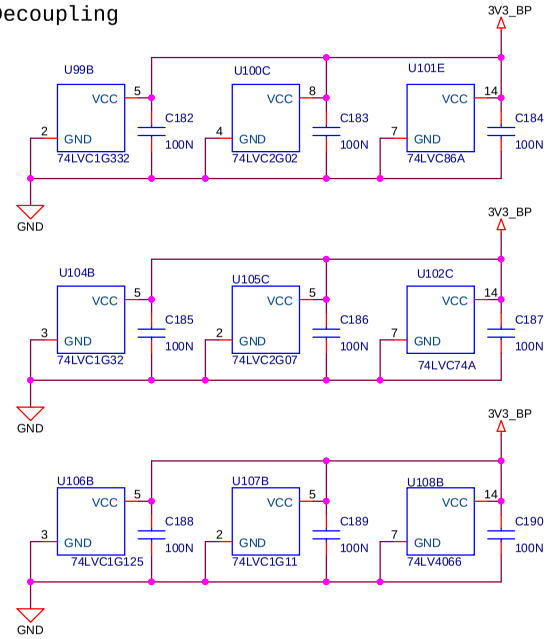
Module 6 Acknowledge & Power flag





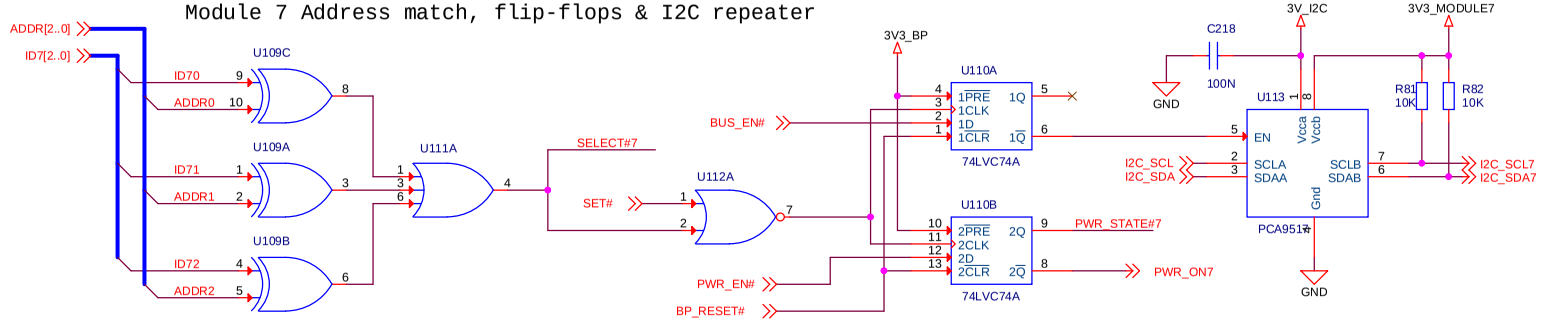
<Schematic Path>
TOP

Decoupling

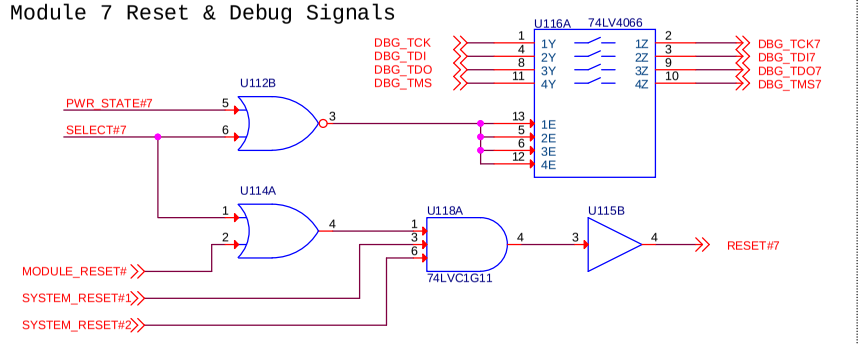


Schematic Title	**NUTS Backplane**
Page Title	**Bus Logic Module 1 & 2**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **<Doc>** / Revision: A01
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Tuesday, March 15, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet: 9 of 11

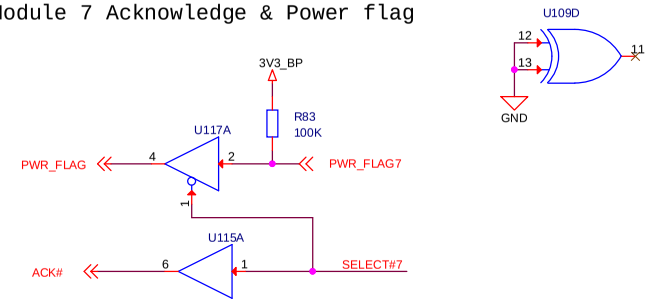
Module 7 Address match, flip-flops & I2C repeater



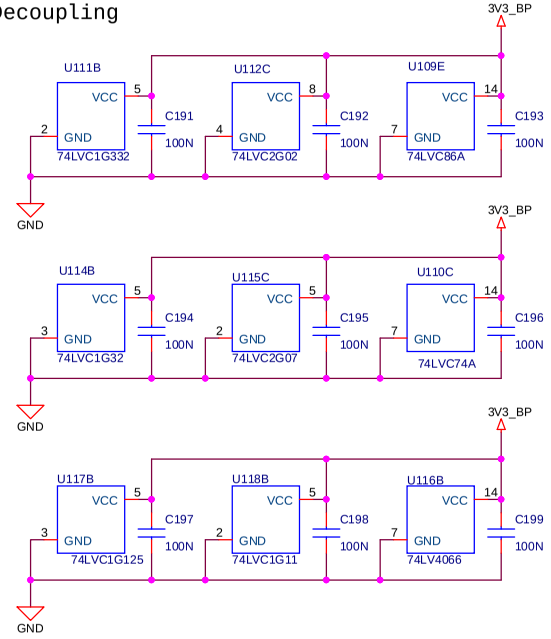
Module 7 Reset & Debug Signals



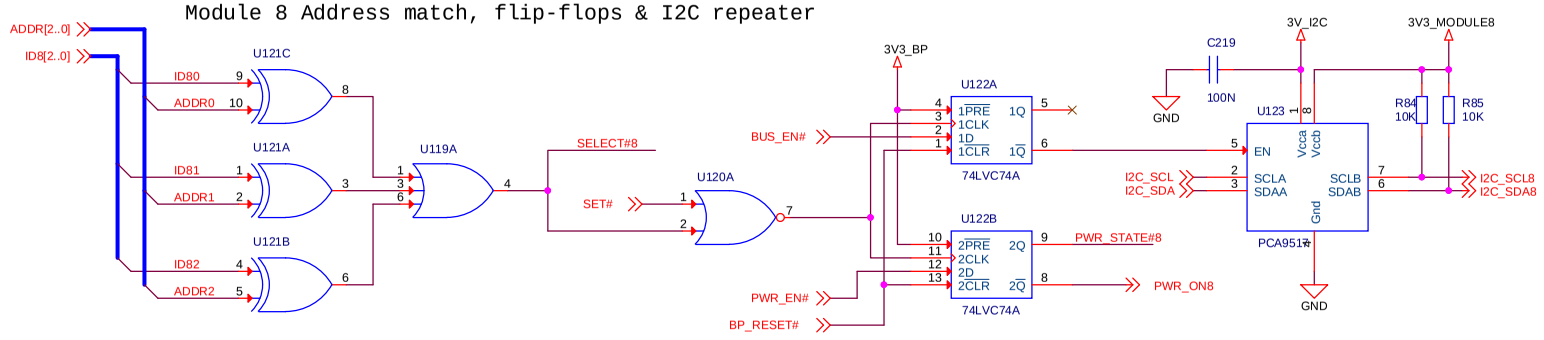
Module 7 Acknowledge & Power flag



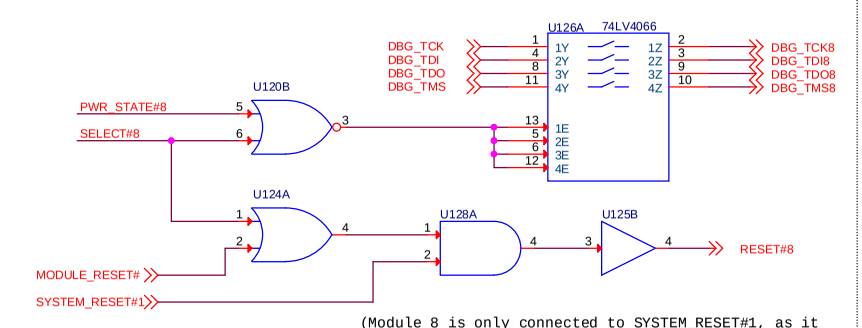
Decoupling



Module 8 Address match, flip-flops & I2C repeater

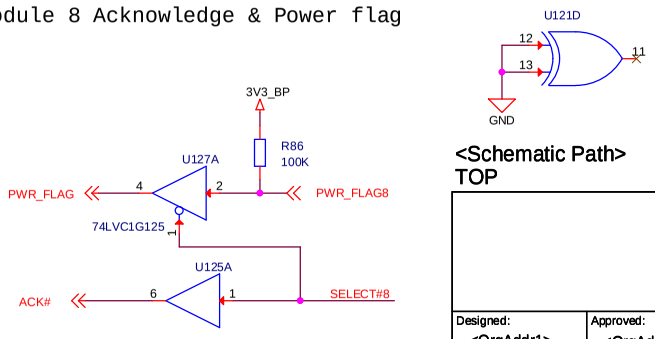


Module 8 Reset & Debug Signals

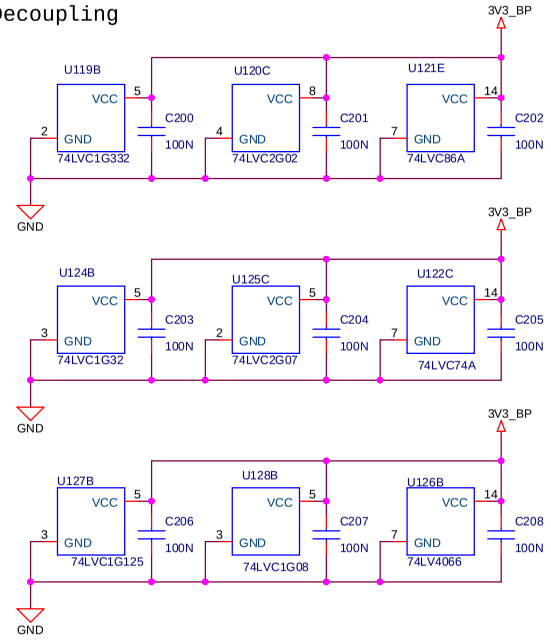


(Module 8 is only connected to SYSTEM_RESET#1, as it supplies the SYSTEM_RESET#2 signal, and connot be allowed to reset itself in this way).

Module 8 Acknowledge & Power flag



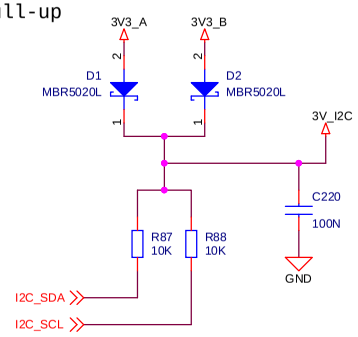
Decoupling



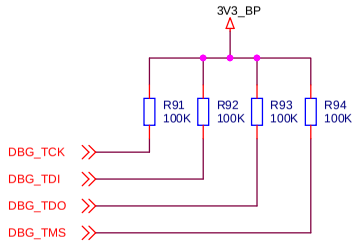
<Schematic Path>
TOP

Schematic Title	**NUTS Backplane**
Page Title	**Bus Logic Module 1 & 2**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number: **<Doc>** / Revision: **A01**
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Tuesday, March 15, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet: 10 of 11

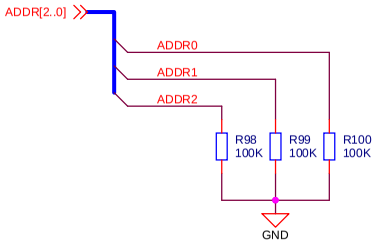
I2C Bus Pull-up



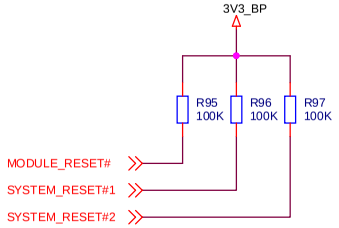
DEBUG Pull-up



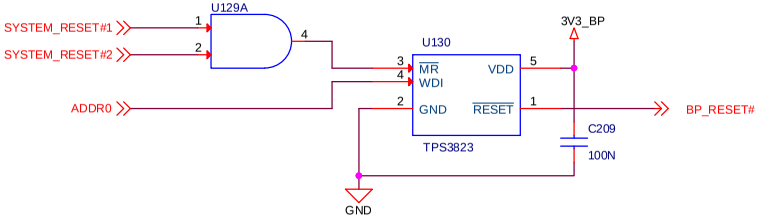
ADDR Pull-down

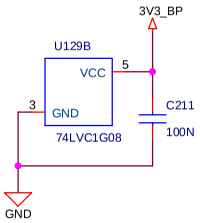


RESET Pull-up

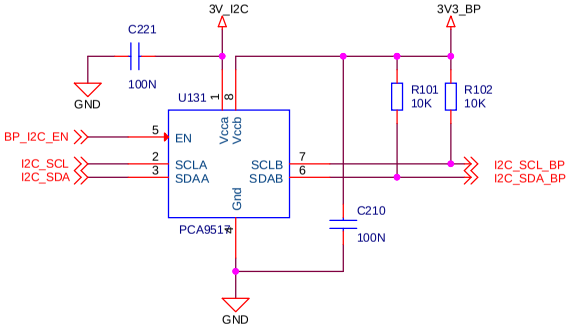


Backplane Reset Monitor

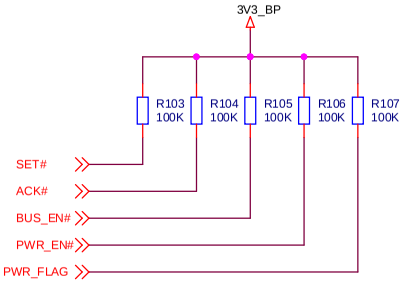


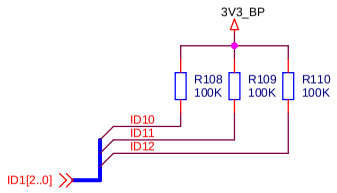


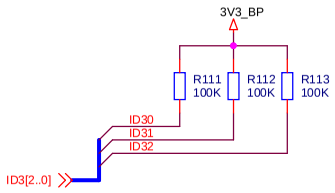
Backplane I2C Isolation

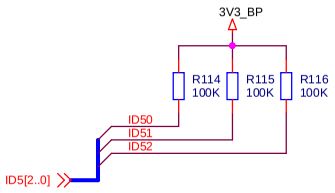


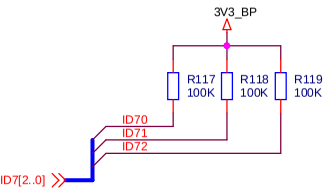
Control signals Pull-up

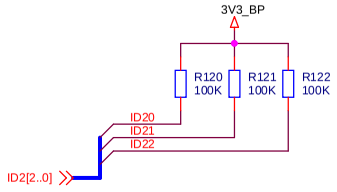


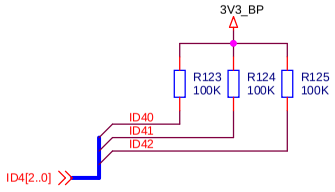


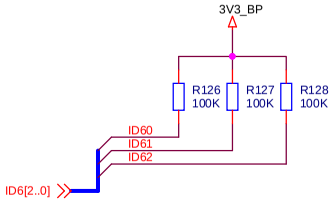


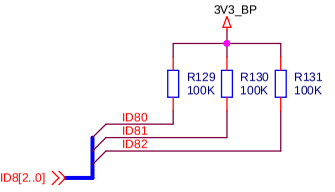












<Schematic Path>
TOP

Pull-ups on all ID lines

Schematic Title	**NUTS Backplane**
Page Title	**Backplane Logic & Pull-ups**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **<Doc>** / Revision: A01
Design Created Date: Friday, February 18, 2011	Sheet Created Date: Friday, March 11, 2011 / Sheet Modified Date: Tuesday, March 29, 2011 / Sheet: 11 of 11

NTNU

Det skapende universitet

NUTS BCR Prototype	
Board Function	**Page**
Front Page	1
Boost Regulator	2
Microcontroller	3
Analog & Power	4

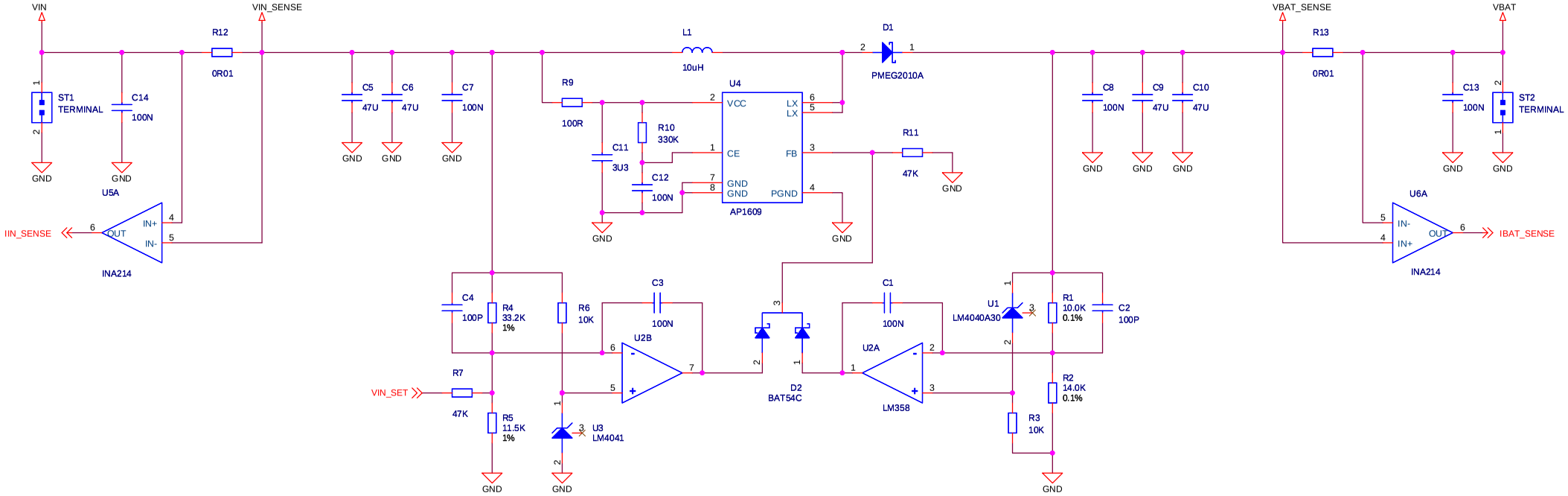
Revision History

Revision	Comment
PA00	Initial Pre-Release Version

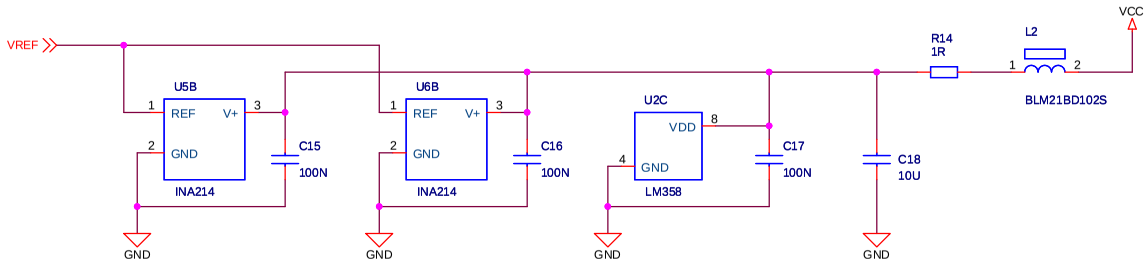
/
SCHEMATIC1

		Schematic Title: **NUTS BCR Prototype**	
		Page Title: **Front Page**	
Designed: <OrgAddr1>	Approved: <OrgAddr2>	Document number: **<Doc>**	Revision: PA00
Size: A3	BOM Doc No: <Cage Code>		
Design Created Date: Wednesday, March 23, 2011		Sheet Created Date: Wednesday, March 30, 2011	Sheet Modified Date: Friday, June 03, 2011 — Sheet: 1 of 4

Battery Charge Regulator

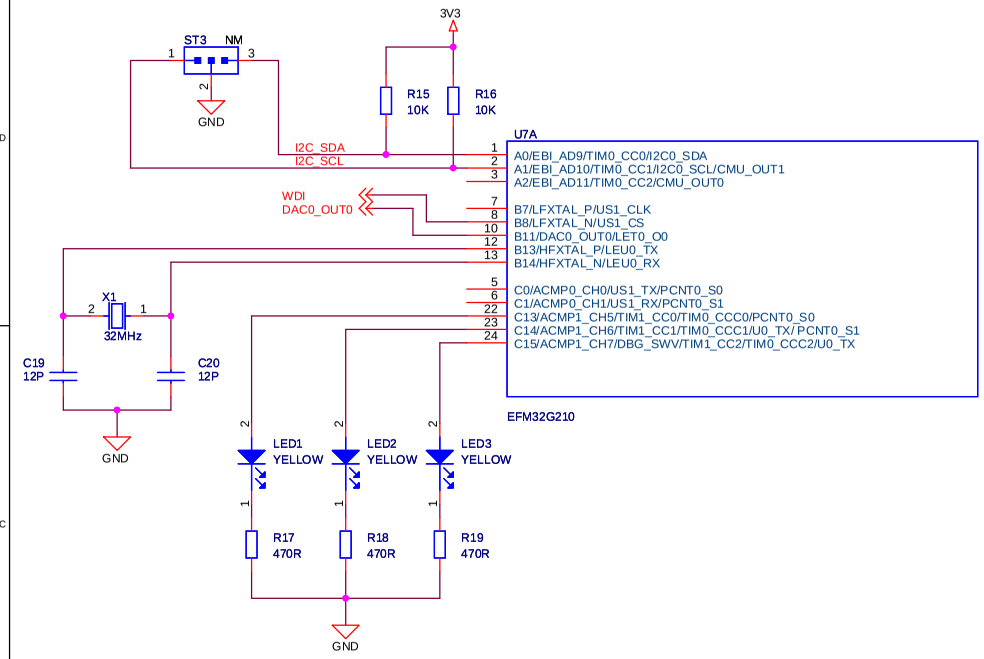


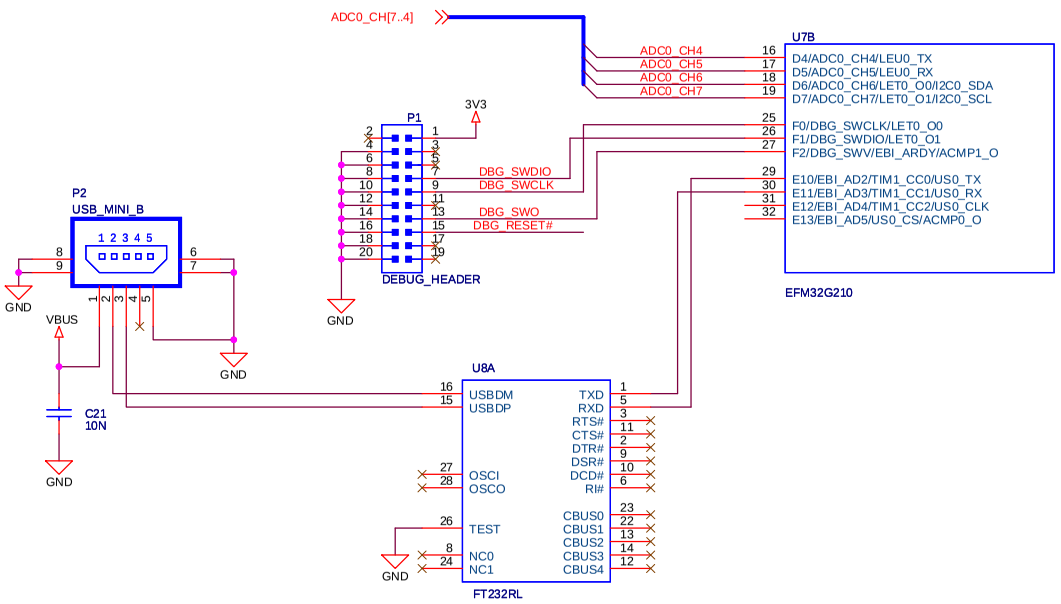
Decoupling

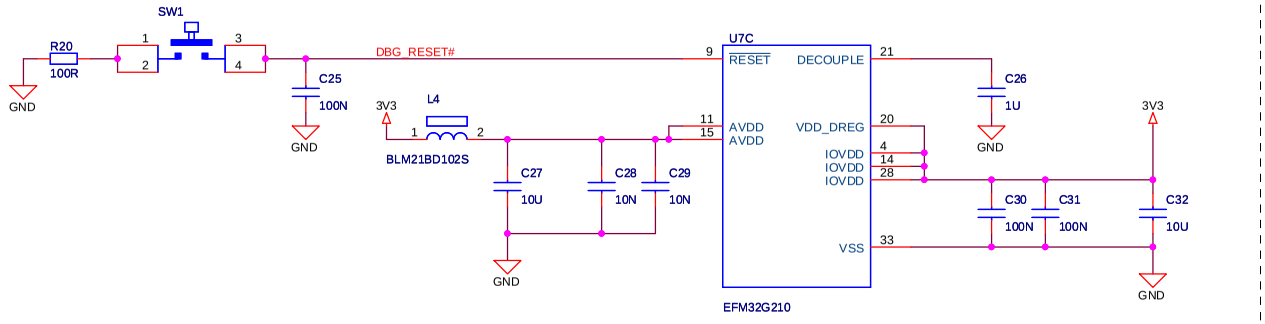


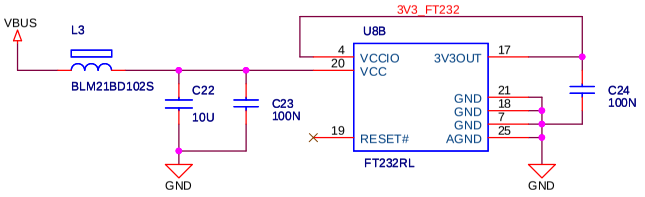
/
SCHEMATIC1

	Schematic Title: **NUTS BCR Prototype**
	Page Title: **Boost Regulator**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number: **<Doc>** / Revision: PA00
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, March 23, 2011	Sheet Created Date: Wednesday, March 23, 2011 / Sheet Modified Date: Monday, June 06, 2011 / Sheet: 2 of 4





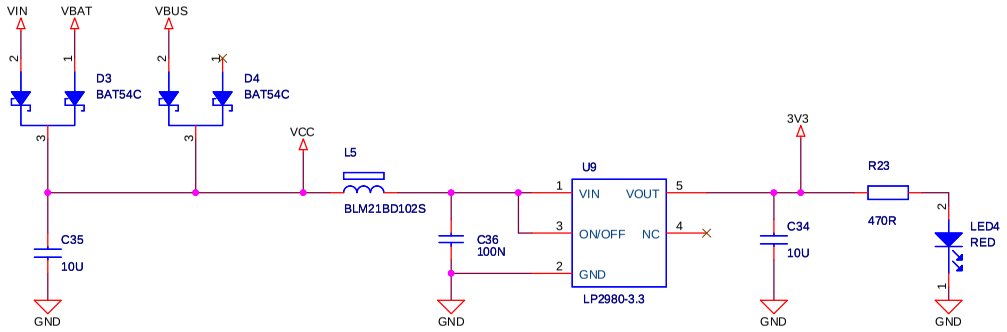




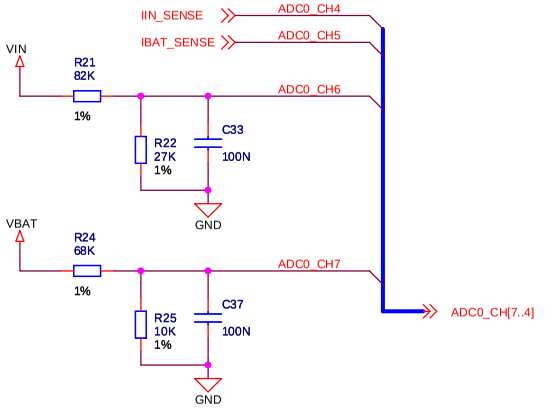
/
SCHEMATIC1

	Schematic Title: **NUTS BCR Prototype**
	Page Title: **Microcontroller**
Designed: <OrgAddr1> / Approved: <OrgAddr2>	Document number: **<Doc>** / Revision: PA00
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, March 23, 2011	Sheet Created Date: Tuesday, March 29, 2011 / Sheet Modified Date: Monday, June 06, 2011 / Sheet: 3 of 4

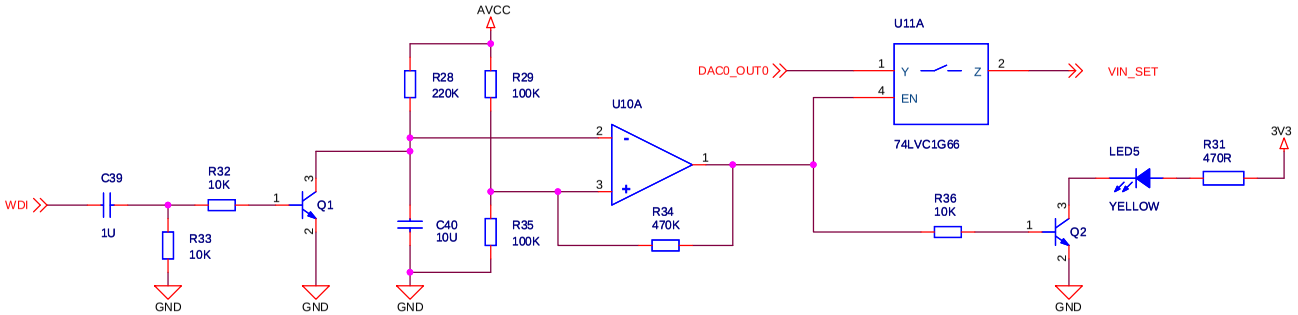
Local Power Supply

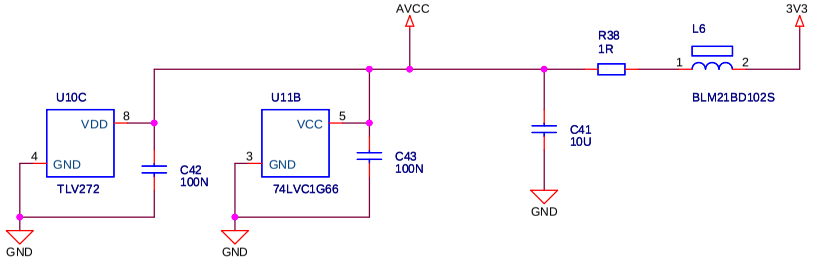


Voltage & Current Measurements

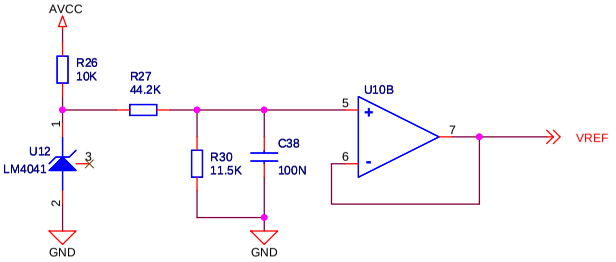


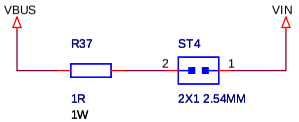
Analog Watchdog Circuit





250mV reference





Charging From USB

Connecting a jumper accross ST1 allows charging the batteries from the USB bus voltage. The resistor is in place to limit the charge current.

/
SCHEMATIC1

Designed: <OrgAddr1>	Approved: <OrgAddr2>
Size: A3	BOM Doc No: <Cage Code>
Design Created Date: Wednesday, March 23, 2011	

Schematic Title
NUTS BCR Prototype

Page Title
Analog & Power

Document number: <Doc>	Revision: PA00
Sheet Created Date: Tuesday, March 29, 2011	Sheet Modified Date: Monday, June 06, 2011
	Sheet: 4 of 4

Appendix C

LabView VIs

C.1 Power Distribution Unit Test Setup

The LabView program used int the power distribution unit test setup actually consists of a number of smaller VIs in a hierarchy. *init.vi*, shown in figure C.1, configures and opens the serial port connection (230400 baud), isues a `reset` command to the EFM32, and flushes the buffers so that the terminal is ready accept new commands.

The lowest level consists of *write.vi* and *read.vi*, shown in figures C.3 and C.2. These blocks format the address and data values into the form expected by the software running on the EFM32. In *write.vi*, the formatted string is sent through the serial port, while in *read.vi*, the `read` command is first sent, and then the response is read and returned.

On the next level of the hierarchy, VIs such as *adc.vi* (figure C.4) and *dac.vi* (figure C.5) use the low level read and write commands to access ADC data or write directly to the DAC output register.

The ADC is set up by the software running on the EFM32 to continuously sample and to use the DMA to transfer samples to the memory at address `0x20001000`. The memory location has been arbitrarily chosen inside the RAM space. By reading a 32-bit word from this memory location, *adc.vi* can obtain two 16-bit values corresponding to the measured voltage and current in the test setup.

The *dac.vi* VI accepts an output voltage between 0 and 1.25 V, which is converted to a 12-bit value and written directly into the DAC output register (address `0x40004020`).

C.1.1 Current Ramp

The VI used to generate a linear current ramp is shown in figure C.6. 1000 iterations with a period of 4 ms incrementally increases the load current. During each

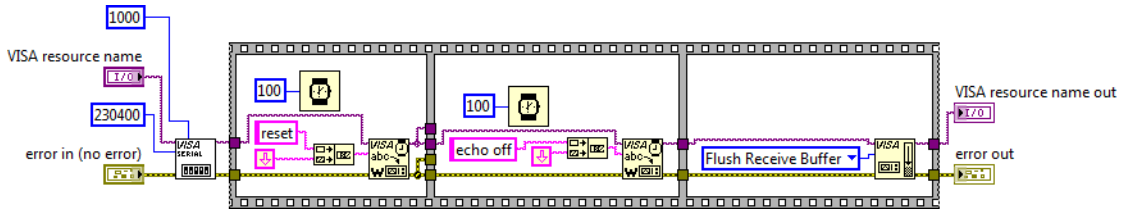


Figure C.1: init.vi

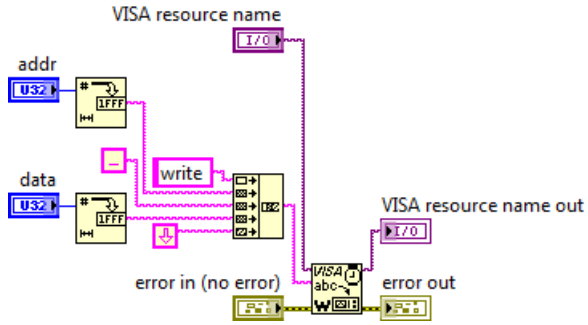


Figure C.2: write.vi

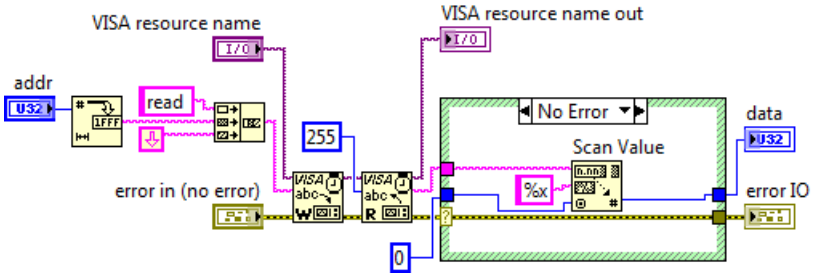


Figure C.3: read.vi

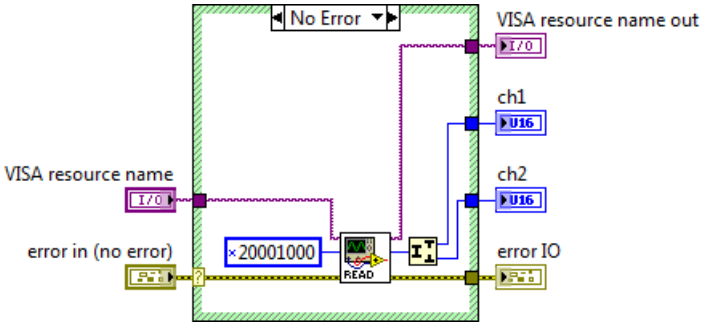


Figure C.4: adc.vi

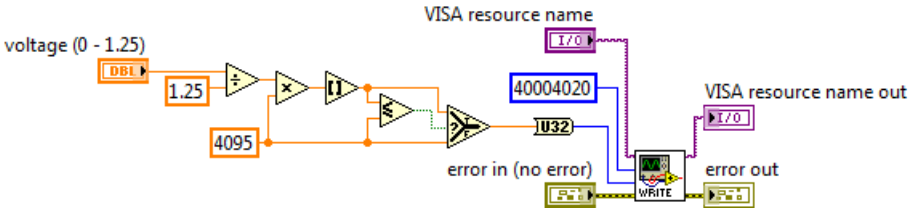


Figure C.5: dac.vi

iteration the DAC VI is used to set the load current, and the ADC VI is used to get the voltage and current measurements.

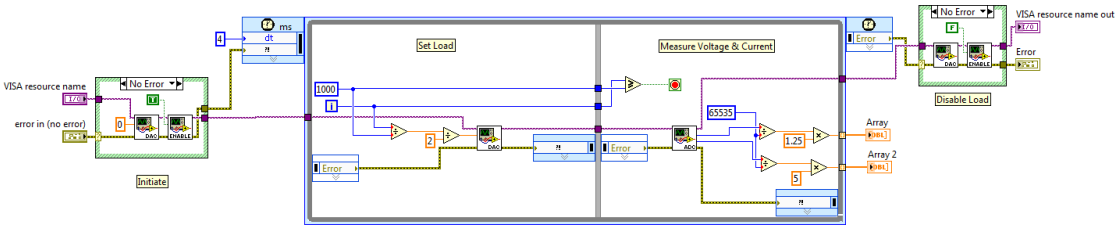


Figure C.6: Current Ramp VI

Appendix D

Loadbench EFM32 Source Code

The code listings presented here were used in the EFM32 microcontroller during testing of the backplane power distribution unit.

D.1 main.c

Sets up the ADC to continuously sample and the DMA to transfer samples to memory location `0x20001000`. Also sets up communication, then enters an eternal loop waiting for commands.

```
#include <stdbool.h>
#include <string.h>
#include <stdio.h>
#include "efm32_chip.h"
#include "efm32_dac.h"
#include "efm32_adc.h"
#include "efm32_cmu.h"
#include "efm32_adc.h"
#include "efm32_dma.h"
#include "efm32_prs.h"
#include "efm32_gpio.h"
#include "efm32_timer.h"
#include "dvk.h"
#include "usart.h"
#include "cmdshell.h"
#include "dmactrl.h"

#define ADC_FREQ     800
#define NR_CHANNELS 2

#pragma location=0x20001000
__no_init volatile unsigned short ADC_data[NR_CHANNELS];


/* DMA callback structure */
DMA_CB_TypeDef DMAcallBack;

/**************************************************************************//**
 * @brief
```

```
 *    Callback function called when the DMA finishes a transfer.
 * @param channel
 *    The DMA channel that finished.
 * @param primary
 *    Primary or Alternate DMA descriptor
 * @param user
 *    User defined pointer (Not used in this example.)
 *****************************************************************************/
void DMATransferComplete(unsigned int channel, bool primary, void *user)
{
  /* Refresh the DMA control structure */
  DMA_RefreshPingPong(0,
                      primary,
                      false,
                      NULL,
                      NULL,
                      NR_CHANNELS - 1,
                      false);
}

/*****************************************************************************//**
 *
 * Initiate Digital to analog converter
 *
 *****************************************************************************/
void DAC_setup(void)
{
  DAC_Init_TypeDef dacInit = DAC_INIT_DEFAULT;
  DAC_InitChannel_TypeDef channelInit = DAC_INITCHANNEL_DEFAULT;

  /* Digital to Analog Converter */
  DAC_Init(DAC0, &dacInit);

  channelInit.refreshEnable = true;
  DAC_InitChannel(DAC0, &channelInit, 0);

  DAC0->CH0DATA = 0;

  DAC_Enable(DAC0, 0, true);
}

/*****************************************************************************//**
 *
 *    Setup DMA in ping pong mode
 *
 *****************************************************************************/
void DMA_setup(void)
{
  /* DMA configuration structs */
  DMA_Init_TypeDef        dmaInit;
  DMA_CfgChannel_TypeDef chnlCfg;
  DMA_CfgDescr_TypeDef    descrCfg;

  /* Initializing the DMA */
  dmaInit.hprot        = 0;
  dmaInit.controlBlock = dmaControlBlock;
  DMA_Init(&dmaInit);

  /* Set the interrupt callback routine */
  DMAcallBack.cbFunc = DMATransferComplete;

  /* Callback doesn't need userpointer */
```

```
  DMAcallBack.userPtr = NULL;

  /* Setting up channel */
  chnlCfg.highPri   = false; /* Can't use with peripherals */
  chnlCfg.enableInt = true;  /* Interrupt needed when buffers are used */

  chnlCfg.select = DMAREQ_ADC0_SCAN;
  chnlCfg.cb     = &DMAcallBack;
  DMA_CfgChannel(0, &chnlCfg);

  /* Setting up channel descriptor */
  descrCfg.dstInc = dmaDataInc2;
  descrCfg.srcInc = dmaDataIncNone;
  descrCfg.size   = dmaDataSize2;

  /* We have time to arbitrate again for each sample */
  descrCfg.arbRate = dmaArbitrate1;
  descrCfg.hprot   = 0;

  /* Configure both primary and secondary descriptor alike */
  DMA_CfgDescr(0, true, &descrCfg);
  DMA_CfgDescr(0, false, &descrCfg);

  /* Enabling PingPong Transfer*/
  DMA_ActivatePingPong(0,
                       false,
                       (void *) &ADC_data[0],
                       (void *) &(ADC0->SCANDATA),
                       NR_CHANNELS - 1,
                       (void *) &ADC_data[0],
                       (void *) &(ADC0->SCANDATA),
                       NR_CHANNELS - 1);
}


/**************************************************************************//**
 *
 * Initiate Analog to Digital converter
 *
 *****************************************************************************/
void ADC_setup(void)
{
  ADC_Init_TypeDef adcInit  = ADC_INIT_DEFAULT;
  ADC_InitScan_TypeDef scanInit = ADC_INITSCAN_DEFAULT;

  adcInit.timebase = ADC_TimebaseCalc(0);
  adcInit.prescale = ADC_PrescaleCalc(8000000, 0);
  adcInit.ovsRateSel = adcOvsRateSel256;

  /* Init adc */
  ADC_Init(ADC0, &adcInit);

  /* Setup continous scan conversion on channels 0 to NR_CHANNELS-1 */
  scanInit.resolution = adcResOVS;
  scanInit.prsSel = adcPRSSELCh0;
  scanInit.prsEnable = true;
  //scanInit.input = (0xff >> (8-NR_CHANNELS)) << _ADC_SCANCTRL_INPUTMASK_SHIFT;
  scanInit.input = ADC_SCANCTRL_INPUTMASK_CH1 | ADC_SCANCTRL_INPUTMASK_CH2;

  /* Setup single conversion, continous */
  ADC_InitScan(ADC0, &scanInit);
}
```

```
/**************************************************************************//**
 *
 * Initiate Timer to trigger ADC
 *
 *****************************************************************************/
static void TIMER_setup(void)
{
  uint32_t timerTopValue;

  /* Use default timer configuration, overflow on counter top and start counting
  * from 0 again. */
  TIMER_Init_TypeDef timerInit = TIMER_INIT_DEFAULT;

  /* Timer clock source:
   */
  timerInit.clkSel = timerClkSelHFPerClk;
  timerInit.prescale = timerPrescale8;

  TIMER_Init(TIMER0, &timerInit);

  /* PRS setup */
  /* Select TIMER0 as source and TIMER0OF (Timer0 overflow)
     as signal (rising edge) */
 PRS_SourceSignalSet(0, PRS_CH_CTRL_SOURCESEL_TIMER0,
                     PRS_CH_CTRL_SIGSEL_TIMER0OF, prsEdgePos);

  /* Calculate the proper overflow value */
  timerTopValue = CMU_ClockFreqGet(cmuClock_TIMER0) / 8 / ADC_FREQ;

  /* Write new topValue */
  TIMER_TopBufSet(TIMER0, timerTopValue);

}


/**************************************************************************//**
 *
 * Main function: program entry point.
 *
 *****************************************************************************/
int main()
{
  int i;

  /* Chip errata */
  CHIP_Init();

  /* Switch to HFXO (32MHz crystal clock) */
  CMU_ClockSelectSet(cmuClock_HF, cmuSelect_HFXO);
  CMU_OscillatorEnable(cmuOsc_HFRCO, false, false);

  /* Board support */
  DVK_init();
  DVK_enablePeripheral(DVK_RS232A);

  for(i = 0; i < NR_CHANNELS; i++)
    ADC_data[i] = 0;

  /* Start clocks */
  CMU_ClockEnable(cmuClock_ADC0, true);
  CMU_ClockEnable(cmuClock_DMA, true);
```

```
  CMU_ClockEnable(cmuClock_DAC0, true);
  CMU_ClockEnable(cmuClock_TIMER0, true);
  CMU_ClockEnable(cmuClock_PRS, true);
  CMU_ClockEnable(cmuClock_GPIO, true);

  GPIO_PinModeSet(gpioPortD, 15, gpioModePushPull, 1);


  /* Setup DMA and peripherals */
  DMA_setup();
  ADC_setup();
  TIMER_setup();
  DAC_setup();

  /* Initialize USART */
  USART1_Init();

  /* Enter cmdshell loop. will not return */
  cmdshell();
}
```

D.2 cmdshell.c

Implements parsing of commands from the USART, using standard ANSI C functions. If a match is found, the associated function pointer is called, with the arguments passed along.

```
#include <stdio.h>
#include <string.h>
#include "cmdshell.h"
#include "cmdlist.h"
#include "usart.h"
#include "efm32_emu.h"

char echo = 1;

int cmdParse( char *line );

static char cmd_buffer[CMD_MAX_LENGTH];

static int cmdGetLine( char *cmd_buffer, size_t max )
{
  char c = 0;
  unsigned int length = 0;

  while( c != '\n')
  {
    c = getchar();
    if( c == '\r' )
      c = '\n';

    switch(c)
    {
    case 0xff:
      EMU_EnterEM1();
      break;

    case '\n':
      *cmd_buffer = '\0';
      if(echo)
        putchar('\n');
      break;

    case '\b':
      if( length > 0 )
      {
        if(echo)
        {
          putchar('\b');
          putchar(' ');
          putchar('\b');
        }
        *(--cmd_buffer) = '\0';
        length--;
      }
      break;

    default:
      if(length != max - 1)
      {
        *cmd_buffer++ = c;
```

```
        if(echo)
          putchar(c);
        length++;
      }
      break;
    }
  }
  return length;
}

int cmdParse( char *line )
{
  char *argv[CMD_MAX_PARAM];
  int argc;
  int index;
  int i;

  argv[0] = strtok(line, " ");
  argc = 0;
  do
  {
    argc++;
    argv[argc] = strtok(NULL, " ");
  }while(argv[argc] && (argc < CMD_MAX_PARAM));

  if( !strcmp(argv[0], "h") )
  {
    printf("Available commands:\x0D\x0A");
    for(i = 0; i < NR_COMMANDS; i++)
      printf( "%s:\r\x1B[20C%s\n",commands[i], descriptions[i] );
    printf("\necho - turn echo on or off, h - show available commands\n");
    return 0;
  }

  if( !strcmp(argv[0], "echo") )
  {
    if( !strcmp(argv[1], "on") )
      echo = 1;
    else if( !strcmp(argv[1], "off") )
      echo = 0;

    return 0;
  }

  index = -1;

  for(i = 0; i < NR_COMMANDS; i++)
  {
    if( !strcmp(argv[0], commands[i]) )
      index = i;
  }


  if(index == -1)
  {
    if(echo)
      printf("Command not found\n");
    return -1;
  }
  else
    return ((*func_table[index])(argc,argv));
}
```

```
void cmdshell(void)
{
  while(1)
  {
    /* Display echo */
    if(echo)
      printf(cmdPrompt);

    /* Get and parse a command line */
    if( cmdGetLine( cmd_buffer, CMD_MAX_LENGTH ) )
      cmdParse( cmd_buffer );
  }
}
```

D.3 cmdlist.h

Contains a list over all the available commands for the command shell together with their names and descriptions. Used by `cmdshell.c` to detect a command match.

```
#ifndef __CMDLIST_H_
#define __CMDLIST_H_

#define NR_COMMANDS (sizeof(func_table)/sizeof(cmd_t))

extern int verCmd (int argc, char *argv[]);
extern int writeCmd (int argc, char *argv[]);
extern int readCmd (int argc, char *argv[]);
extern int resetCmd (int argc, char *argv[]);

typedef int (*cmd_t)(int, char*[]);

/*
*   Command function table.
*   Enter command functions here in the same order as they
*   are entered into the name and the descripton tables.
*/
const cmd_t func_table[] = {
  verCmd,               // 1
  writeCmd,             // 2
  readCmd,              // 3
  resetCmd              // 4
};

/*
*   The names to call the commands
*/
const char  *commands[] = {
  "ver",                // 1
  "write",              // 2
  "read",               // 3
  "reset"               // 4
 };

/*
* A short description for each command
*/
const char *descriptions[] = {
  "show version information",               // 1
  "write a memory location",                // 2
  "read a memory location",                 // 3
  "reset the microcontroller"               // 4
};


#endif //__CMDLIST_H_
```

D.4 cmd_funcs.c

The actual commands that have been implemented: `read`, `write`, `reset` and `ver`. Each command take the standard form `int cmd(int argc, char *argv[])`.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "usart.h"
#include "efm32.h"

#define VER "Dewald's command shell v1.2 running on EFM32\n"

extern char echo;

int verCmd(int argc, char *argv[]);


/*
* Display some version info
*/
int verCmd(int argc, char *argv[])
{
  (void)argc;
  (void)argv;

  printf(VER);
  return 0;
}

/*
 *
 */
int writeCmd(int argc, char *argv[])
{
  uint32_t addr;
  uint32_t data;

  if(argc != 3)
  {
    printf("Usage: write <32bit addr> <32bit data>");
    return -1;
  }

  sscanf(argv[1], "%x", &addr);
  sscanf(argv[2], "%x", &data);

  if(echo)
    printf("Setting memory address %x to value %x\n", addr, data);

  if (addr > 0x40010C00 || addr < 0x40000000)
  {
    printf("Error! Not a memory system peripheral address!\n");
    return -1;
  }

  *((unsigned int *) addr) = data;

  return 0;
}
```

```
/*
 *
 */
int readCmd(int argc, char *argv[])
{
  uint32_t addr;
  uint32_t data;

  if(argc != 2)
  {
    printf("Usage: read <32bit addr>");
    return -1;
  }

  sscanf(argv[1], "%x", &addr);
  data = *((unsigned int *) addr);

  printf("%x\n", data);

  return 0;
}

int resetCmd(int argc, char *argv[])
{
  (void) argc;
  (void) argv;

  if(echo)
    printf("Resetting EFM32\n");

  NVIC_SystemReset();
  return 0;
}
```