Developing a green power system for a telecom network in rural Tanzania

CAMILO JOSE CHAVES MONTILLA
camilojchaves@gmail.com

Examiner:

Peter Sjödin psj@kth.se

Advisors:

Björn Pehrson bpehrson@kth.se
Robert Olsson robert@herjulf.net

Javier Simó Reigadas javier.simo@urjc.es

A mi mamá, quien me ha enseñado

a hacer siempre mi mejor esfuerzo.

Contents

1. Introduction

ICT project[38] started in Tanzania in 2006, as a KTH's communication system project, its main purpose was to support the U.N. Millennium Goals (eHealth, eSchool and eGovernment), so they started the creation of a network around the Bunda-Mugumu power line in Tanzania, bringing the equipment and making the design. In 2010, BURUCA [37] team deployed the network between Bunda, Nata and Mugumu, including fiber optic, 2 Bifrost routers, servers, DNS, etc. Once developed, one of the main problems has been the power breakouts, happening randomly almost every day and leading to service interruption, equipment damage and customer complaints.

To help to solve BURUCA's problem, students and staff at the KTH telecommunication systems laboratory have developed a low-power router based on open source software and off-the-shelf hardware components. The router has an integrated power management module to accommodate the use of different types of power sources, such as solar and wind, as well as different kind of power storage, including conventional batteries and ultracapacitor cells. Testing tools for measuring of performance and power consumption have also been developed. Besides laboratory tests, there are also field tests in progress [38].

To resume and complement ICT´s and BURUCA's work, the purpose of this thesis project is to analyze, design and implement solutions to solve the power supply problems in BURUCA telecommunications network in Tanzania, while making studies, analysis, documentation and designs also applicable to other similar networks in developing countries.

The first analysed option was to take the design of the not yet completed power system for BURUCA and take its control circuitry to the next level, an intelligent system. There were choices to make and test, we finaly decided that the network equipments could be feed by either the Main power grid, a common battery, solar panels or ultracapacitors, an intelligent system should take measurements and make decisions, selecting the different power sources and control the energy transfer, while reporting data to the user through the operation and management console.

The starting point of this thesis project is an incomplete and untested design for BURUCA coming out of the Minne3 project. As the main technical activity for accomplishing the goal, the task has thus been to complete and test this design and to make a new iteration resulting in an improved design. The methodology and tools used are discussed as well as the results of testing the Minne3 design and the Minne4 iteration. As results, we developed and tested several solutions, while documented all the technical issues to make it easier for future improvement teams [see deliverables list], and we produced a control circuitry capable of:

- A micro controller based managing system, composed by:
    - Several voltage and current measurements.

- PWM control over the boost/buck converter.
- Managed by user commands through a serial communication.
- Capacity for remote software upgrades over internet.
- Temperature measurement.

- A balancing circuit to keep every ultracapacitor with the same Voltage level and avoid damages for unbalancing.

- A High current (5A) Buck/Boost DC-DC voltage converter to adjust the charging voltage for the ultracapacitors.

- A Current limiter circuitry to prevent component damages due to the ultra low ultracapacitor's internal resistance.

1.1. [En español] Introduccion

El proyecto ICT [38] comenzó en Tanzania en el 2006, como un proyecto del departamento de comunicaciones de KTH, su propósito principal fue soportar los objetivos del milenio de las N.U. (eHealth, eSchool and eGovernment), así que KTH comenzó la creación de una red a lo largo de la línea de energía entre Bunda y Mugumu en Tanzania, llevando allí los equipos y realizando el diseño. En 2010 el equipo BURUCA[37] desplegó la red entre Bunda, Nata y Mugumu, incluyendo la fibra óptica, enrutadores basados en Bifrost, servidores, equipos DNS, etc. Una vez desplegada la red, uno de los mayores problemas ha sido las caídas de energía, que ocurren aleatoriamente casi todos los días, produciendo interrupción del servicio, daños en los equipos e insatisfacción de los usuarios.

Para solucionar el problema de BURUCA, y en etapas previas a el presente trabajo de tesis, estudiantes del sistema de comunicaciones de KTH han desarrollado un router de bajo consumo de energía basado en código abierto y componentes distribuidos. El enrutador tiene un sistema de rectificación de energía y se han hecho pruebas de campo y d elaboratorio sobre su rendimiento bajo alimentación por energías renovables como eólica y solar [38].

El presente trabajo busca complementar los esfuerzos de ICT y BURUCA, diseñando e implementando soluciones para solucionar los problemas de energía en la red de telecomunicaciones de Tanzania, al tiempo que realiza estudios, análisis y documentación aplicables a otras redes similares en países en vía de desarrollo.

La primera opción considerada fue retomar el diseño no completado del sistema de energía de BURUCA y tomar su circuitería al nivel inteligente. Realizando análisis, decisiones y pruebas, finalmente se decidió que la red debería ser alimentada por la red pública de energía, baterías de acido comunes, paneles solares ó supercapacitores, un modulo inteligente debería controlar, monitorizar y tomar las decisiones, seleccionar la fuente de energía, controlar la transferencia de energía y reportar los datos obtenidos hacia una consola de operación y mantenimiento con interacción de usuario.

El punto de partida del presente trabajo es un incompleto diseño realizado por Minne3[1], como una de las tareas principales para cumplir el objetico final, fue necesario completar y probar el diseño de Minne3, para luego diseñar un nuevo y mejorado sistema llamado Minne4 (presente trabajo). En el presente documento se explica la metodología utilizada, los resultados de las pruebas al diseño de Minne3 y Minne4, y se ha documentado todas las tareas técnicas para facilitar a los grupos siguientes la mejora continua de redes de éste tipo [ver lista de entregables]. El resultado palpable ha sido un sistema compuesto por:

- Un sistema de administración de energía montado en una placa que incluye:
    - Varios medidores de voltaje y corriente.

- Control PWM sobre los conversores boost/buck.
- Administración de usuario vía comandos sobre comunicación serial.
- Soporte de actualizaciones remotas sobre internet.
- Mediciones de temperatura.

- Un sistema de balanceo para mantener cada supercapacitor con el mismo voltaje y asi evitar daños por desbalanceo, el mayor enemigo de los supercapacitores.

- Conversores Buck/Boost DC-DC de alta corriente (5A) para ajustar la carga de voltaje de los supercapacitores.

- Un limitador de corriente para prevenir daños en los componentes debido a la Resistencia ultra baja de los supercapacitores.

1.2. Background

KTH's students and teachers have developed a good networking proposal[38] for any location where there are not enough economic, or infrastructure resources, they have Tanzania's network as their first target, that network is currently running and still needing improvements.

The URJC (Universidad Rey Juan Carlos) in Madrid, and its Signal & Telecommunications department give efforts to develop ICT4D projects and have developed a master program totally oriented to provide knowledge to solve telecommunication and information systems problems in areas where the deployment is difficult due to reasons such as isolation or socio-economic problems, the project running in KTH has been done by students, and there is still much to keep doing. This work seeks to analyze, design and implement solutions to solve the power supply problems in BURUCA telecommunications network in Tanzania, while making studies and designs also applicable to other similar networks in developing countries, it is also the final work for the "Master in Telecommunications Networks for Developing Countries", Madrid, Spain. [26]

KTH has pointed to investigate and implement super capacitors as a power source for BURUCA Network, and in 2010, the Nobel Prize for physics went to Andre Geim and Konstantin Novoselov, for their discovery of grapheme, (an evolution of our super capacitors electrolyte), a sheet-like substance made of carbon atoms bonded together in a repeating hexagonal pattern. It is the first essentially two-dimensional material ever made. Part of the current project aims to replace the common battery by the BCA3000F super capacitor, its combination of enormous surface area and extremely small charge separation gives a 3000F capacitance, but with advantages such as over one million times charges, low heating, 0.29mΩ ESR, 147A constant or 2000A peak current.

1.3. [En español] Background

Estudiantes y profesores de KTH, han desarrollado una buena propuesta para redes desplegadas en cualquier ubicación en donde no existen suficientes recursos económicos o de infraestructura, han elegido Tanzania como la primera red objetivo, dicha red está actualmente en funcionamiento bajo el nombre de BURUCA, pero debido a diferentes problemas y falta de recursos, dicha red sin embargo, necesita constantes mejoras en calidad de servicio y operación y mantenimiento.

La Universidad Rey Juan Carlos (URJC) en Madrid, España, y su departamento de señales y telecomunicaciones se han esforzado en desarrollar proyectos de ICT4D y han desarrollado el programa de maestría “Redes de telecomunicaciones para países en desarrollo” [26], totalmente orientado a proveer conocimientos para resolver problemas en telecomunicaciones y sistemas de información en áreas en donde el despliegue es difícil debido a razones como aislamiento ó problemas socio económicos. El presente trabajo busca analizar, diseñar e implementar soluciones para el problema de energía en BURUCA o otras redes similares, también hace parte del trabajo final para terminar dicha maestría.

Adicionalmente, parte del presente proyecto busca reemplazar las baterías comunes por el super capacitor BCA3000F, que gracias a su enorme área de superficie y extrema separación de cargas puede producir 3000F de capacitancia, con ventajas como: hasta un millón de cargas, bajo calentamiento, 0.29m0 de ESR, 147A constantes ó 2000A de corriente de pico. El uso de los supercapacitores fue inspirado por el premio Nobel de física en 2010, otorgado a Andre Geim y Konstantin Novoselov, por el descubrimiento del grapheme, una evolución del electrolito utilizado en nuestros supercapacitores, es una sustancia tipo papel hecha de átomos de carbón unidos unos a otros en un patrón hexagonal repetitivo, es el primer material de dos dimensiones nunca antes hecho.

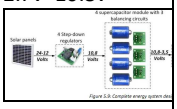
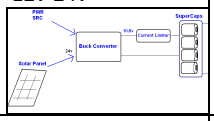
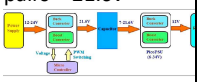
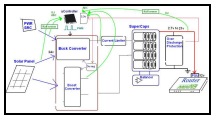
1.4. Previous works

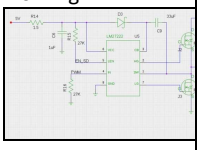
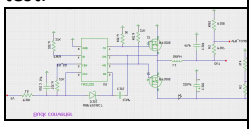
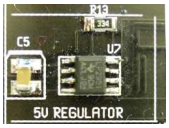
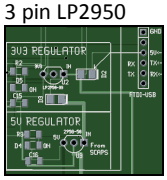
The physical created system in the present thesis is called Minne4. Minne1 (2009), Minne2 (2010) and Minne3 (2010) were also communication system projects oriented to the development, aimed to develop improvements for the Tanzanian network.

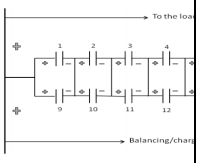
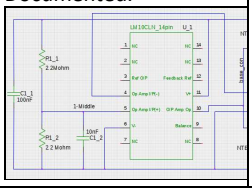
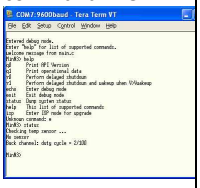
ICT project[38] started in Tanzania in 2006, as a KTH's communication system project, its main goal was to support the U.N. Millennium Goals (eHealth, eSchool and eGovernment), so they started the creation of a network around the Bunda-Mugumu power line in Tanzania, bringing the equipment and making the design. In 2010, BURUCA [37] team deployed the network between Bunda, Nata and Mugumu, including fiber optic, 2 Bifrost routers, servers, DNS, etc.

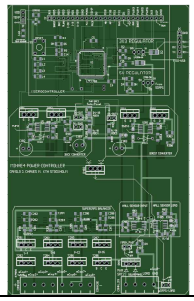
1.3.1 Minne1, Minne2, Minne3, Minne4, Minne5 overview

Here in Minne4, we introduce a brief comparison table for Minne project history, it will give a brief idea about the project evolution and will suggest task for the upcoming Minne5 team.

Minne project history					
	Minne1	**Minne2**	**Minne3**	**Minne4**	**TODO in Minne5**
Overall	4 SuperCaps 2.7v=10.8v 	4 SuperCaps 2.7v=10.8v SolarPanel 12v-24v 	16 superCaps in pairs = 21.6v 	16 superCaps in pairs = 21.6v 	More tests.
Router	First design.	27.95 w max 543 pps	26.3 w max 732 pps Hibernation. Sensing. MotherBoard: Supermicro X7SPA-HF-D525 Intel Atom. Interface: Masters Niagara 42084 Bifrost 6.	No HW tests. Produce Bifrost 7 documentation.	Program more features. Log collection, alarms, O&M, etc.

Steep down	4 x 10.8 v for supercaps. High current.	1 steep down. 10.8v High current. LM3150. Input 6-24v Programmable output. 10A. 	In PCB: Not working 	Design in bread board working. Design in PCB ready to test. 	Mount Minne4 PCB and test.
Regulators			8 pin LP2951. Bad design. 3V3 and 5V 	Minne3 design tested by Minne4. New design tested and documented. 3V3 and 5V 3 pin LP2950  PCB not tested.	Mount Minne4 PCB and test.
Steep UP	Output:12v 3 modules in parallel. LM2557(3A)800 ma load current each.	Same as Minne I	In PCB: Not working.	Design in bread board not working. Design in PCB ready to test.	More bread board tests. Mount Minne4 PCB and test.
Current limiter	none	2 power resistors.	Not documented. Not tested.	Software module. Needed to don't damage the circuitry. Hall sensors designed in Minne4 PCB, tested en bread board. Software part not developed.	Mount Minne4 PCB and test. Program the algorithm
PicoPSU	dc-dc converter	dc-dc converter 6v-34v Output:5v,12v Could replace the stepUP module. Turns off when input is below 6v, then it makes discharge protection.	Not in the PCB. 12v output	Same.	Same. Check if other Motherboards require simpler PSUs.

Overdischarge protection	None. -> high current when low voltage.	LTC1440 (comparator) Disconnects the load (router) when v<2.7v	Router scripts tested by Minne3. Microcontroller part not developed, not tested.	It is a SW module. ADC in microcontroller ready. Serial communication to router ready. Not tested	Program the algorithm in uController and test with Minne3 scripts in the router.
Supercaps	4 6.6A to charge a discharged cap. 0.75v needs 4.8A to charge.	15 min up.	16 supercaps =3000F x 16 . 2 hours up.	Same. Testing done, Balancing improvement.	Same.
Balancing	3 balancing circuits. OpAmps	LM10 opAMP 1.1v-40v + Darlington TIP120 TIP125: 5A balancing current.	5A Darlington. Current limitation: 10A  PCB design not working.	Tested and working with 4 superCaps. Documented. 	Mount Minne4 balancing PCB and test with 16 superCaps.
MicroController	None.	Proposed.	LPC1768. G++ lite. Yagarto ToolChain ADC, PWM: not tested UART: to communicate to Flashing: JTAG	LPC1768 Eclipse Yagarto Toolchain UART: to communicate with router Flashing: ISP (UART). ADC ok, PWM ok, UART ok. -Improved user command line.	Mount the uController in Minne4 PCB. Test and improve.
UART	none	none	Basic user command line. 	Enhanced user command line, PWM, ADC, status commands. 	Improved the command line. Include new commands and features. (send log data to router, send alarms to router, etc)
Flashing	none	None.	Through JTAG.	JATG or ISP in blueboard. ISP in Minne4 PCB. Tested and documented.	Mount Minne4 PCB and test the flashing.
Thermometer		None.	The design does not work. Tests not documented. 2 wire DS18B20	Included in Minne4 PCB, tested but the uController part not working. 2 wire DS18B20	Find bugs in the uController code for temperature.

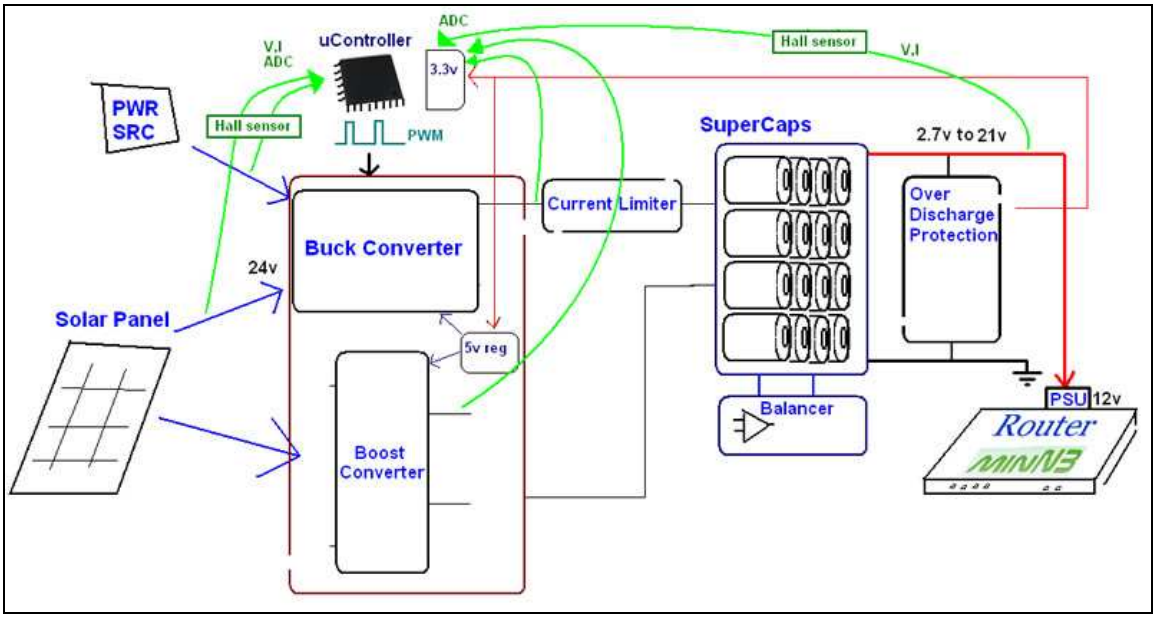
PCB		2 boards, not printed.	PCB. 35 um, 160x160mm Not tested by Minne3. 	Minne4 tested Minne3 PCB. New PCB designed. Well designed and documented. Not tested 	Mount and test the PCB. Propose improvements. Design a new PCB if needed.
Solar Panel			30W 24-12v	Not tested. Included in PCB design.	Apply the solar panel as power source.

1.5. Scope of the project

The present work is intended to be deployed in the Tanzania's network "BURUCA", where there are frequent power outages and the power consumption is a critical value. The project's main objectives are to solve the power problems in BURUCA network by building a power controller system capable to:

- manage the super capacitors charge and discharge
- Manage the solar panels output power
- Manage the temperature sensors
- Keep communication with the router so it can take decisions based on the collected data.
- Remote upgrades with none onsite operations.

The main system diagram summarises the proposed technical approach:



The main task for the system is to keep a router powered through an intelligent system, the microcontroller holds and runs the code for reading different voltages and currents (through the hall sensor) in the system and take different actions, such as produce the proper PWM signals to make the buck/boost converter produce the charging voltage for the super capacitors. The solar panel gives the necessary energy when there is no other power source. The current limiter system prevents high currents to destroy the circuitry and the over discharge protection system prevents data lose in the router and peak currents when charging. The balancer keeps all superCaps with the same voltage to prevent their damage. The system should run with no user operation,

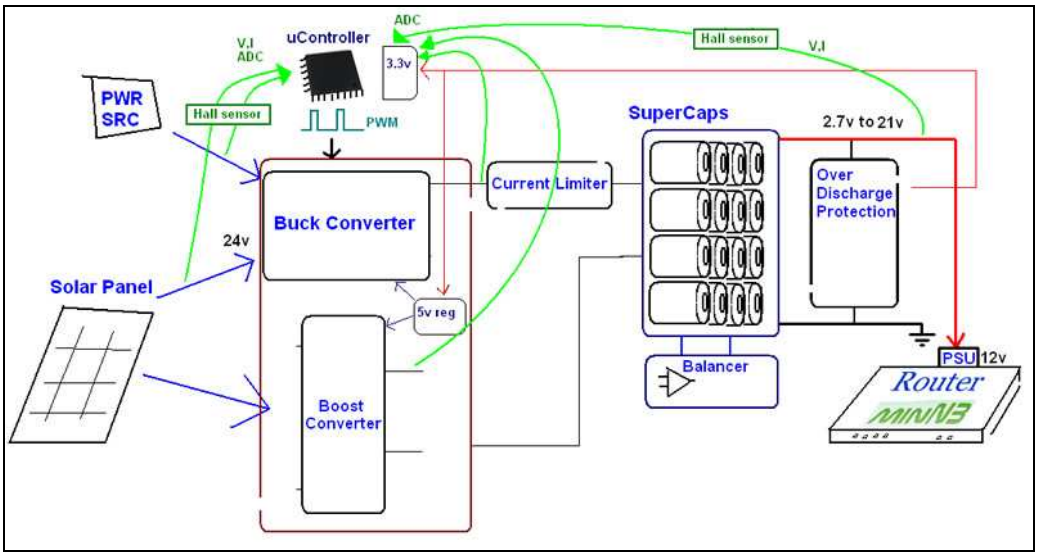
however there is a wide command user interface is available through a serial communication from the router.

1.6. [En Español] Alcance del proyecto

El presente trabajo está orientado a solucionar los problemas de energía en la red de Tanzania, en donde existen frecuentes caídas de voltaje y el consumo de potencia es un factor crítico. La tarea principal para solucionar el problema en dicha red es diseñar y construir un controlador de poder capaz de:

- Controlar la carga y descarga de los supercapacitores.
- Controlar la energía entregada por el panel solar.
- Controlar los sensores de temperatura.
- Mantener comunicación con el enrutador, tal que éste pueda tomar decisiones basadas en los datos recolectados.
- Realizar actualizaciones remotas sin ninguna operación en sitio.

El siguiente diagrama principal del sistema puede resumir el proyecto:



La tarea principal es mantener energizado un enrutador a través de un sistema inteligente, el micro controlador almacena y ejecuta el código para leer los diferentes voltajes y corrientes en el sistema y toma diferentes acciones, como generar la señal PWM apropiada para que el *buck/boost converter* produzca el voltaje de carga para los supercapacitores. El panel solar debe entregar la energía necesaria, cuando no hay otra fuente de poder. El *current limiter* previene que altas corrientes destruyan los circuitos y el sistema de *over discharge protection* previene la pérdida de datos en el enrutador y picos de corriente al iniciar la carga. Para prevenir su daño, el *balancer* mantiene a los super capacitores con un voltaje uniforme. El sistema debe funcionar sin operación de usuario, sin embargo existe una amplia interfaz de comandos a través de una comunicación serial desde el enrutador.

1.7. Project goals

Main goal

To propose, design and develop a power control system to solve the power supply problems in BURUCA telecommunications network in Tanzania, it should also be applicable to other similar networks in developing countries.

Secondary goals

- Based on BURUCA, as the target network, test, identify bugs and develop improvements for the Minne power controller, the routing equipment and the power source, including mounting and testing of the microcontroller, charging, balancing and super capacitors circuit.
- To develop a remote upgrade system for the proposed solution.
- To produce enough documentation so further work teams could continue the improvements.
- To test solar panels as one of the power sources. Non Accomplished.
- To improve the O&M system in the router. Non Accomplished.

1.8. Results and contributions

Based on the main goal of the project, "To propose, design and develop a power control system to solve the power supply problems in BURUCA telecommunications network in Tanzania, also applicable to other similar networks", Minne3 team in KTH has analysed the target network and its needs, and has proposed and developed a solution for the power problems happening in that network, the solution includes a power system fed by super capacitors, a solar panel or the main grid, all controlled by a microcontroller.

Starting point for the project

Several other teams have contributed to build, deploy and improve BURUCA's network, that network is running and main router element, also developed by the teams before us, is a high performance, low power consumption router based on Bifrost OS, including a serial communication to the uController. The last team was called Minne3's its website is http://csd.xen.ssvl.kth.se/csdlive/content/minne3 [1].

The main proposal to accomplish the main objective includes the technical realization of the Power controller system, we found two main options, to start over the designs, or to finish and test the proposal left by the last team, Minne3. We chose to continue and finish Minne3's work, they developed a Printed Circuit Board (PCB) containing the circuitry for their design, including the uController, the buck and boost converters, the current limiter and the superCaps balancer. That PCB was never mounted or tested, so that became our starting point.

Main contributions

Our goal is to solve the power problems in BURUCA, so we aimed to produce a ready to use power controller system for them, We mounted the elements in the PCB, tested all the modules, documented, corrected the bugs, included new features and finally, we designed a new PCB with the improvements and the new designs, The deliverables list is found in the end of the document.

Minne4 has delivered the following main achievements:

- **Understand the project and provide guides and documentation**

  We have developed a web site [10] containing the project description and all deliverables, also, we wrote guides and tutorials for every tested module.

- **Mount, test, identify bugs and provide new designs for all the modules in the Minne3 PCB.**

- **Design a new Printed Board with the new circuitry, including the following features:**

- Simple design
- Standard board for any other project
- Easily flasheable thought ISP
- Powered by USB or external source
- Pins match the Blueboard, Blueboard works.
- Leds showing the state, power ON, reset, test, etc.
- Modularity. Modules work separately.
- Double side Euroboard size 100x160mm
- Well labeled and documented

The designed board is shown in the following image:

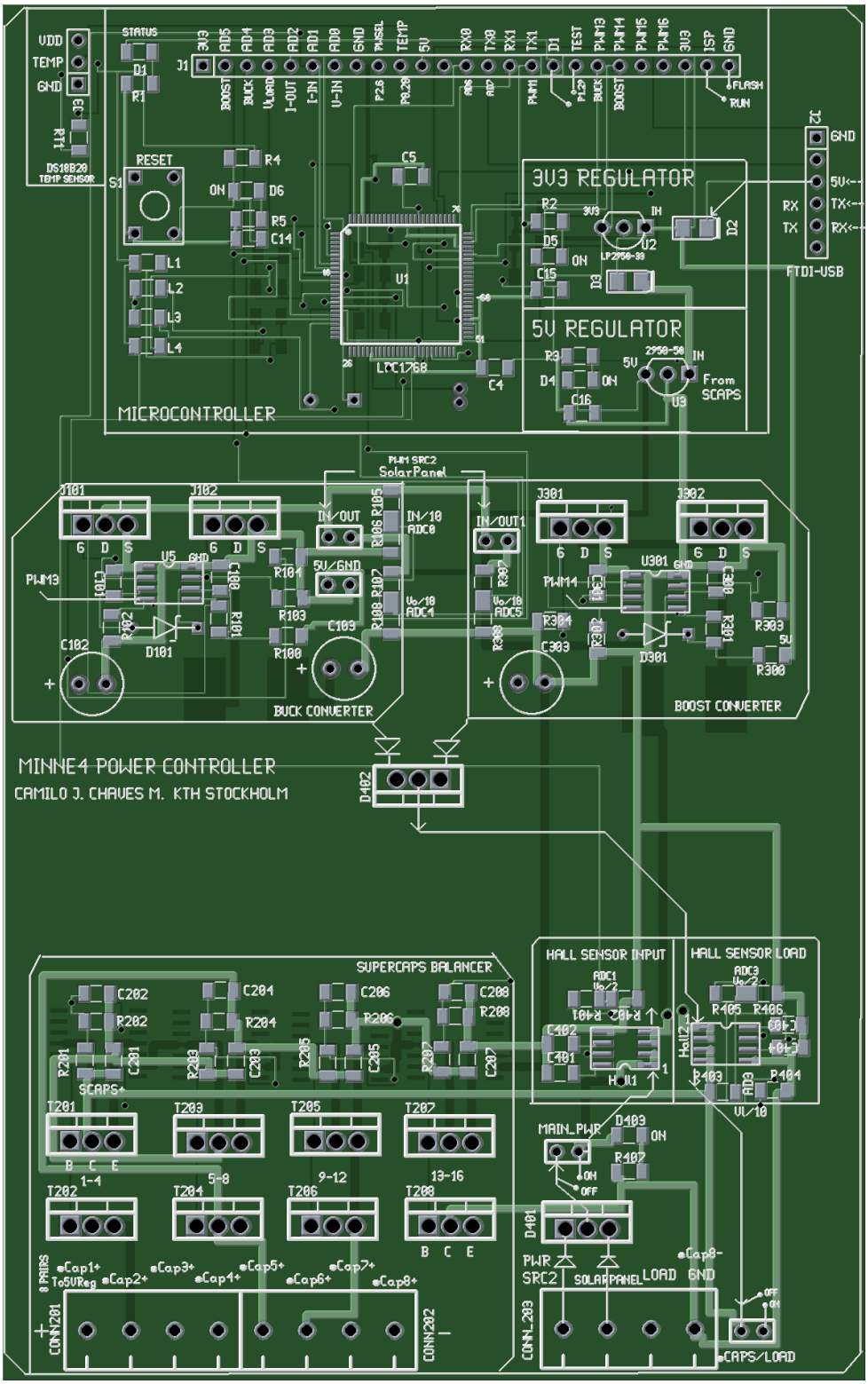


Image 2: Minne4 PCB board (zoom to see details)

1.9. Produced deliverables

Minne4 website	http://web.ict.kth.se/~cjcm/ Website containing the project description and all the deliverables
Minne1, Minne2, Minne3 and Minne4 comparison.pdf	Minne projects comparison table. http://web.ict.kth.se/~cjcm/deliverables/comparison/Minne1,%20Minne2,%20Minne3%20and%20Minne4%20comparisonV1.0.pdf
LPC1768H uController Software setup [Windows] v1_0.pdf	Detailed guide for setting up the development environment to work with the microcontroller. http://web.ict.kth.se/~cjcm/deliverables/LPC1768H_uController_SW_setup_Windows_v1_0.pdf
How to flash the microController v1.0.pdf	Detailed guide for flashing the LPC1768 microcontroller http://web.ict.kth.se/~cjcm/deliverables/how_To_flash/How%20to%20flash%20the%20LPC1768v1.0.pdf
Minne4 LPC1768 ADC programming v1.4.pdf	Detailed guide for programming the ADC module in LPC1768 http://web.ict.kth.se/~cjcm/deliverables/ADC/Minne4%20LPC1768%20ADC%20programming%20v1.4.pdf
Minne4 LPC1768 PWM programming v1.0	Detailed guide for programming the PWM module in LPC1768 http://web.ict.kth.se/~cjcm/deliverables/PWM/Minne4%20LPC1768%20PWM%20programming%20v1.0.pdf
Deploying Bifrost into virtual machine or USB stick v1.2.pdf	Detailed guide for deploying Bifrost 6 and 7, including a starting guide. http://web.ict.kth.se/~cjcm/deliverables/bifrost/Deploying_Bifrost_to_virtual_machine_or_USB_stick_v1.2.pdf
Minne4 superCaps v1.3.pdf	Maxwell BCA3000P superCaps analysis for Minne4 project, including charging and discharging profiles and the balancing circuit design. http://web.ict.kth.se/~cjcm/deliverables/supercaps/Minne4%20superCaps%20v1.3.pdf
Minne4 regulators.pdf	Report for the 3V3 and 5v regulators used in Minne4 project http://web.ict.kth.se/~cjcm/deliverables/regulators/MINNE4_regulatorsv1.3.pdf
Minne4 Hall Sensors.pdf	Report for Hall sensors used in Minne4 project http://web.ict.kth.se/~cjcm/deliverables.html
Eclipse workspace	Repository for Eclipse workspace with the latest LPC1768 code. http://web.ict.kth.se/~cjcm/deliverables.html
Minne4 introduction slides	Slides used as the project introduction http://web.ict.kth.se/~cjcm/deliverables/spring_ppt/Minne4%20project%20status%202011-08.pdf
Minne4 gEDA files	gEDA Schematics for all circuits in Minne4, including used symbols http://web.ict.kth.se/~cjcm/deliverables.html
Minne4 PCB files	.pcb files for the final PCB design http://web.ict.kth.se/~cjcm/deliverables.html

2. Methodology

In this section, based on the selected technical approach, we explain the different modules of the project, the previous technical works and the methods used by the present project Minne4.

2.1. Microcontroller module

As we needed an intelligent system capable to read data, take decisions and communicate with an administrator user, we had several options to fulfil this task:

- A discrete circuit capable to accomplish the needed tasks. The project requirements are complex and it results tedious and very difficult to develop the needed logic based on discrete elements, so this option was soon disposed.
- A computer running software capable to accomplish the requested tasks. If developing computer software is a realizable option, the power consumption, complexity of the hardware and the size of the elements would make it non-portable. Basically, to put a laptop onsite to control an energy system didn't seem to be a smart idea.
- A microcontroller running the needed logic. To keep the legacy design form Minne3 and develop a microcontroller based system, with low power consumption, remote administration and smart user interaction was a good option, we had the previous work documentation and they had also ran studies that leaded to a microcontroller design, so we decided to continue with this approach.

The designed microcontroller module is composed by the following sub-modules:

- **Microcontroller chip LPC1768**

  Minne1 and Minne2 for managing the power, used discrete elements mounted in universal soulding boards, there was no intelligence nor communication with the user or the router, and the results were limited to the commercial available circuits and its compatibility with the other modules. Minne3 and Minne4, in their final PCB include a microcontroller chip, storing and running the code, The LPC1768 chip was chosen by Minne3 team, considering its numerous general purpose pins, 8 ADC channels, 6 PWM channels, ISP feature, all by a reasonable 10€ price [11].

- **LPC1768H developing board (Blueboard)[13]**

  To develop a proper code, easy flash and test it, we used a developing board, when the code is ready, we can just flash it into the already mounted chip in the PCB. There are other developing boards, such as MBED [15], but the blueboards were already bought and after testing they shown to work fine, with lots of general purpose, ADC, PWM, etc pins to test, and a JTAG interface for debugging and flashing. Eclipse, OpenOCD, Yagarto toolchain, LPC21ISP, TeraTerm are some of the used software for the development.

- **Open-OCD module**

  OpenOCD (On Chip Debugger) is used for flashing and debugging, this last is a very useful feature for developing software, however, it requires an external programmer chip, Minne3 used OPENOCD-USB by embedded-projects [13], not for debugging but for flashing. Minne4 uses OpenOCD only for development on the Blueboard and it is not included in the final PCB, since it add complexity to the board and the flashing is better done using the ISP feature.

- **3.3v regulator**

  The 3.3v regulator feeds the microcontroller, Minne3 used the 8 pin LP2951, To keep the circuit simple, Minne4 uses the simpler 3 pin LP2950-33 despite it does not have any shutdown feature, and both offer 100ma. The 3.3v regulator can be feed by the 5V from the USB cable or by the superCaps output.

- **Serial communication with the router**

  The power system is connected to the router though a serial communication via UART, Minne3 and Minne4 use a FTDI cable [16], and Minne4 includes a complete command set to query or control the system modules.

- **ADC (Analog to Digital Converter) module**

  The microcontroller measure different voltages and currents (represented as voltages) through the ADC. Minne4 can read 6 different voltages and the system also apply algorithms to get reliable data. ADC can be queried, stopped or started through command line.

- **PWM (Pulse Wave Modulation) module**

  The microcontroller produces up to 4 PWM signals, 2 of them are used for driving the buck and boost converters. PWM can be queried, stopped or started through command line.

- **Firmware flashing module**

  The LPC1768 in the PCB can be flashed through ISP, a feature over the serial communication. In the developing board JTAG and Flashmagic can also be used. All scenarios were analysed and documented.

2.2. Power supply Module

We had different options while selecting the power environment, like this:

1. Re design the power system, create a new model using different power sources, power guidelines and components. We discarded this option because we understood that it would take much more time than we have to re start the main power structure design, also we had already chosen to continue with the original design got from Minne3.

2. To keep the original design from Minne3 and make the needed changes. This is the advised given by the couching team and the most reasonable approach, since the starting point of the project was an untested design got from the previous team (Minne3).

The power modules composed of low power and high power sub-modules:

- **5v regulator (low power)**

  The 2 Hall sensors are used to measure the input and output current, and the mosfets drivers need 5V to operate. Minne3 designed the 8pin LP2951 regulator, Minne4 tested it but chosed the simpler LP2950-50, both offer 100ma. It can be fed by the super capacitors or an external power source.

- **3.3v regulator (low power)**

  See the Microcontroller module above.

- **Super capacitors (High power)**

  Minne3 designed a super capacitors bank composed of 16 BCA3000P superCaps, each 3000F and 2.7v. those 16 superCaps in 8 pairs can provide 21.6v and support the system for 2 hours [6]. Minne4 did not test other configurations but the design allows any configuration with any number of capacitors forming up to 8 equivalent series capacitors. Minne4 also tested in the lab the charging/discharging profile and the needed balancing circuit.

- **Balancing circuit (high power)**

  Minne3 team designed and operation amplifier based balancing circuit, needed to keep every capacitor with the same voltage and prevent damages. Minne4 tested and optimized the design and considered other balancing options. The final design is included in the Minne4 PCB.

- **Solar panel**

  Non solar panel was tested by Minne4, but the designs were made based on a solar panel providing 6 to 24 volts and a input for it was added to the PCB. Other students tested the solar panel while feeding the Minne router, but results have not been published yet.

- **Secondary power source**

  When available, any power source up to 40v DC could feed the Minne4 PCB, which is prepared to receive up to 2 different power sources, such as a regulated voltage coming from the main power grid or a battery. If a battery is used, Minne4 does not designed a charging system for it.

2.3. Power distribution module

Once the system is powered by low and high power sources, the power should be measured and adapted to the charge the super Caps (Vcharge is usually some volts higher than the caps voltage) while powering the load all the time. It is composed by the following sub-modules:

- **Buck converter**

  The system can be fed by a solar panel, or any other DC source from 6v up to 40v, this voltage should me transformed to produce the proper charging voltage for the super capacitors, in the default configuration (8 superCap pairs), they are charged up to 21.6v, but in other configurations this voltage can vary up or down. We tested and documented several circuits and chose the best one for the Minne4 PCB.

- **Boost converter**

  As the buck converter, but increasing, the boost converter produces the proper charging voltagefor the superCaps. Several circuits were taken into account and we kept the basic design, similar in construction to the buck converter.

- **Current limiter**

  The current limiter was planned in Minne2 using a high power resistor. We considered that this resistor will consume useful power, so a more intelligent approach was needed.

The supercapacitor bank offers 21.v volts when fully charged, and a peak current of 3000A, so there is no risk of damage them for high current, but other parts of the circuit does not support high currents. Minne4 monitors the current in both input and output.

Over Discharge Protection

SuperCapacitors have a very low ESR (Eq. Series resistance), BCA3000P have 0.29 mΩ!! [17], when the charger is connected they draw a high charging current (6.3A according to [24], theoretically about 20A), similar to shortcut the circuit. This can cause the charger to burn or damage. For preventing it, Minne2 planned to keep each superCap above 0.67V (with discrete circuits), Minne3 planned a constant charging current measurement and a constant voltage measurement as well. The microcontroller will disconnect the charger or decrease the charging voltage if the current is too high, it will also disconnect the load if any supercap is below 0.67 (or any other settable value).

2.4. Routing module

The main load in the system is the Minne router [4], a low power consumption router designed by KTH. This router is based on Bifrost OS and communicates with the power system through UART serial connection. It is composed by the following sub-modules:

- **Routing Hardware module**

  Minne1 to Minne3 teams designed this 25 Watts router. No tests or improvements were made to the router's hardware.

- **Power Supply Unit Module**

  The router includes a 120W, 12v mini-box picoPSU [18]. It is meant to receive the power from the superCaps (up to 21.6v) and rectify 12V for the router. Minne4 has always thought of replace this PSU by a discrete circuit in the PCB. We have proposed to use a second buck converter in the superCaps output and, driven by the microcontroller, produce the needed voltage for the router. But this design must go after the test and approval of Minne4 Buck converter circuit.

- **Serial Communications module**

  Minne3 developed a protocol for exchanging data with the microcontroller, so the router can know the voltage and current state in the power system. Some measurement values may trigger safety routines in the router, the only developed scripts are for the router soft switching off[19]. Minne4 did not developed any new automatic routine. Minne4

analysed different flashing options for the microcontroller, and by adding the LPC21ISP [21] software to the Bifrost distribution, developed a complete remote flashing procedure through the serial communication so, after log in the router by SSH it is possible to upgrade the microcontroller firmware.

- **Bifrost module**

  As Bifrost OS [21] runs on the router, for testing the flashing procedure and the serial communications module, in Minne4 we made basic bifrost tests and wrote a guide (available in the deliverables) about how to deploy Bifrost OS. Bifrost OS accomplishes the routing task, but it needs developments and new features that Minne4 could not accomplish, such as O&M system, or a graphic interface for the power system.

3. Technical realization and implementation

In this section, we discuss the different considered methods for achieving the goals and we expose the done work for it.

3.1. Microcontroller

3.1.1. Terms Definition

GDB

GDB, the GNU Project debugger, allows you to see what is going on `inside' another program while it executes -- or what another program was doing at the moment it crashed. GDB can do four main kinds of things (plus other things in support of these) to help you catch bugs in the act [27]:

- Start your program, specifying anything that might affect its behaviour.
- Make your program stop on specified conditions.
- Examine what has happened, when your program has stopped.
- Change things in your program, so you can experiment with correcting the effects of one bug and go on to learn about another.

GDB is used here to perform on-Chip debugging over the LPC1768 ARM microcontroller. With the dongle interface and the LPC1768H board connected, having the project set up in Eclipse, OpenOCD running, then Eclipse will launch GDB and will perform on-Chip debugging at the assembly level including step forward, step into, resume, etc.

UART

A UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems and other serial devices.[28]

JTAG

Joint Test Action Group (JTAG) is the common name for what was later standardized as the IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture. It was initially devised for testing printed circuit boards using boundary scan and is still widely used for this application.[31]

3.2. Developing environment

For a detailed guide about how to set up the environment, see "LPC1768H uController Software setup [Windows] v1_0.pdf", in the deliverables list.

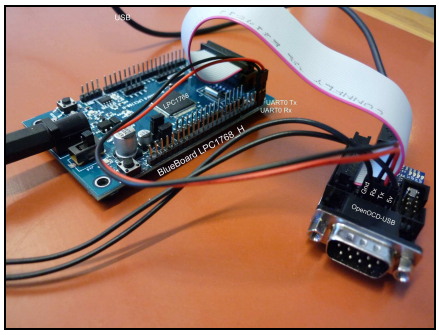
All the logic is implemented both in the router and in a micro controller. The router communicates with the power controller through UART (Universal Asynchronous Transmitter/Receiver). For developing, improving and maintenance of the power controller system, there has to be a working environment for carrying the following tasks:

- To develop new firmware for the power controller
- To test and debug the firmware in real time
- To test the UART communication
- To download new firmware to the power controller

The working environment for this project has always been based on Open Source and GNU resources. In the Minne3 project final report we can find overall instructions for setting the environment using a Unix-based OS, Sourcery G++ Lite for linux, OpenOCD for linux and minicom, more information could be found in the Minne3 Project website [9] and in the project's final report [6], the present document is a guide for setting the environment under Windows XP using free GNU tools, we consider it useful for developers and new co-workers who use Windows as their operative system.

Before flashing the final code in the PCB, during the development stage, it comes mandatory to set up the proper hardware and software developing environment, when the developed softwareir ready, it can be moved to the final PCB.

The chosen micro controller is the ARM Cortex M3 LPC1768 by NXP, the developing hardware in the following image is the LPC1768H board by NGX, connected to the computer through the OpenOCD-USB dongle, by embebed projects. The interface dongle runs OpenOCD , used to program the flash, GDB (GNU Debugger) used to debug on chip, and UART, used to serial communicate to the microcontroller.



LPC1768H (Blueboard) and the Open-OCD dongle (zoom for details)

The used hardware is shown in the picture in the introduction, it is composed of a LPC1768_H development board (Blueboard) and a dongle interface OpenOCD-USB. The Blueboard works with 5 volts and the dongle gets the power from the USB, a simple linking cable can power the blueboard from the dongle instead of using a wall adaptor. The connection is explained in the Minne3 project document “Connecting Hardware Components” [19].

It is also possible (not used here) to power the blueboard with a wall adapter, power the dongle with the USB instead of cascade the power from the computer’s USB port.

- The UART cable is used for UART communication only.
- The JTAG cable is used for flashing, debugging and other JTAG communication.
- The usb cable connects the dongle to the computer’s USB port.

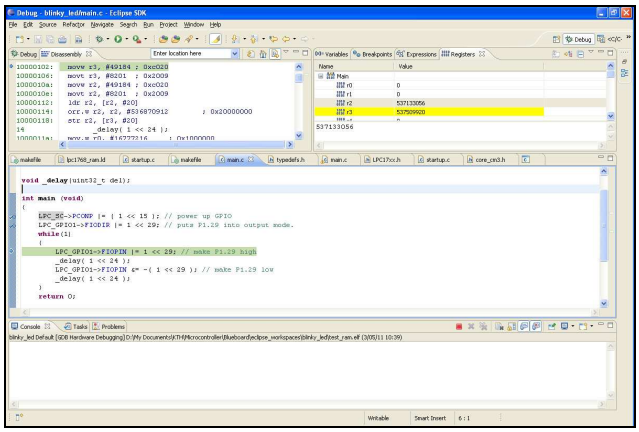
Following the KTH open source initiative, and keeping in mind that the project is intended to low budget networks in developing countries, we preferred GNU software and opened hardware. The project’s writer uses Windows XP and it is also expected that some of the future developers use it sa well, so we planned the software for both system Windows and Ubuntu. We present the chosen software in the following table and a detailed deployment guide “LPC1768H, IDE Software Installation” was also written and included in the deliverables. Everything was meant to act both as result document and also a guide for the future project holders.

	Linux	Windows
Operative System	Ubuntu 2.6.35	Windows XP SP2
IDE	Open Source Eclipse Helios 3.6.2 EPL license. Download: http://download.eclipse.org/eclipse/downloads/drops/R-3.6.2-201102101200/index.php License EPL: http://www.eclipse.org/org/documents/epl-v10.html	
C,C++ plugin	C/C++ Development Tooling (CDT) 7.0.2 Download: http://www.eclipse.org/downloads/download.php?file=/tools/cdt/releases/helios/dist/cdt-master-7.0.2.zip License EPL: http://www.eclipse.org/legal/epl/notice.php	
Toolchain	GNU ARM toolchain for linux	YAGARTO GNU ARM toolchain Download: http://sourceforge.net/projects/yagarto/files/YAGARTO%20for%20Windows/20110318/ License: GNU GPL http://www.yagarto.de/index.html#ncv
GDB, JTAG and OpenOCD	GNU OpenOCD for Linux http://openocd.berlios.de/web/	GNU OpenOCD for Windows Download: http://www.freddiechopin.info/index.php/en/download/category/4-openocd License: GPL: http://www.gnu.org/licenses/gpl-faq.html#WhatDoesGPLStandFor
Serial communication	Minicom	TeraTerm: Download: http://en.sourceforge.jp/projects/ttssh2/releases/ License BSD Free software: http://ttssh2.sourceforge.jp/manual/en/about/copyright.html
Flashing	GNU LPC21ISP	

Chosen development software for Minne4

It was also needed to import Minne3 code, review the ARM function sets, correct the bugs, add libraries, set the compile, make and debug environment, etc. After setting the environment as explained in the written guide, it is possible for the developer to:

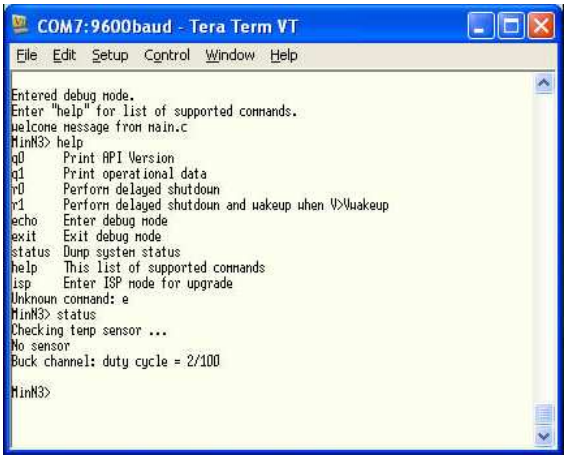
- Write new code for the microcontroller
- Compile, build and debug on chip the code
- Flash the microcontroller
- Communicate to the Minne4 firmware through serial communication



Eclipse IDE while debugging (zoom for details)

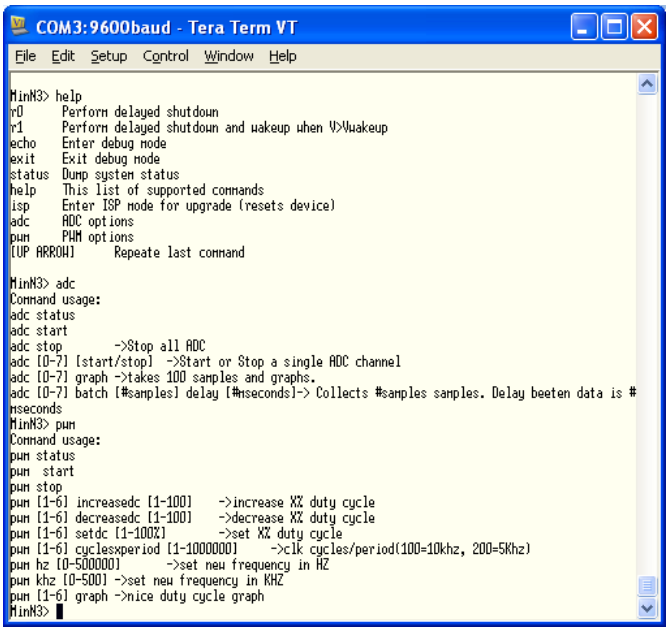
3.3. Serial communication router-microcontroller

Minne3 left a basic serial communication development. When connected to the microcontroller via Minicom, It was possible to run “rustic” commands, as shown in the following image:



Minne3 Command Line Interface(Zoom for details)

In Minne4, we improved the communication protocol, by adding good programming practices and a whole new command set for querying the status, operating the ADC and PWM, flashing the chip, etc. The whole Minne4 command set is shown in the following image:



Minne4 command set in the serial communication (zoom for details)

3.4. ADC Module

For a complete guide about how to use and program the ADC in the LPC1768, see "Minne4 LPC1768 ADC programming.pdf" in the deliverables list.

ADC functions in the project

The LPC1768 micro controller is part of the power system designed for the Tanzanian network, as seen in the introductory image, the system is feed by a solar panel or other external power source, this power must be converted using the buck/boost circuits to charge the super capacitors, which feed the router and other network elements.

The microcontroller uses 3.3v and the buck/boost converter and the hall sensor use 5v, the router uses 12 v regulated by an internal PSU. All the different voltages, the charging current for the superCaps and the load current for the router must be monitorized, so the micro controller can take actions to keep the system working properly.

The microcontroller measures the following voltages and currents:

Voltages	Currents
• Input voltage from the solar panel • Output voltage in the buck converter • Output voltage in the boost converter • Output voltage in the superCaps.	• Input current from the solar panel • Output current to the load

The software implements the following general ADC functions, ready to use in any other project.

ADC	PWM
• Start/Stop the ADC • Start/Stop one ADC channel (8). • Measure the input voltage from any of the 8 ADC channels. • Take several samples and try to find a reliable value. • Prints the ADC status and values • Batch mode: Prints X number of data every Y seconds.	• Start/Stop the PWM • Change duty cycle in any PWM channel (6). • Increase/Decrease duty cycle X % • Change frequency in HZ or KHZ • Set number of clk cycles per PWM period • Prints a nice PWM signal graph

In Minne4 we programmed a whole ADC module, for the microcontroller to reed voltages through its 8 ADC channels and for the user or developer to stop, start or query the channels.

The ADC function set is ready to use in the power system algorithm (not developed yet), and is shown in the following table:

Function	Description
ADC_Start()	Starts ADC in the chip. Does not start any channel.
ADC_Stop()	Stops ADC function in all channels.

ADC_IsChannelStarted(int channel)	Returns 1 if the channel is started.
ADC_StartChannel(int channel)	Starts ADC in a channel.
ADC_StopChannel(int channel)	Stops ADC in a Channel.
Int ADC_ReadChannel(int channel, int reliability)	Returns the ADC value in mvolts. Reads with certain reliability.
Int ADC_IsStarted()	Returns 1 if ADC is started in the chip.
Int ADC_displayTable();	Takes 100 samples and prints a graphs showing that the ADC is not always accurate.
Int ADC_displayGraph(int channel)	Prints a graphs showing that the ADC is not always accurate.
Int ADC_BatchReading(int channel, int times, int reliable, int delayms)	Takes *times* samples. 1 sample every *delayms* milliseconds. Prints a result table.

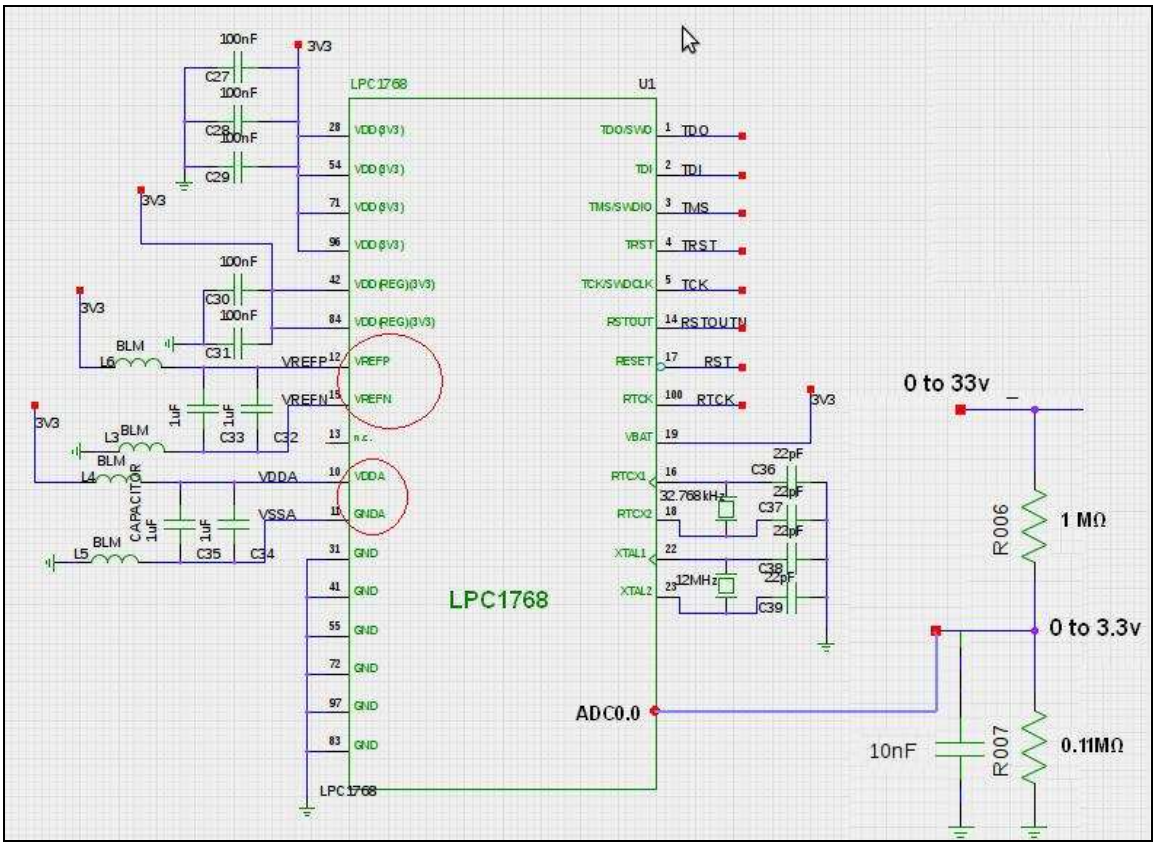
3.4.1.1. ADC Hardware description

The project is developed using the development board LPC1768H blueboard, but the final software will be flashed to the proper PCB. The ADC circuit construction is easy to make, keeping into account the following facts [29]:

- LPC1768 ADC measurement range is 0v to VCC (3.3v)
- The ADC pins support -0.5v to 5.5v.
- The maximum input current should be 100 ma.

To widen the input measurement range, we use a voltage divider, dividing by 10 the input voltage. The final voltage measurement range is 0v to 33v. The 10nF condenser will help to reduce voltage ripple and peaks. This capacitor introduces a low pass filter so it will decrease the allowed measurement speed. In the following picture:

V_{ADC} =0.1Vin V_{ADC}=Vin*R007/(R006+R007) Imax=33v/1M Ω = 33uAmp.



Basic circuitry for working with ADC. Vadc=Vin/10

3.4.2. ADC developed API

It is important to keep close at hand the LPC1768 user manual, Chapter 29: LPC17xx Analog-to-Digital Converter (ADC).

The LPC1768 has the following features:

- 12 bit resolution analog to digital converter,
- 8 input pins, power on/off,
- Power on/off
- Measurement range up to VCC (3.3v)
- Multiple or single Burst convertion reading mode

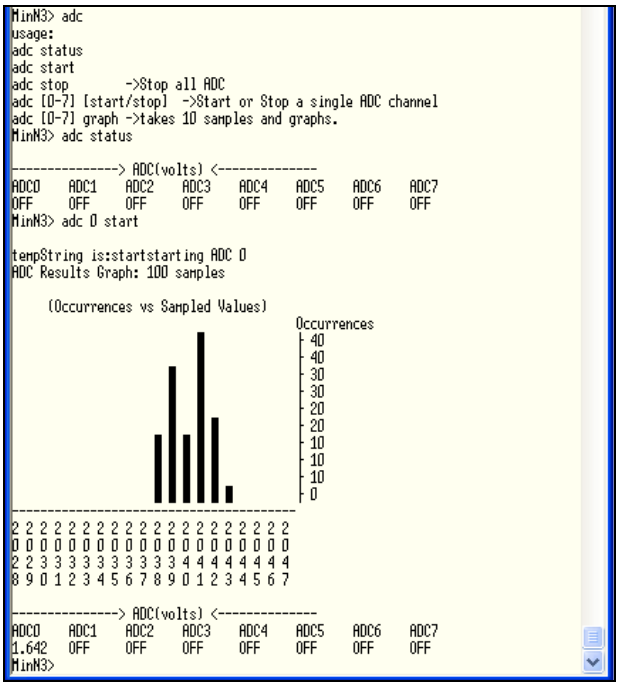
Needed pins:

AD0.0 to AD0.7	Analog inputs, The ADC can read the voltage in any of these 8 pins, the digital function is disabled in a pin when ADC is enabled in it.
V_{REFP}, V_{REFN}	Voltage references, Provide the work range for the convertion. It is usually 3.3v and GND.
V_{DDA}, V_{SSA}	Analog power and ground, they are also 3.3v (isolation is recommended in the user guide but not done here) and GND.

Typical registers:

ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and **the result of the most recent A/D conversion**.
ADINTEN	A/D Interrupt Enable Register. This register contains enable bits that allow the DONE flag of each A/D channel to be included or excluded from contributing to the generation of an A/D interrupt.
ADDR0 to ADDR7	A/D Channel Data Register. This register contains the result of the most recent conversion completed on each from 0 to 7.
ADSTAT	A/D Status Register. This register contains DONE and OVERRUN flags for all of the A/D channels, as well as the A/D interrupt/DMA flag.

The ADC user operation and the voltage scaling used in Minne4 PCB are summarized in the following image.



Minne4 ADC User Operation and scaling measurement circuit (zoom for details)

3.5. PWM module

See the document "Minne4 LPC1768 PWM programming.pdf" in the deliverables list. for a complete guide about how to use and program the PWM in LPC1768.

Similar to the ADC, In Minne4 we programmed a whole PWM module, for the microcontroller to produce PWM signals to drive the buck and boost converter by any of its 6 PWM outputs. The microcontroller was test together with the Buck converter prototype, and we wrote the needed commands for the user or developer to stop, start or query the PWM channels.

3.5.1.1. PWM functions in this project

The LPC1768 micro controller is part of the power system designed for the Tanzanian network, as seen in the introductory image, the system is feed by a solar panel or other external power source, this power must feed the buck/boost circuits to charge the super capacitors with an appropriate voltage and current, The superCaps feed the router and other network elements. The buck/boost circuits are controlled using PWM signals from the microcontroller.

The microcontroller uses 3.3v and the buck/boost converter and the hall sensor use 5v, the router uses 12 v regulated by an internal PSU. All the different voltages, the charging current for the

superCaps and the load current for the router are monitorized by the ADC, so the micro controller can modify the PWM signal to control the buck/bosst voltage output and keep the system working properly.

The PWM function set is ready to use in the power system algorithm (not developed yet), and is shown in the following table:

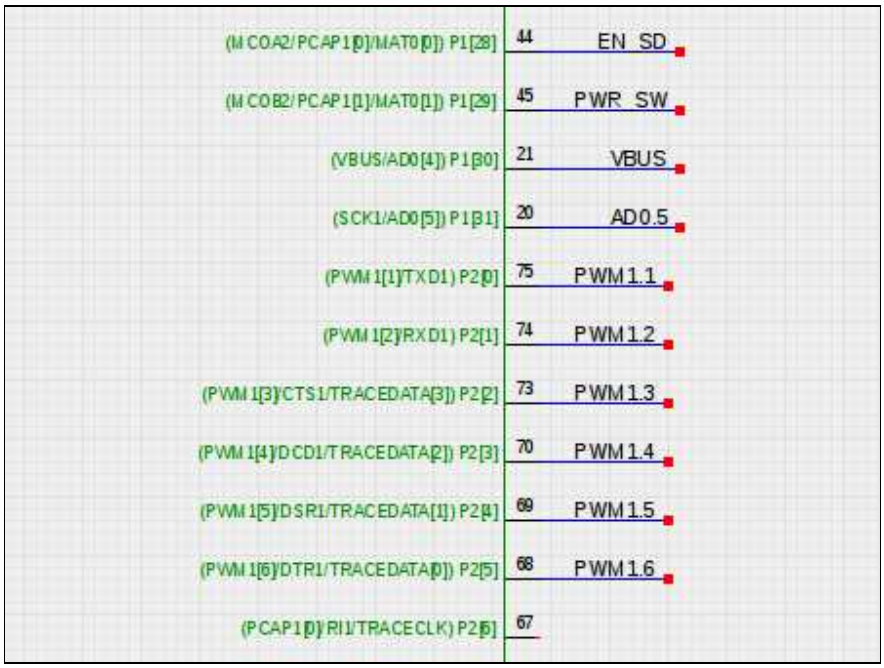
Function	Description
PWM_Start()	Starts PWM in the chip. Also starts the channels.
PWM_Stop()	Stops PWM function in all channels.
PWM_Status()	Prints a table with the PWM status
PWMSetDutyCycle(int channel, int newDutyCycleInt)	Sets a new duty cycle for the given channel
PWM_IncreaseDutyCycle(int channel, int increment)	Increments the new duty cycle for the given channel
PWM_DecreaseDutyCycle(int channel, int increment)	Decrements the new duty cycle for the given channel
PWMSetRate(const int newRate)	Change the number of clk cycles for a PWM period. It means to change the PWM frequency.
PWMsetFrecuency(int frec)	Sets a new PWM frequency in HZ.
PWM_ApplyChanges()	Refresh the PWM settings after new settings.
PWM_printgraph(int dutycycle)	Prints a nice graph for the PWM channel.

3.5.2. PWM developed Hardware

The project is developed using the development board LPC1768H blueboard, but the final software will be flashed to the proper PCB. The PWM circuit construction is easy to make, keeping into account the following facts summarized from the user guide [29]:

- LPC1768 has 6 PWM channels (PWM1 to PWM6)
- The PWM output voltage is 0 or 3.3v.
- There are single or double pulse output.
- Double edge controlled PWM outputs can be programmed to be either positive going or negative going pulse

There is no needed circuitry for the PWM, the 6 outputs can be generated in several different general purpose pins, by default in the following pins:



Used PWM pins in Minne4

3.5.3. PWM developed API

It is important to keep close at hand the LPC1768 user manual, Chapter 28: Chapter 24: LPC17xx Pulse Width Modulator (PWM).

The LPC1768 has the following features:

- Counter or timer operation.
- 6 single edge or 3 double edge outputs. (We use single edge)
- All PWM outputs will occur at the same repetition rate. (They start at the same time)
- A 32-bit Timer/Counter with a programmable 32-bit prescaler
- A capture event may also optionally generate an interrupt.

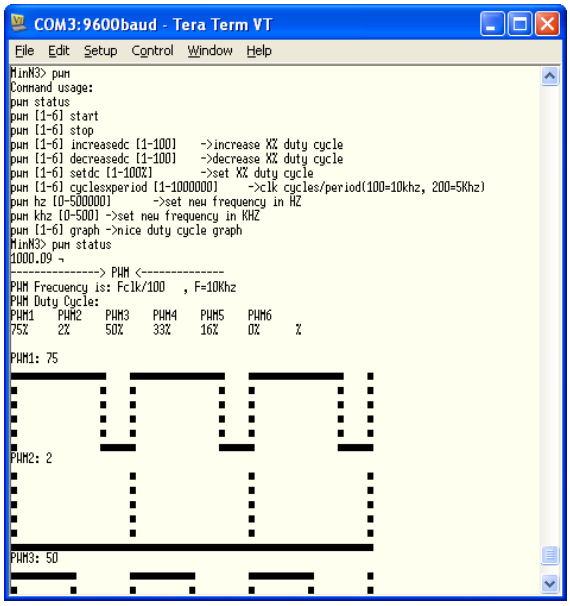
Needed pins:

PWM1.1 to PWM1.6	Analog inputs, The ADC can read the voltage in any of these 8 pins, the digital function is disabled in a pin when ADC is enabled in it.

Typical registers:

IR	Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.
TCR	Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR
MCR	Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs
MR0	Match Register 0. MR0 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC sets any PWM output that is in single-edge mode.
MR1 to MR6	Match Register 1 to 6. MR1/6 can be enabled in the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt when it matches the TC. In addition, a match between this value and the TC clears PWM1 in either edge mode.
PCR	PWM Control Register. Enables PWM outputs and selects PWM channel types as either single edge or double edge controlled.
LER	Load Enable Register. Enables use of new PWM match values.

The PWM user operation is summarized in the following image.

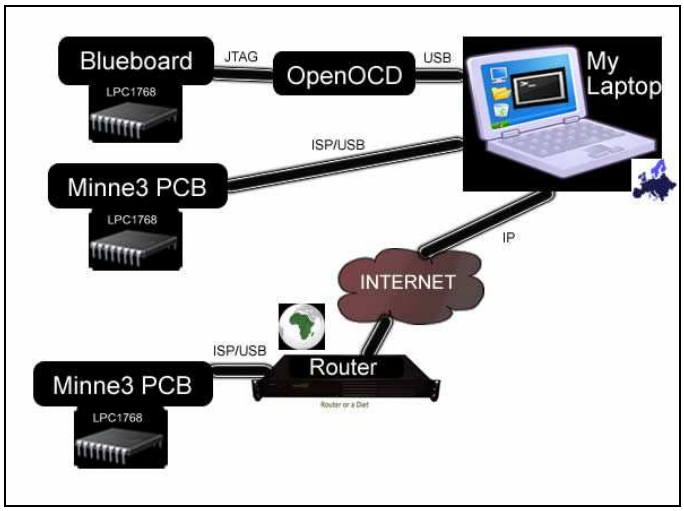


Minne4 PWM User Operation (zoom for details)

3.6. Firmware flashing module

The developed code has to be flashed in every produced PCB. We tested the different scenarios, including an option where the PCB could be remotely flashed after SSH login to the router, and we decided to don't include JTAG in the final PCB, because it is most useful in the developing stage than in the production one. We tested the three ways LPC1768 could be flashed:

- Using OpenOCD and an OpenOCD dongle as an interface for sending JTAG [22] commands to the LPC1768.
- Using Internal System Programming (ISP) [23], an LPC1768 on chip feature for flashing it.
- Making the device appear as an external storage device in Windows and copy there the .bin file.



Minne4 flashing scenarios

3.6.1.1. Flashing with JTAG

When the LPC1768 is in the blueboard, we can connect the JTAG cable, run OpenOCD in the computer and type some commands for flashing the .bin or .hex in the chip.

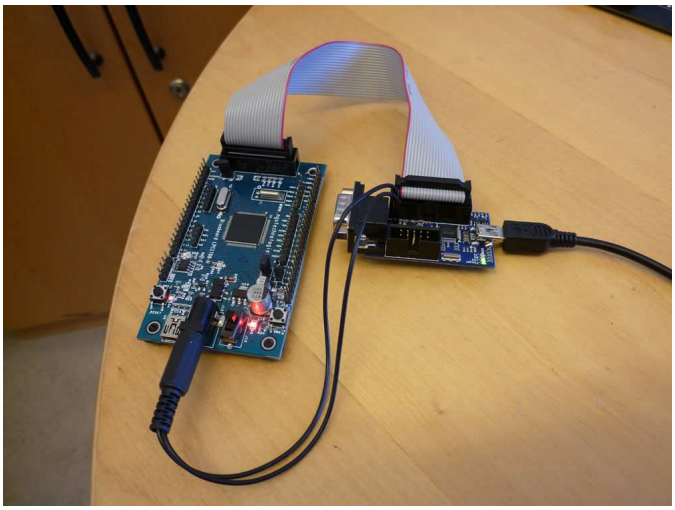
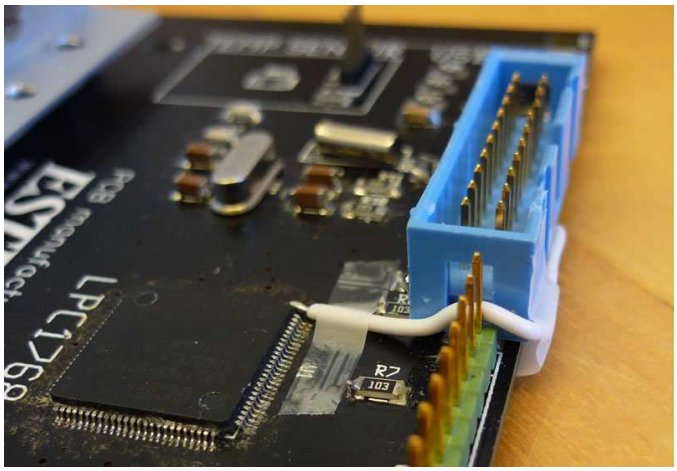


Image: Blueboard LPC1768H and the OpenOCD-USB dongle.

In the Image, the power comes from the USB, so the data. 5v are passed to the Blueboard and the 20 pin JTAG connector for flashing or debugging.

We tested Minne3 PCB (in the picture below), When the LPC1768 is in a PCB, the 20 pin JTAG connector should be mounted for JTAG features, such as flashing or on chip debugging, it will however, consume some space and components, adding complexity to the circuit.



Minne3 JTAG connector in PCB for minne3, the DTO line was manually cabled for testing.

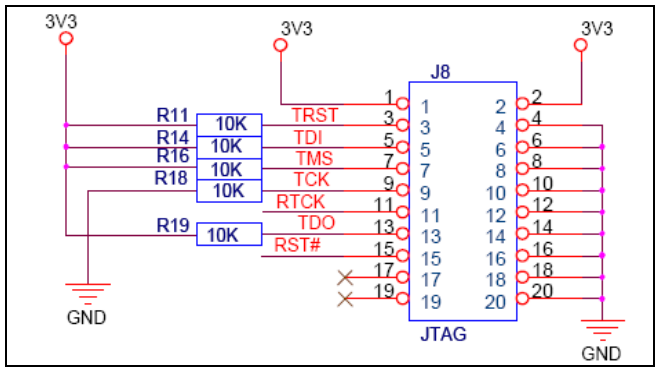


Figure: JTAG schematics used in the Blueboard.

3.6.1.2. Flashing with ISP

Internal System Programming is a way to write the flash memory through the UART0 ports in the microcontroller, when entering the bootloader after a reset, or launching the ISP mode from a already running user code. The bootloader controls initial operation after reset and also provides the tools for programming the flash memory. This could be initial programming of a blank device, erasure and re-programming of a previously programmed device, or programming of the flash memory by the application program in a running system [29].

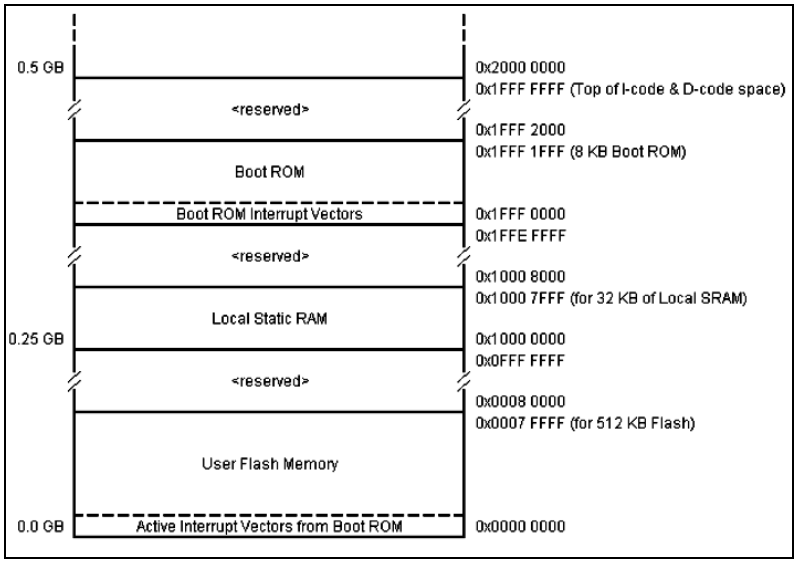


Figure: Lower map of the memory

The bootloader is launched after reset, it will check some time (3ms) if there is any ISP mode request, if there is not, then it will launch the user code, if there is any. In LPC17XX, the ISP mode is entered when the P2.10 pin (pin 53) is low when the chip restarts, it could be done either manually setting p2.10 low, then reset and then release P2.10, or anytime, an external software, i.e. running on Linux or Windows sets the needed voltages in pin P2.10 and nReset pin (pin 17).

After either way, the chip will enter the ISP mode and will wait for instructions through the UART0 interface (pin 98 and pin 99).

In normal cases, to perform a normal restart, when ISP mode is not needed, ISP pin should go to a resistor and 3.3v, that will avoid unexpected entrances to ISP mode, and an optional switch may be used to enter the ISP mode.

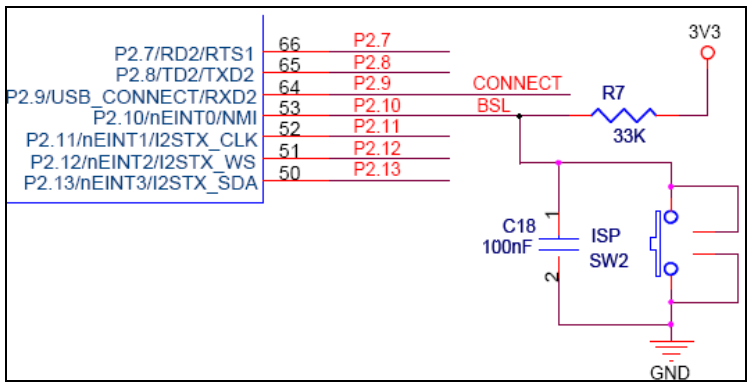


Figure: The circuitry used for ISP in the Blueboard

Once in the ISP mode, the LPC17XX waits for the the auto-baud routine to synchronize with the host via serial port 0. The host should send a “?” (0x3F) as a synchronization character and wait for a response. The host side serial port settings should be 8 data bits, 1 stop bit and no parity.

The auto-baud routine measures the bit time of the received synchronization character in terms of its own frequency and programs the baud rate generator of the serial port. It also sends an ASCII string ("Synchronized<CR><LF>") to the host. In response to this the host should send the same string ("Synchronized<CR><LF>"). The auto-baud routine looks at the received characters to verify synchronization. If synchronization is verified then "OK<CR><LF>" string is sent to the host. The host should respond by sending the crystal frequency (in kHz) at which the part is running. For example, if the part is running at 10 MHz, the response from the host should be "10000<CR><LF>". "OK<CR><LF>" string is sent to the host after receiving the crystal frequency. If synchronization is not verified then the auto-baud routine waits again for a synchronization character. For auto-baud to work correctly in case of user invoked ISP, the CCLK frequency should be greater than or equal to 10 MHz.

If, for testing, the autobaud procedure is manually done using serial communication software with default settings (9600, 8, n,1), the console would appear like this:

```
?
Synchronized
Synchronized
OK
```

That procedure could be used to test the ISP mode and the serial connection, however, once in the command line, there are no useful operations to be done by hand, so the following commands are for your reference only but they won't be used in a manual way, they will be executed by the flasher software running in the host computer.

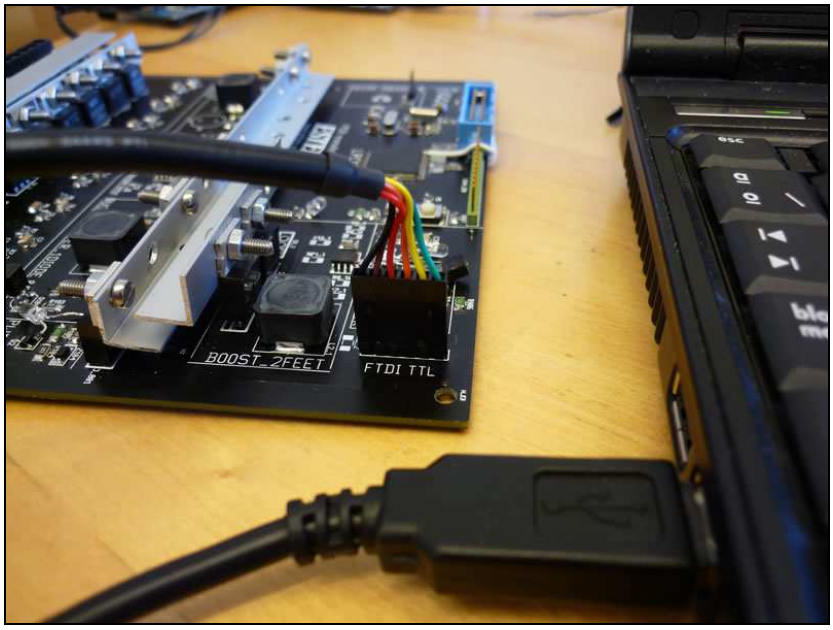
3.6.1.3. ISP Command Usage:

We tested the following ISP commands, and we decided that they should not be used for the end user and they are not necessary in the development neither.

Unlock: U <Unlock Code>
Set Baud Rate: B <Baud Rate> <stop bit>
Echo: A <setting>
Write to RAM: W <start address> <number of bytes>
Read Memory: R <address> <number of bytes>
Prepare sector(s) for write operation: P <start sector number> <end sector number>
Copy RAM to Flash: C <flash address> <RAM address> <number of bytes>
Go: G <address> <Mode>
Erase sector(s): E <start sector number> <end sector number>
Blank check sector(s): I <start sector number> <end sector number>
Read Part ID: J
Read Boot Code version: K
Read serial number: N
Compare: M <address1> <address2> <number of bytes>

So, we mounted Minne3 PCB elements, corrected the bugs and tested ISP. The microcontroller can be flashed and the UART serial communications works fine.

3.6.2. Hardware used for ISP Flashing

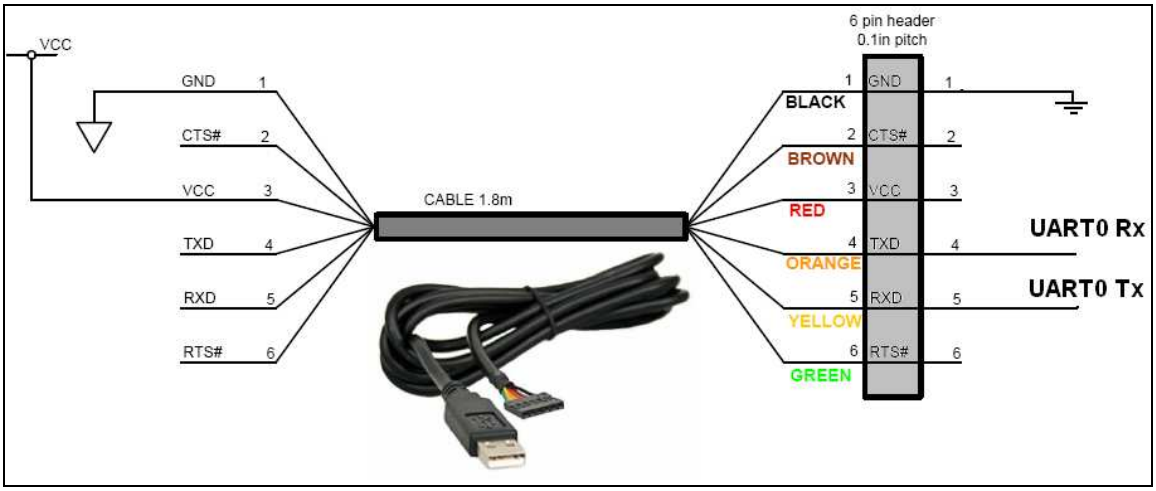


UART0 pins (98, 99) have been connected to the FTDI TTL port in the minne3 PCB, as well as the GND and GCC (5v), that is enough for performing serial communication with another equipment.

The FTDI cable used is a TTL-to-USB converter by FTDI, TTL-232R-3v3, http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf

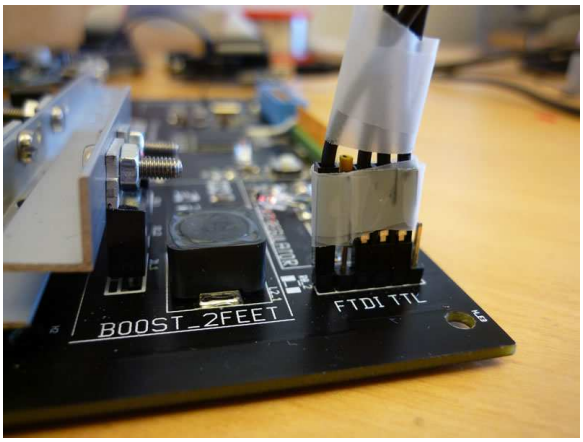
In the previous image we can see the pin out and the colour code, from the 6 available cables, we need GND, TXD and RXD only.

The circuit is in the USB connector and it is shown in the datasheet also:



The TTL-232R-3v3 diagram

In the Minne3 PCB, Tx and Rx were mistaken and it was necessary to switch them. Using manual grounding of the ISP pin and pressing the reset button, it is possible to enter the ISP mode and flash the microcontroller in the PCBv1 board.



Bug example: TX-RX switched in Minne3 PCB

3.6.3. ISP Developed API

Entering to the ISP mode in the minne3PCB

In the PCBv1 There is none switch attached to the ISP pin but there is a reset switch, both are needed to perform a manual entering to the ISP mode, it is also desirable to be able to perform upgrades remotely, from the minne3 router, without having anyone to push the buttons.

Entering the ISP mode by hardware

Flash Magic and LPC21ISP could send the appropriate voltage sequence to ISP and RST pins through the RS232 pins DTR and RTS, but we don't have DTR output in the FTDI3v3 cable. It is an option to replace the cable and use a universal USB to RS232 (3.3v) cable, which implies to include a RS232 db9 jack in the PCB.

Entering the ISP mode by software

The second option is to enter ISP mode from the already running user code, namely the minne3 code. It is done using a ARM feature called **IAP** (In Application Programming), which allows to write the Flash memory as well as call the ISP mode.

The “Enter ISP mode” function has been added to the minne3 communication menu, so it is possible to jump to ISP mode from the running code, then upgrade it and then launch the new firmware version.

```
MinN3> help
q0      Print API Version
q1      Print operational data
r0      Perform delayed shutdown
r1      Perform delayed shutdown and wakeup when V>Vwakeup
echo    Enter debug mode
exit    Exit debug mode
status  Dump system status
help    This list of supported commands
isp     Enter ISP mode for upgrade
```

After type ‘isp’ the LPC1768 will enter the ISP mode, then wait for the synchronise character ‘?’ to be send by FlashMagic or LPC21ISP.

Exiting the ISP mode

The main idea of entering the ISP mode is to flash the new firmware in the device, so we can perform remote upgrades, then it is critical to be able to start the new uploaded code or resume the program if entered to ISP by mistake, for example, when typing “isp” in the minne3 command line. After the LPC1768 enters the ISP mode, it will stay there until:

- It receives a Reset through its RST pin.
- It receives a command to jump to address 0x00000000 (where the user code is).
- A watchdog timer expires and the microcontroller resets itself.

For the first option, avoiding to push the RST button, it is possible to cable the RST pin to the RST line in the RS232 cable, but we should assure that the RST line is not used by mistake by the computer connected to the PCB, for example, in a equipment reset, or when running a 3rd part application.

For the second option, it consists on send a “GO 0 T” (go to address 0) command from the ISP command line, it is also automatically sent at the end of the firmware upload, but while testing we found that it does not work for the minne3 program, but it works for smaller programs.

The third option has been added to the code, with a configurable timer, the microcontroller will reset X seconds after it gets to an unusual state, i.e. the ISP mode. If we set that timer to 10 min, it mins that the operator has 10 minutes to upgrade the software, after that time, the microcontroller will reset and the new code will be applied.

In Minne4, we developed ISP and the watchdog is set before entering ISP mode with this simple code:

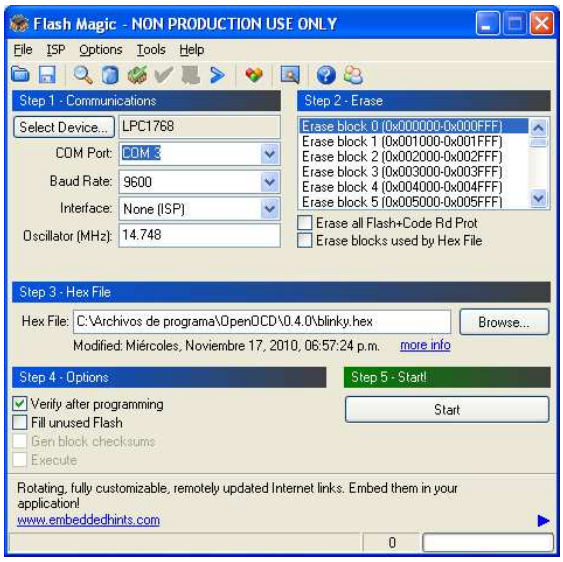
```
//set the number of wd ticks to wait (1wd tick = 1/16 clock ticks)
LPC_WDT->WDTC = 100000000; //90 sec aprox
//activate the watchdog
LPC_WDT->WDMOD = 0x03;
//Start the watchdog
LPC_WDT->WDFEED = 0xaa;
LPC_WDT->WDFEED = 0x55;
```

3.6.4. Software evaluation for ISP

3.6.4.1. Flash Magic

Flash Magic is a windows based NXP software made by Embedded Systems Academy, intended for using ISP functions in ARM microcontrollers, the features listed it the manual are:

- Erasing the Flash memory (individual blocks or the whole device)
- Programming the Flash memory
- Modifying the Boot Vector and Status Byte
- Reading Flash memory
- Performing a blank check on a section of Flash memory
- Reading the signature bytes
- Reading and writing the security bits
- Direct load of a new baud rate (high speed communications)
- Sending commands to place device in Bootloader mode



Flashmagic can be downloaded from http://www.flashmagictool.com/.

3.6.4.2. LPC21ISP

Lpc21ISP is a GNU software intended to easily flash some microcontrollers over linux or windows, using the ISP (Internal System Programming) feature, it is an alternative to the windows FlashMagic software. At the moment it is not a well known software and difficult to find, despite we tried it and it works quite fine. Lpc21ISP is found in both source code or the windows executable. With the source code, it could be built for several linux distributions.

Getting LPC21ISP:

Lpc21ISP project website is http://sourceforge.net/projects/lpc21isp/, but all the files and the discussions are hold in a Yahoo group, you should join the group to download files. The used version, also the last one, can be found here:

http://groups.yahoo.com/group/lpc21isp/files/Beta%20versions/lpc21isp_181.zip

LPC21ISP v 1.81, including the windows .exe, the Ubuntu executable and 2 test .bin files can be found in the CJCM website:

http://web.ict.kth.se/~cjcm/how_To_flash/lpc21isp_181_deployed_to_ubuntu_and_winxp.rar

Lpc21ISP for windows:

Windows XP is used here, It is only necessary the lpc21isp.exe file, included in the lpc21isp package. There is no manual for the software, but a explanation message when executing it:

```
>lpc21isp
Portable command line ISP for NXP LPC2000 family and Analog Devices ADUC 70xx
Version 1.81 compiled for Windows: Apr  4 2011, 01:07:55
Copyright (c) by Martin Maurer, 2003-2009, Email: Martin.Maurer@clibb.de
Portions Copyright (c) by Aeolus Development 2004, www.aeolusdevelopment.com

Syntax:  lpc21isp [Options] file[ file[ ...]] comport baudrate Oscillator_in_kHz

Example: lpc21isp test.hex com1 115200 14746

Options: -bin        for uploading binary file
       -hex        for uploading file in intel hex format (default)
       -term       for starting terminal after upload
       -termonly   for starting terminal without an upload
       -localecho  for local echo in terminal
       -detectonly  detect only used LPC chiptype (PHILIPSARM only)
       -debug0     for no debug
       -debug3     for progress info only
       -debug5     for full debug
       -donotstart  do not start MCU after download
       -try<n>     try n times to synchronise
       -wipe       Erase entire device before upload
       -control    for controlling RS232 lines for easier booting (Reset = DTR, EnableBootLoader = RTS)
       -controlswap swap RS232 control lines   (Reset = RTS, EnableBootLoader = DTR)
       -controlinv  Invert state of RTS & DTR    (0=true/assert/set, 1=false/deassert/clear).
       -verify     Verify the data in Flash after every writes to sector. To detect errors in writing to Flash ROM
       -logfile    for enabling logging of terminal output to lpc21isp.log
       -halfduplex  use halfduplex serial communication (i.e. with K-Line)
       -ADARM      for downloading to an Analog Devices  ARM microcontroller ADUC70xx
       -PHILIPSARM  for downloading to a microcontroller from  NXP(Philips) LPC13xx/LPC17xx/LPC2000 family (default)
```

Lpc21ISP for Linux :

In Linux, there is no executable for LPC21ISP, it comes in its source code, so it could be deployed in any target Linux using make.

Installing LPC21ISP in a Linux which has *make*

LPC21ISP has to be deployed to the target Linux from the source code inside the downloaded package, the process is easy (skip this step if you already have the linux executable for your linux distribution).

For installing LPC21ISP into a linux which has *make:* (Bifrost does not have make)

1. Extract LPC21ISP to some directory
2. Go to the LPC21ISP directory, then type:

   make -f Makefile clean all

```
root@ubuntu:/mnt/hgfs/KTH/Microcontroller/Blueboard/ISP stuff/ISP for linux/lpc21isp_181# make
-f Makefile clean all
rm -f adprog.o lpcprog.o lpcterm.o lpc21isp
gcc  -Wall -c -o adprog.o adprog.c
gcc  -Wall -c -o lpcprog.o lpcprog.c
gcc  -Wall -c -o lpcterm.o lpcterm.c
gcc  -Wall -o lpc21isp lpc21isp.c adprog.o lpcprog.o lpcterm.o
```

3. LPC21ISP is now installed. Execute the new created file lpc21isp

   ./lpc21isp

Installing lpc21isp in Bifrost

LPC21ISP comes in its source code, so it could be deployed in any target linux. The standard Bifrost does not include make or gmake, needed to “make” the executable binary lpc21isp. The solution for installing new software to Bifrost is to use wget and download the software from the internet, or to make the compatible binaries in another linux and bring them to Bifrost.

To create a Bifrost compatible binary, we should create an executable including the dependent libraries:

make CFLAGS=-static

```
root@ubuntu:/mnt/hgfs/KTH/lpc21isp# make CFLAGS=-static
gcc  -static -c -o lpcprog.o lpcprog.c
gcc  -static -c -o lpcterm.o lpcterm.c
gcc  -static -o lpc21isp lpc21isp.c adprog.o lpcprog.o lpcterm.o
```

The result is a file called lpc21isp. It includes the dependencies needed in Bifrost.

Finding the USB port name

In Linux (Ubuntu 2.6.35 is used here), after connecting the FTDI TTL cable, you could find the interface name using the dmesg command:

```
root@ubuntu:/# dmesg |grep FTDI
[369227.113885] USB Serial support registered for FTDI USB Serial Device
[369227.115493] ftdi_sio 2-1:1.0: FTDI USB Serial Device converter detected
[369227.123014] usb 2-1: FTDI USB Serial Device converter now attached to ttyUSB0
[369227.124770] ftdi_sio: v1.6.0:USB FTDI Serial Converters Driver
```

*note: it has happened that the system has to be restarted for detecting the USB cable.

TODO: how to force the USB device detection without restarting.

3.6.4.3. Testing the UART connection:

1. If the minne3 firmware is running in the microcontroller, you should see the minne3 prompt when you connect using #minicom -o.

2. If there is a unknown firmware running, then enter ISP mode in the PCB (or blueboard) by manually setting down the ISP pin and then push the RST switch, then enter minicom using:

   #minicom -o

   Optionally, you manually force the sync process (as described above in the ISP theory), type '?' and [enter], then the uController will answer 'Synchronized', you should type Synchronized' again and [enter], you will receive an 'OK', and you will be logged in the ISP mode. [So you have test it, LPC21ISP will work fine].

```
Welcome to minicom 2.4

OPTIONS: I18n
Compiled on Jun  3 2010, 13:46:31.
Port /dev/ttyUSB0

Press CTRL-A Z for help on special keys
?
Synchronized
OKnchronized
OK
```

3.6.5. Flashing software evaluation results

The different tests and results can be summarized in the following table:

	OS	License	Hardware	Comments	Remotely?	
External Storage		GNU	Usb cable	Drag & Drop, uses sec bootloader	No	
JTAG		GNU	OpenOCD dongle + USB cable + 20 pin conenctor	Allows debugging	No	
FlashMagic		Free for development	FTDI USB 6pin cable	Easy to use. Hex files only	No	
LPC21ISP		GNU	FTDI USB 6pin cable	Bin&hex files	Yes, from Bifrost router	

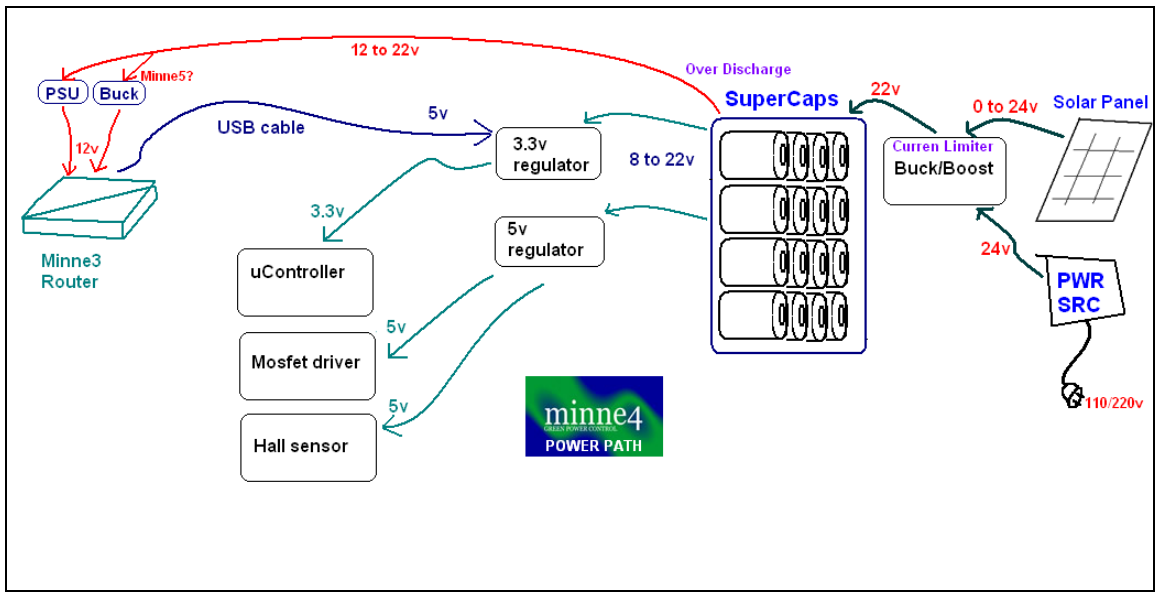
The chosen option consists on, from any host in the internet send the firmware file to the router through SFTP and SSH login to the router, use the serial UART communication to run the "ISP" command in the PCB command line and then run the flashing software LPC21ISP, which will reset the device after flashing, when entering ISP mode, a watchdog is activated to reset the device after a settable time (5 min) if there is a failure in the process.

See the deliverables list for the complete guide "Minne4 How to flash the LPC1768.pdf"

3.7. Power supply system

In Minne4, we tested Minne3 design and its PCB and we made some improvements to the power flow. Minne4 system developed the following schema in the PCB designed:

- The router is all the time powered by the superCaps (on-line system).
- A solar panel and/or a secondary power source feed the Buck/Boost converter.
- The Buck/Boost converter produce the charging voltage for the superCaps
- 3.3v and 5v regulators are all the time powered by the superCaps
- A FTDI USB cable and/or the superCaps power the uController.



Minne4 Power suppy system.

- When there is power from the main power grid, the superCaps get the power from it.
- When a sudden break-out occurs in the main power grid, and **there is sun**, the solar panel feeds the superCaps
- When a sudden break-out occurs in the main power grid, and **there is no sun**, the superCaps are on their own with the load and starts discharging until there is not enough power (2 hours aprox.) Then the router, thanks to the data received from the uController, switchs off.

The different scenarios are summarized in the following table:

Main power grid	SUN power*	SuperCaps state**	Result
ON	ON	Charged	Router OK.
ON	ON	Discharged	superCaps charging. Router OK.
ON	OFF	Charged	Router OK.
ON	OFF	Discharged	superCaps charging. Router OK.
OFF	ON	Charged	Router OK.
OFF	ON	Discharged	superCaps charging. Router OK.
OFF	OFF	Charged	superCaps Discharging. Router OK.
OFF	OFF	Discharged	Router OFF

*Solar panel power> 30W
**SuperCaps voltage > 15v, I=2A (P>30W)

3.7.1. 3.3V and 5v regulators

For more information, see the document "Minne4 regulators.pdf" in the deliverables list. The regulators for the system were designed to accomplish the following:

- Stable output
- Low noise, short and thick paths to the load
- Enough output power for the Mosfer driver, hall sensors, uController
- "ON" indicators
- Additional OUT/IN pins for testing or powering external loads.
- Input could be the superCaps, the 5V USB cable or a external power system.

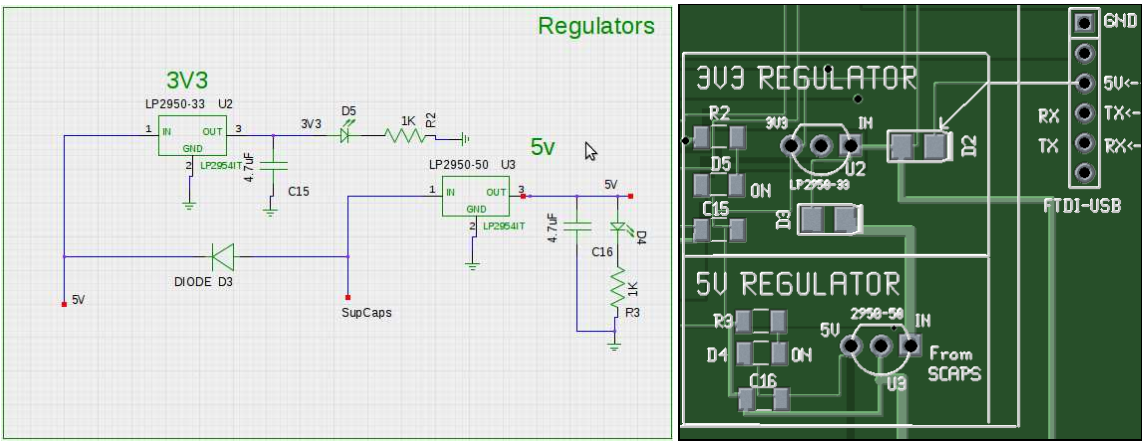
The different tested regulators are described in the following table:

Part	Description	Comments
LP2951D 	30V; Output Voltage:Adjustable 1.23V to 30V; No. of Pins:8; Output Current:100mA; Max Vdropout(Vin-Vout)=600mv	Can also give 5v fixed in the output. Tested in PCB

	http://www.ti.com/lit/ds/slvs582g/slvs582g.pdf	
LP2950-33 	Output Voltage: 3.3V; No. of Pins: 3; Output Current:100mA; Case Style:TO-92; Max Vdropout(Vin-Vout)=600mv http://www.ti.com/lit/ds/symlink/lp2951-33.pdf	Tested in the PCB
LP2950-50 	Output Voltage: 5V; No. of Pins:3; Output Current:100mA; Max Vdropout(Vin-Vout)=600mv http://www.ti.com/lit/ds/symlink/lp2951-50.pdf	

Minne4 tested regulators

Minne3 Used the LP2951D regulator, but we decided to use the LP2950-33/50 instead of the LP2951 because LP2950 provides the same output current and it uses only 3 pins, giving either 3.3 or 5v in a very simple way. The final design is showed in the following image:

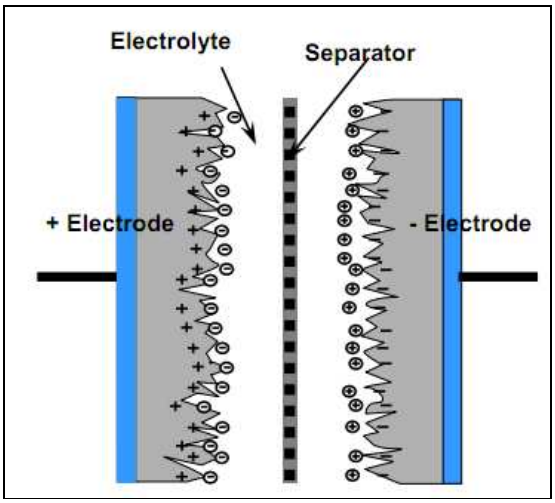


Minne4 regulation system schematics and PCB design

Produced deliverables: Minne4 regulators.pdf, gEDA schematics and PCB files.

3.7.2. Super Capacitors

Minne3 team designed a 8 pairs x 2.7v super capacitors bank, producing 21.6v when fully charged, in Minne4, we kept the bank design and we performed and documented charging and discharging tests, also, we re-designed the balancing circuit and analysed other balancing options, for more information see the document “Minne 4 SuperCaps analysis-pdf” in the deliverable list.



The Super Capacitors construction[34]

The Theory behind super capacitors is better explained in the manufacturer's guide [34], Electrochemical double layer capacitors (EDLCs) are similarly known as supercapacitors or ultracapacitors. An ultracapacitor stores energy electrostatically by polarizing an electrolytic solution. Though it is an electrochemical device there are no chemical reactions involved in its energy storage mechanism. This mechanism is highly reversible, allowing the ultracapacitor to be charged and discharged hundreds of thousands to even millions of times. An ultracapacitor can be viewed as two non-reactive porous plates suspended within an electrolyte with an applied voltage across the plates. The applied potential on the positive plate attracts the negative ions in the electrolyte, while the potential on the negative plate attracts the positive ions. This effectively creates two layers of capacitive storage, one where the charges are separated at the positive plate, and another at the negative plate.

Conventional electrolytic capacitors storage area is derived from thin plates of flat, conductive material. High capacitance is achieved by winding great lengths of material. Further increases are possible through texturing on its surface, increasing its surface area. A conventional capacitor separates its charged plates with a dielectric material: plastic, paper or ceramic films. The thinner the dielectric the more area can be created within a specified volume. The limitations of the thickness of the dielectric define the surface area achievable.

An ultracapacitor derives its area from a porous carbon-based electrode material. The porous structure of this material allows its surface area to approach 2000 square meters per gram, much greater than can be accomplished using flat or textured films and plates. An ultracapacitors charge separation distance is determined by the size of the ions in the electrolyte, which are attracted to the charged electrode. This charge separation (less than 10 angstroms) is much smaller than can be accomplished using conventional dielectric materials. The combination of

enormous surface area and extremely small charge separation gives the ultracapacitor its outstanding capacitance relative to conventional capacitors.

Typical Applications

Pulse Power [34]

Ultracapacitors are ideally suited for pulse power applications, due to the fact the energy storage is not a chemical reaction, the charge/discharge behaviour of the capacitors is efficient. Since ultracapacitors have low internal impedance they are capable of delivering high currents and are often times placed in parallel with batteries to load level the batteries, extending battery life. The ultracapacitor buffers the battery from seeing the high peak currents experienced in the application. This methodology is employed for devices such as digital cameras, hybrid drive systems and regenerative braking (for energy recapture).

Bridge Power

Ultracapacitors are utilized as temporary energy sources in many applications where immediate power availability may be difficult. This includes UPS systems utilizing generators, fuel cells or flywheels as the main power backup. All of these systems require short start up times enabling momentary power interruptions. Ultracapacitor systems are sized to provide the appropriate amount of ride through time until the primary backup power source becomes available.

Main Power

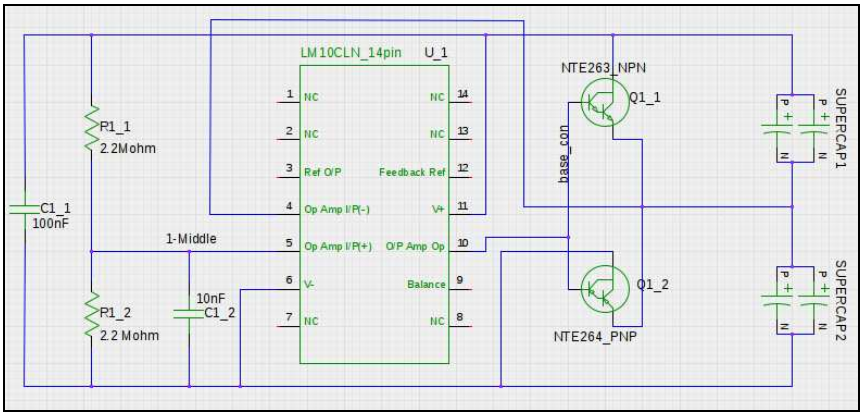
For applications requiring power for only short periods of time or is acceptable to allow short charging time before use, ultracapacitors can be used as the primary power source. Examples of this utilization include toys, emergency flashlights, restaurant paging devices, solar charged accent lighting, and emergency door power.

Memory Backup

When an application has an available power source to keep the ultracapacitors trickle charged they may be suited for memory backup, system shutdown operations, or event notification. The ultracapacitors can be maintained at its full charged state and act as a power reserve to perform critical functions in the event of power loss. This may include AMR for reporting power outage, micro-controllers and board memory. [34]

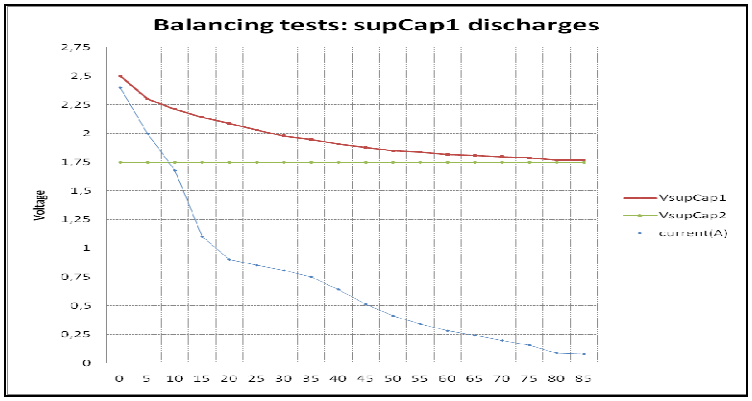
3.7.3. Minne4 balancing circuit

Minne2 team designed a balancing circuit, included in the Minne3 PCB (2010) and tested by us in Minne4, we analysed it and mounted the elements and made some tests, corrected bugs, made modifications and wrote conclusions, we also tested other balancing options but we decided to keep Minne2 design and correct the bugs in Minne3. The tested circuit designed by Minne2 team is:



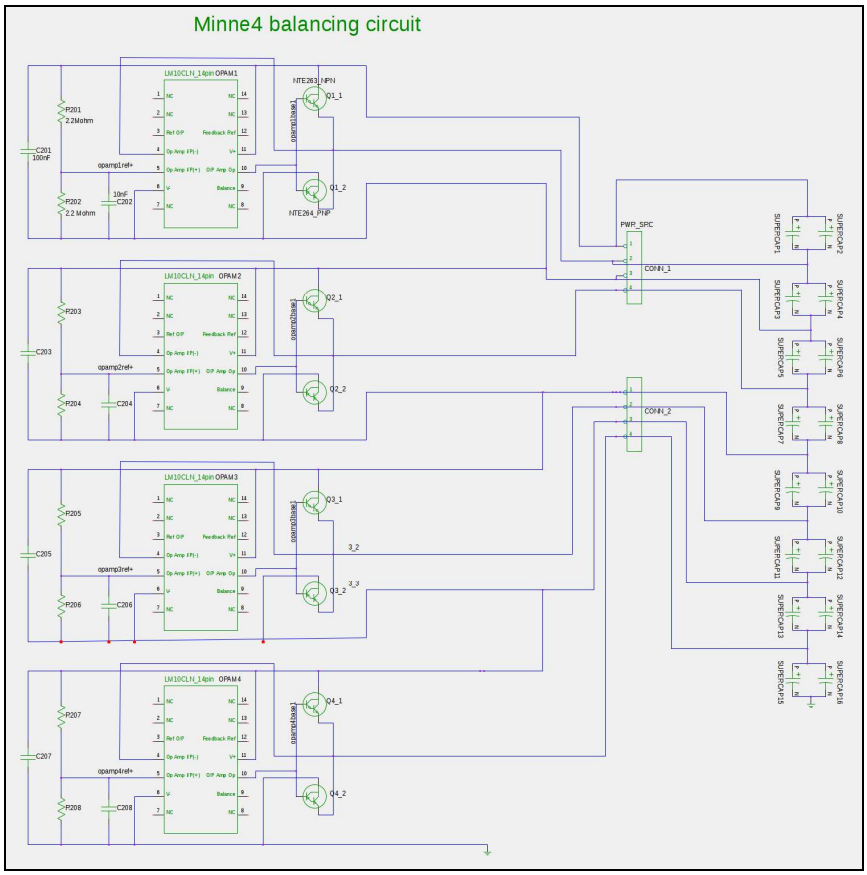
4 superCaps balancer in Minne2, Minne3 and Minne4

The results were as expected, the most charged superCap couple will discharge until it reaches the other superCap couple. The second superCap will keep its current voltage.



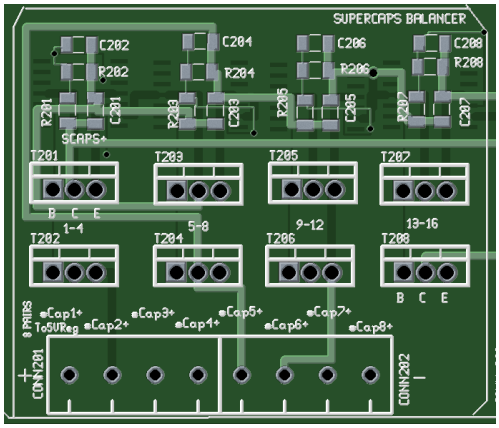
SupCap1 discharges to reach VsupCap2

Compared to Minne3, in Minne4 We reduced from 8 to 4 the operational amplifiers, we reduced from 6 to 2 the needed connectors and we have correctly renamed the elements and networks, the final schematic is shown in the following image.



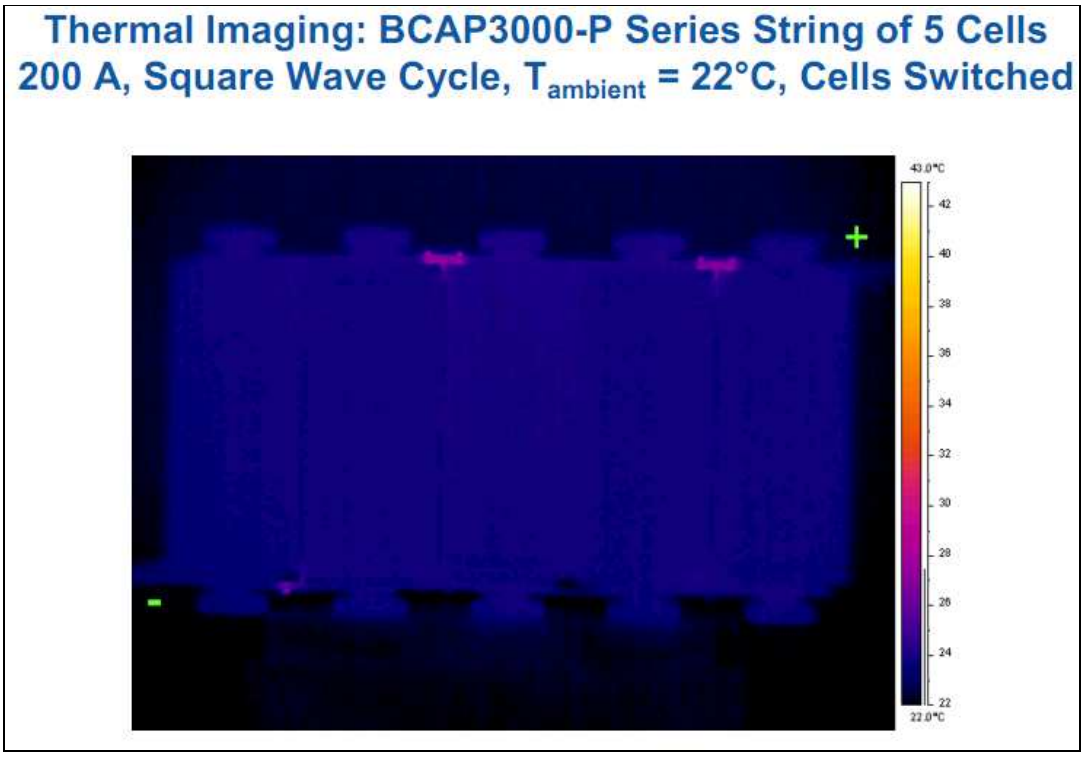
Minne4 Balancing circuit

The new Minne4 PCB design is shown in the following image (zoom for details), smart and simple design, from 2 Up to 16 super capacitors and up to 5A balancing current, also opAMPS are in the back. Produced deliverables: Minne4 superCaps analysis.pdf, gEDA schematics and PCB files.

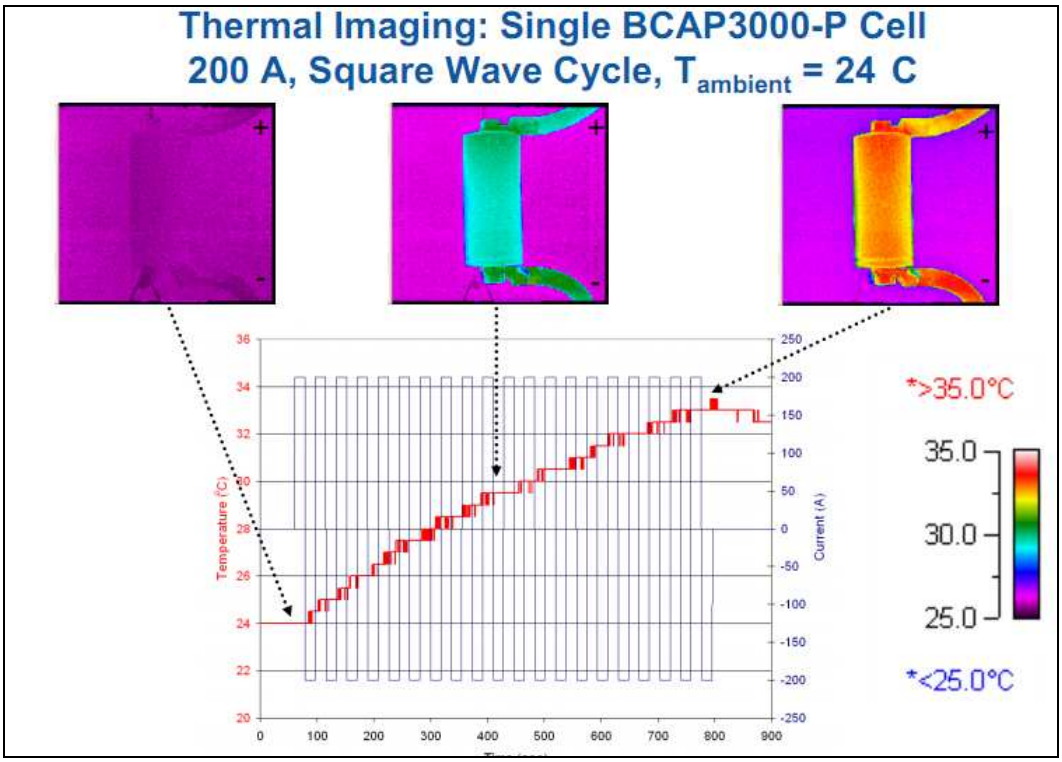


Temperature tests

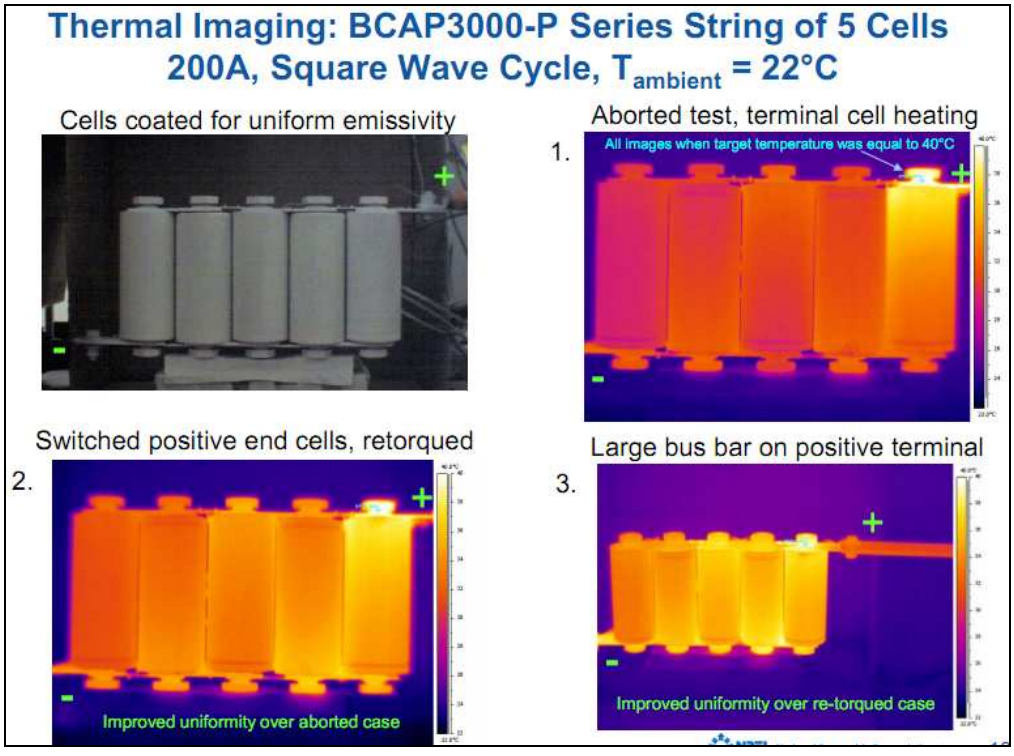
We did not perform any deep thermal test, since under normal operation the heating is not perceptible, in fact, we found that superCaps have a very good thermal response. We will reproduce some MAXWELL BCA3000P test results taken by the US energy department[35].



Under 200A superCaps temperature is under 26 grades C. (image from [35])



SuperCaps can reach 35 grades C when charged repeately with 800A (image from [35])



Positive terminals tend to heat more. A good connection is needed (image from [35])

3.8. Power Distribution system

3.8.1. Previous works

3.8.1.1. Minne1 (Autumn 2009)

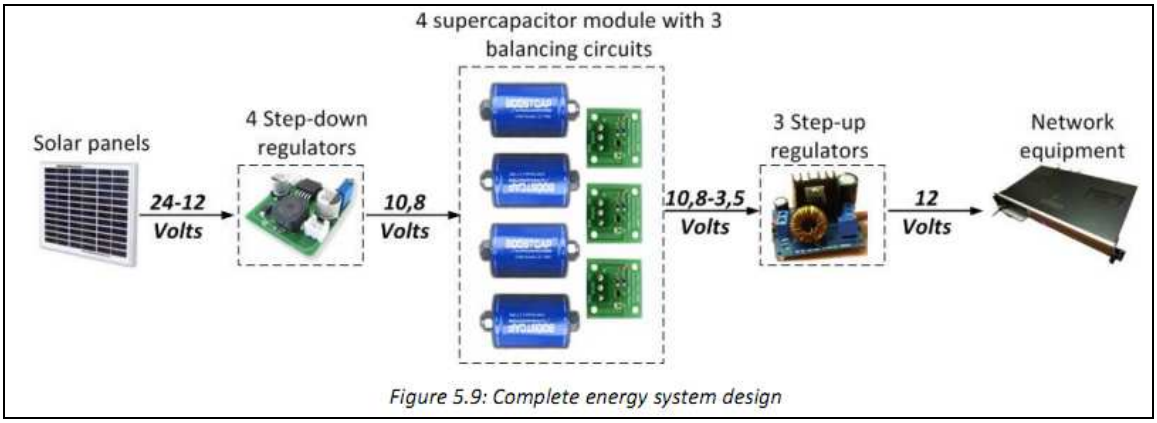


Figure 5.9: Complete energy system design

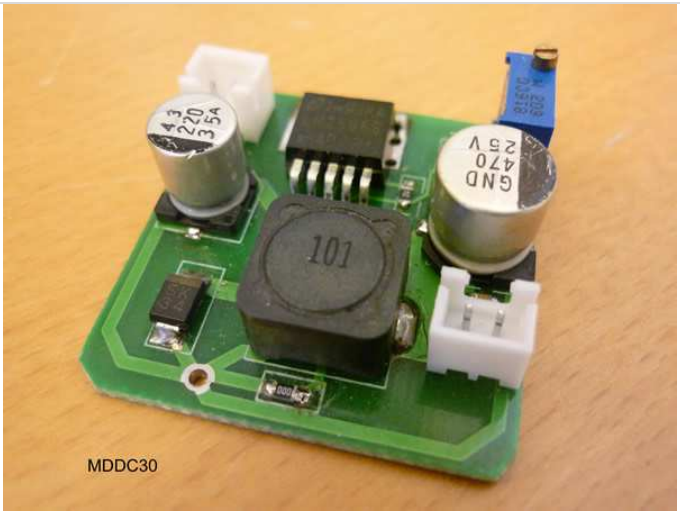
Minne1 team tested 1500F-2.7v super capacitors, charging them with different voltage power sources and the step-down.

Minne1 Considerations[31]

Supercapacitors can operate in voltages between their rated maximum value and zero, when they are completely discharged. For the supercapacitors used in Minne1 tests (Maxwell BCAP1500) the maximum voltage rating is 2.7V. This maximum voltage should not be exceeded.

As a result Minne1 needed step-down regulators (buck converters) that can convert the high voltage of the solar panels to the low voltage of the supercapacitors. Another important feature of the step-down circuit is that it regulates varying input voltages to a stable output value. [31]

Minne1 used a MDDC30 DC-DC voltage regulator to maintain the voltage in an adequate value.



Input Voltage: 3-30v

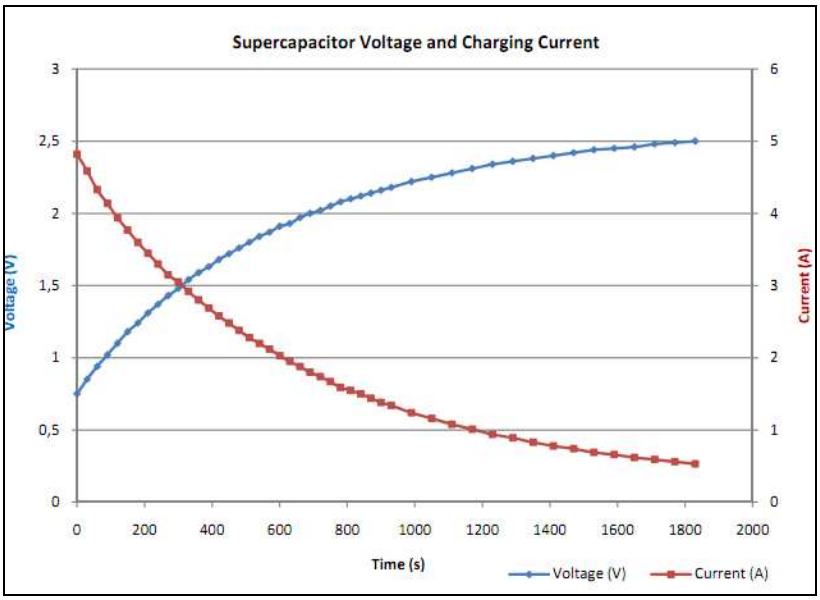
Output Voltage: 1.3-18v

Output current: 3A

Performed tests

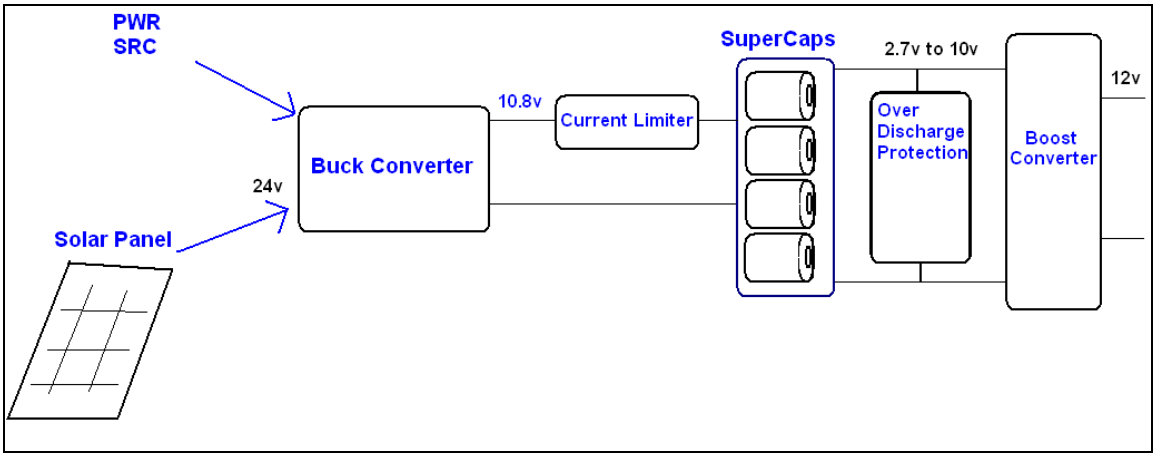
Minne1 team made some test, covered in [2]. The step down regulator takes the 12v from the solar panel and reduces it to 5v average, needed to charge a 2.7v supercapacitor, also regulating the solar panel variable voltage output.

Minne1 team found that they can combine regulators to generate 5v and up to 4.6A output each regulator when loaded with a 0.2 ohm resistor. In some test, the regulator collapsed and produced high heat. When the regulator is adjusted to 2.7v, they can only charge the superCaps to 2.5v, as seen in the graph below. As conclusion, the regulators could be used when combined, but they need a current limitation circuit to avoid damage. They claim that one superCap was destroyed when a 7 amp current was applied for few seconds. However, according to the manufacturer[], super capacitors can support very high currents without damage.



3.8.1.2. Minne2 (2010)

For the energy module, Minne2 team also used 4 superCaps, keeping the same module design as Minne1 but replacing the DC-DC regulator by a more powerful one:



Minne2 Energy system

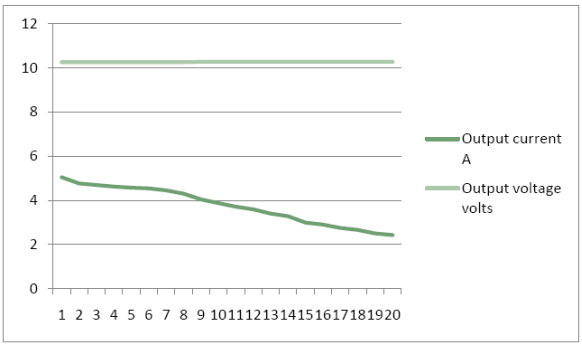
The buck converter is again used to maintain a proper charge voltage for the 4 super capacitors, from the power source or from the solar panel. Minne II used a LM315X voltage regulator evaluation board as the buck converter.

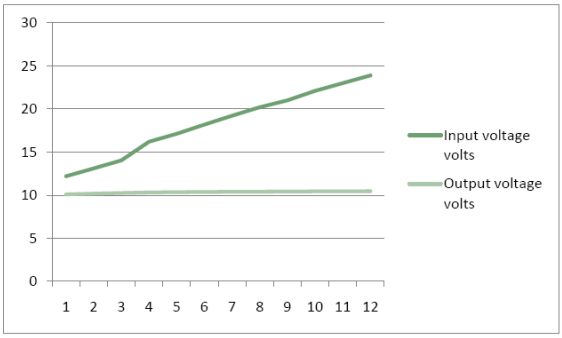
Input voltage: 6-24v

Output voltage:

Output current: 12A

LM315X was set to generate 10.8 volts. The tests results by Minne2 team are shown in the following images, where we can see that the board properly regulates the voltage to 10.8 v:





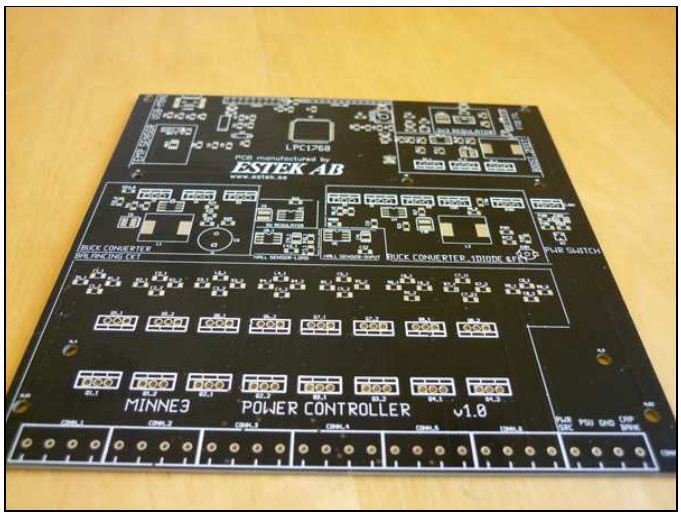
Minne2 tests results [33]:

This design can replace four step down modules in MinNE I by one step down module capable of delivering 12 A load current at 10.8 v output voltage, However Minne2 did not designed their own board because the evaluation board is commercially available.

3.8.1.3. Minne3 (autumn 2010)

Minne3 team made big changes to the energy module, they increased the super capacitor number to 16, included a microcontroller for the controlling part and designed a PCB containing the uController, step-up, step-down, hall sensor and balancing circuit for the capacitors. The final report is available in [1]

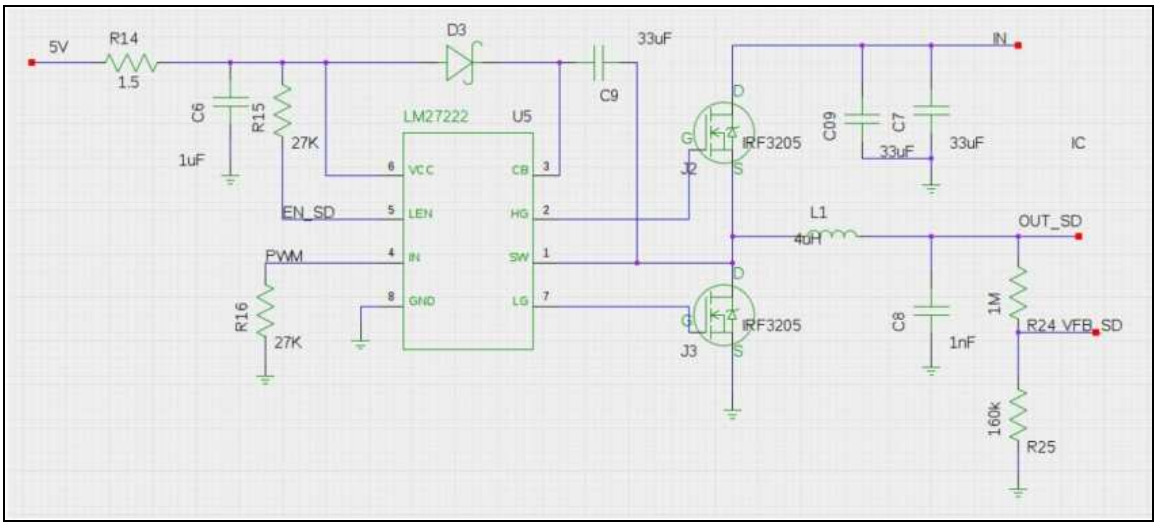
Minne3 team did not assembled the designed PCB but performed some tests in project boards. The PCB tests are done by Minne4 team.



Minne3 PCB

This DC-DC design was outsourced to the System on Chip team, the designed circuit senses the input voltage and automatically performs step-up or step-down. SoC final report is found in [7].

The SoC designed buck converter (step-down) is shown in the following image:



This circuit was tested with a 300ma load current, a PWM signal generator (not the uController), and gave the following results [7]:

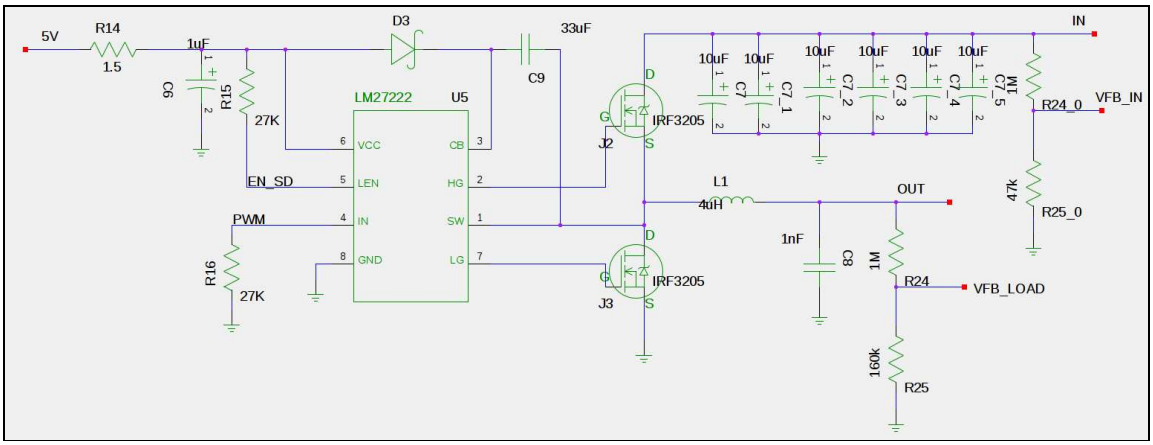
Buck converter with 20% of the duty cycle, power supply 6V, current limitation 1A:

Frequency(kHz)	10	20	30	40	50	60	70	80
Vin(V)	5.85	5.98	5.98	5.98	5.98	5.99	5.99	5.99
Vout(V)	0.90	1.00	1.05	1.10	1.10	1.10	1.10	1.10

Buck converter with 50% of the duty cycle, power supply 6V, current limitation 1A:

Frequency(kHz)	10	20	30	40	50	60	70	80
Vin(V)	2.07	4.98	5.97	5.98	5.98	5.98	5.98	5.98
Vout(V)	0.90	2.30	2.90	2.90	2.90	2.90	2.90	2.90

The final and printed circuit differs from the SoC proposal and is shown here, the main difference is that the printed one uses 6 10nF capacitors (C7) instead of 2 33nF capacitors:



Buck.sch

The Minne3 gEDA files with the schematics can be found in:
http://csd.xen.ssvl.kth.se/csdlive/sites/default/files/projects/Schematics%20Files.tar_.gz

file: buck.sch, boost.sch

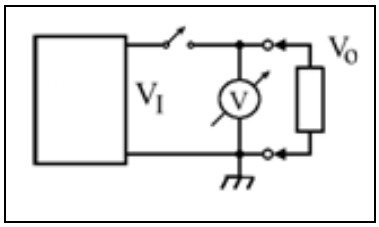
Minne3 Results:

The PCB was not tested, the circuit was only mounted in a project board and tested at 300 ma load and no super capacitors.

3.8.2. Minne4 Buck converter

The Buck Converter theory:

In summary, the working principle for switching regulators is based on manage the time instead of the voltage itself, an analog regulator applies a part of the available voltage during all the time, but a switching regulator applys all the voltage during a part of the time. If in instance, we have a 15v primary source for powering a 15v led. If we apply all 15v directly to the led, it will mighty light and it will maybe be destroyed, I we want this led to light little, with an analog solution, we would insert some resistor before the led and it will reduce the voltage given to the led, while this resistor absorbs and consume some power. With a switching circuit, we could use an interruptor or switch between the source and the load:



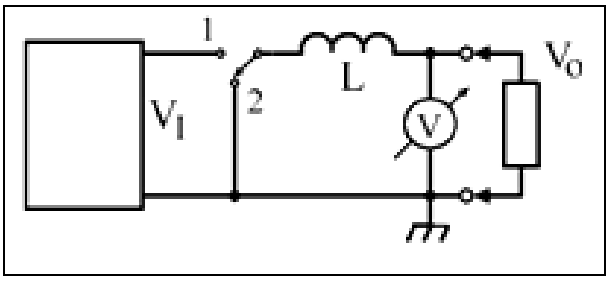
When the switch is closed, V0=Vi the led lits with all intensity, if the switch is opened, then the led goes off, if the switching is fast, then there will be a lite light because the led could not completely get off or on in each cycle. The advantaje of this solution is that the switch does not consume any power so teoreticaly, the efficiency will be η=100%.

If T1 is the time when the switch is on, then, the average voltage in the load is:

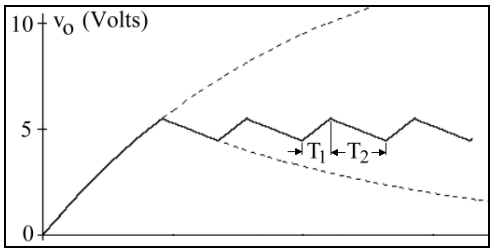
V=V1*T1/T

T1/T is also called duty cycle.

This circuit however, could not be used to feed other electronical circuits, because the resulting cycling voltage could damage it, in practical cases, Vo must be kept as a constant value, not just keep an average value.



If we use a 2 states switch and an inductor, we get a called forward converter, The main idea is that no matter the switch's state, there is always some current flowing to the load. In the position 1, the source charges the inductor and Vo gradually increases, in position 2, the inductor keeps giving current to the load while Vo decreases. In the following image, T1 and T2 are too short, so the inductor does not completely discharges when it receives charge again, obtaining a more constant voltage with some ripple voltage [9].



This ripple voltage is also undesired, and to decrease it, we could reduce T1 and T2, it means to increase the switching frequency, which has some limitations since high frequencies require special elements, it also produces heat and is limited to the available oscillation generators. The voltage can also get better by increasing the inductor value or by adding an output capacitor, as we will see later.

The buck converter in low frequencies

Having the Ripple current equation, and a low frequency, for instance 1Khz, and a 50% duty cycle and Vo=5v and L=10uH, we find that Ripple current is 250A!!! So we can see that low frequencies require high inductance values (bigger inductors). It is then desirable to increase the frequency and have low inductor values.

If we have, for example: F=100Khz, L=10uH, Vo=5v, D=50%, then we have a 2.5A ripple current. (Which is also too high). To have 250mA ripple current, we should then use a 100uH inductor, which is very available.

When the load varies

For the last example, with L=100uH and F=100Khz we found 2.5A ripple current, and with a 5 ohm load we will have Vripple=12.5v and with RL=50ohm we will have Vripple=125v!!, to fix the results we could hardly increase the inductor value or the 100Khz frequency. Fortunately, these handicaps can be solved with an output capacitor.

Draining resistor

As we saw before with all those drastic current changes, we noticed that there always should be a load connected to the inductor, to drain the current stored in the inductor when there is no load connected, so a resistor should be connected to the output.

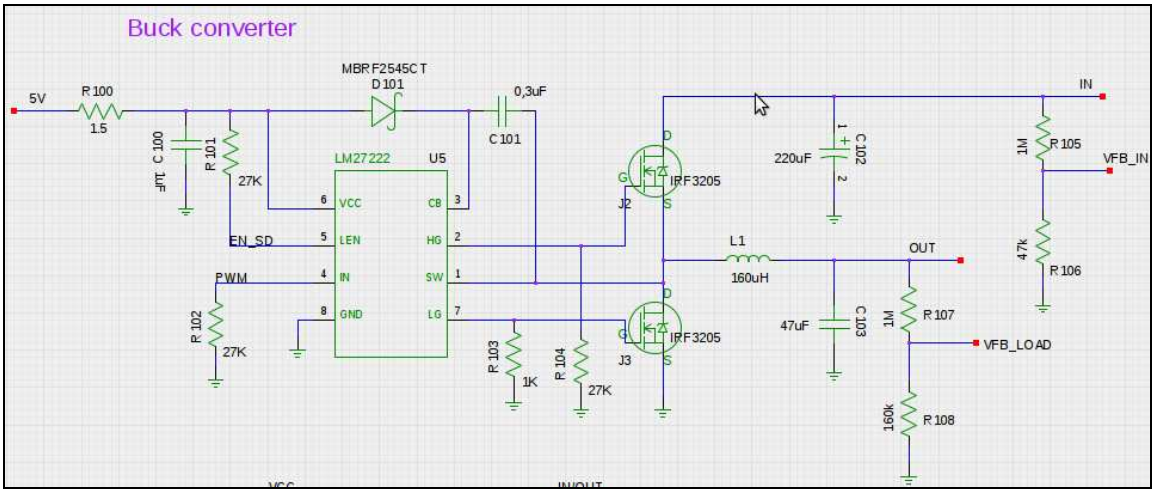
Output capacitor

As said before, it is posible to reduce the ripple voltaje in the output with a capacitor, however, in switching sources, this output capacitor becomes mandatory not just to filter the voltaje but to avoid peaks and to absorb the inductor current when the load current id low and the inductor

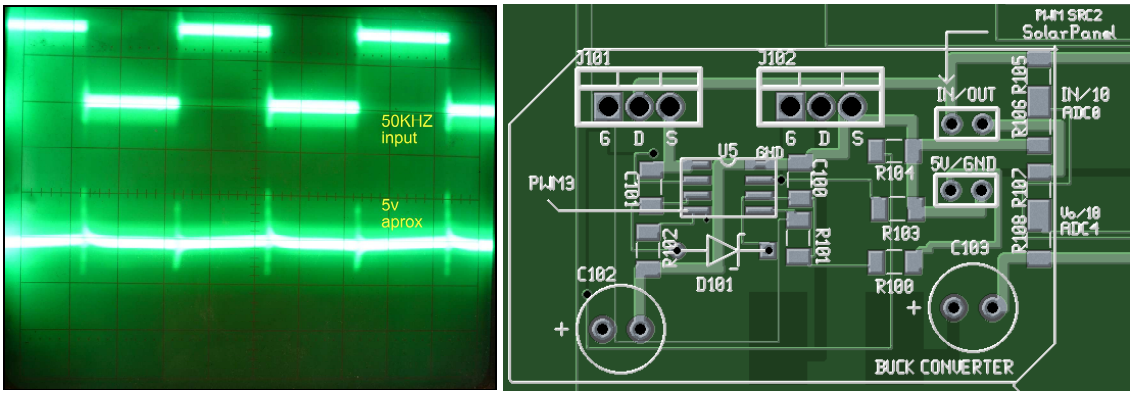
current should flow somewhere, it also stores some charge to help the inductor with the load when the main source is disconnected to the inductor.

3.8.2.1. Minne4 Buck converter proposal

After testing and documented Minne3 circuit and modified some element values, we implemented a new circuit design, following design formulas, good practices and recommendations from switching circuits design guides. The performed tests were done using the PWM output from the Blueboard and all its available command set. We found that higher frequencies produce better results but the circuit is noisier, we also found that higher inductances and capacitances produce better waveforms. The final circuit is shown in the following image:



Minne4 designed circuit



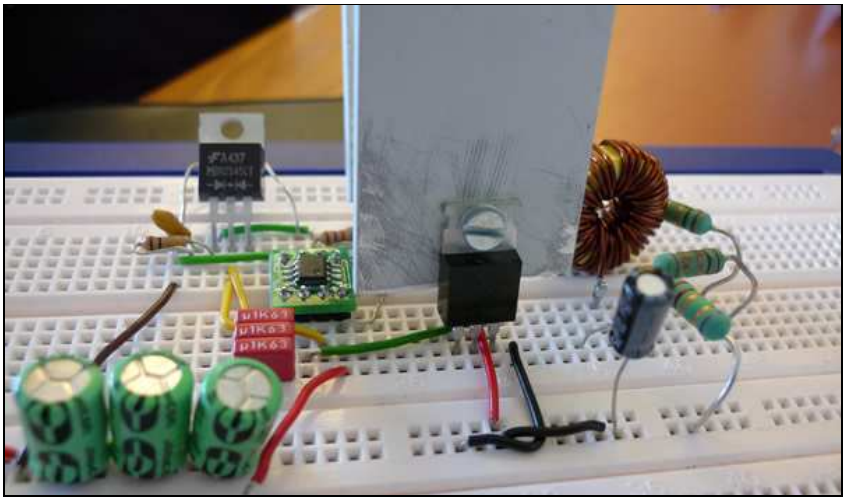
Output waveform and PCB design, F=50Khz

The tests were made in Minne3 PCB and a breadboard, both introducing much noise, we could produce 16W in the output when using a 11v input, Minne4 PCB has been designed following recommendations for high frequencies drivers and high power circuits, we expect that the results will be better when tested in Minne4 PCB.

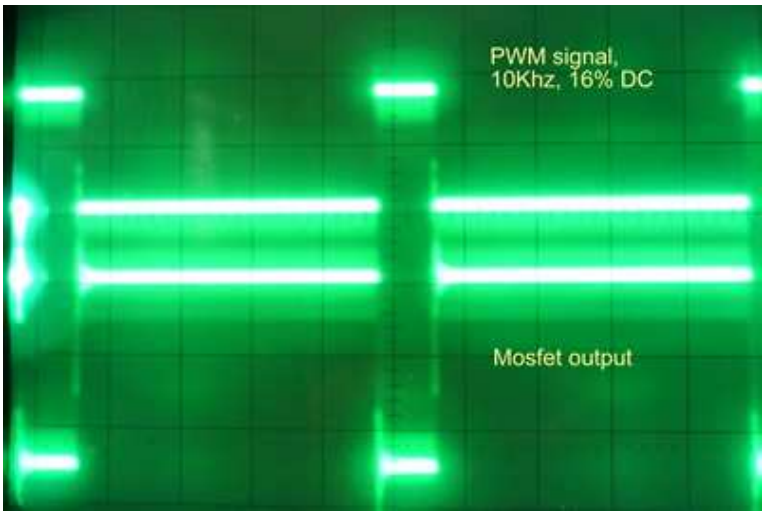
3.8.2.2. PWM signal Tests

The PWM signal is generated by the microcontroller in the Blueboard, the PWM5 pin is cabled to IN pin in the LM27222, and both grounds are linked too. The PWM frequency (1hz to 500Khz) and duty cycle (0% to 100%) can be set by commands via UART. For a better reference of the UART command line, see the Minne4 website.

```
MinN3> pwm hz 10000
Setting new Frecuency for all PWM...

cycles for new frecuency: 100
current(5) cycle% is: 50100  0.09
Current rate is:
PWM Frecuency is: Fclk/100   , F=10Khz
PWM Duty Cycle:
PWM1   PWM2   PWM3   PWM4   PWM5   PWM6
70%    10%    50%    30%    50%    10%

MinN3>
```

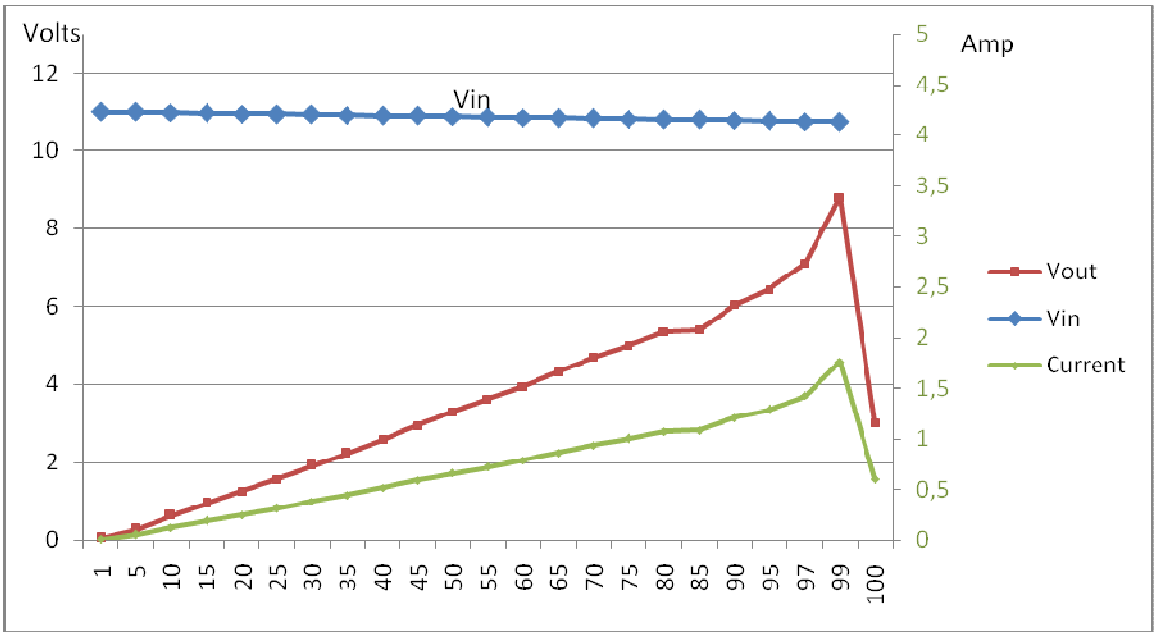


The Buck converter in the project board



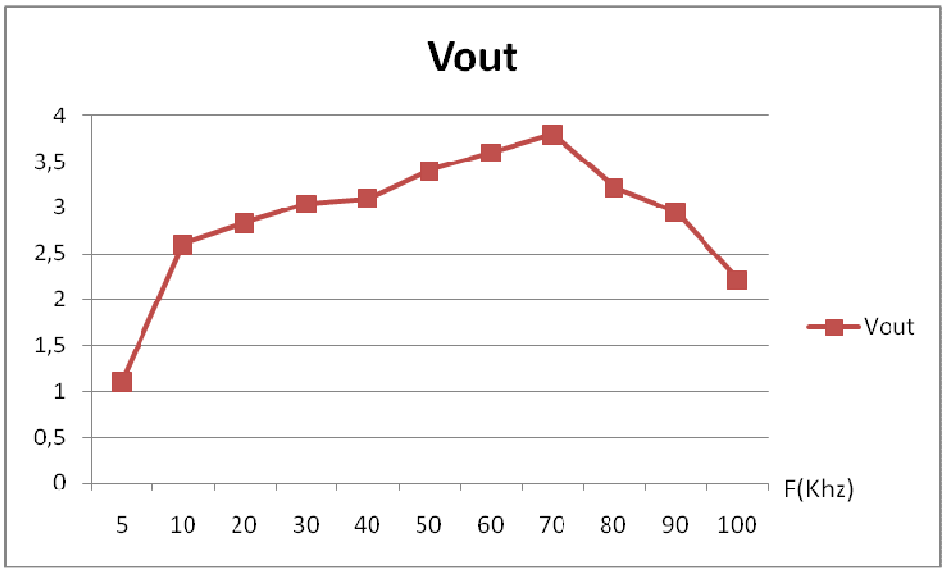
PWM signal from uController and the mosfet driver output

In Minne4 tests, C8 is replaced by a 33uF capacitor. The results are much better, shown in the following image:



Test results: L=4uH, C=33uF, Vin=11v, F=10Khz

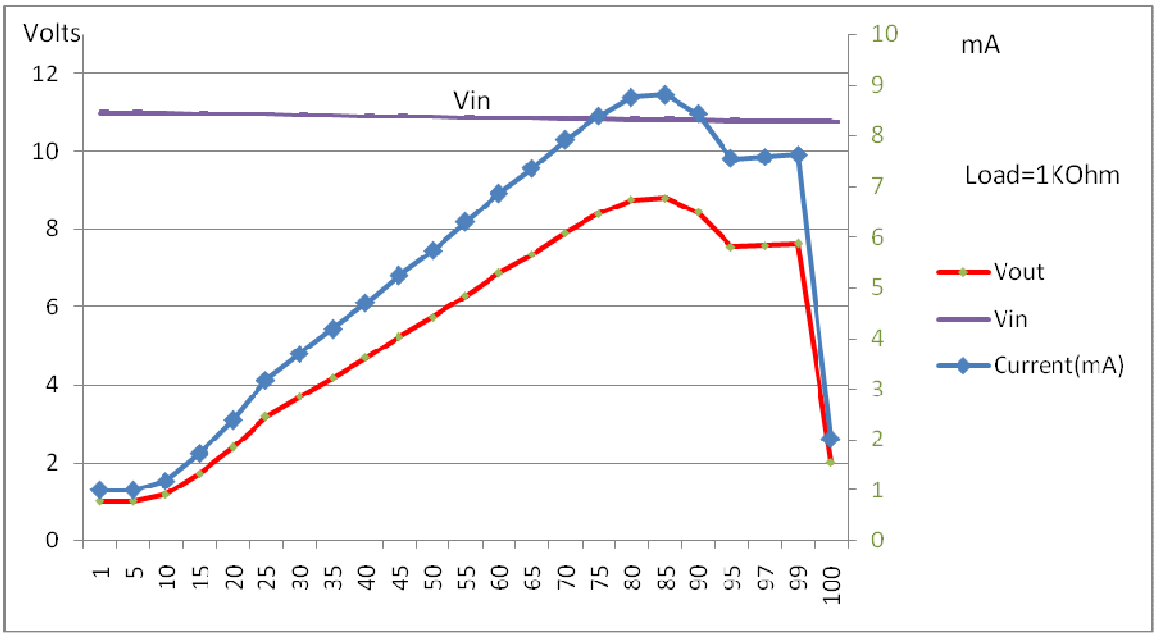
To find out what is the best work frequency for this circuit, we made a series of measurements varying the PWM frequency, resulting in the following graph:



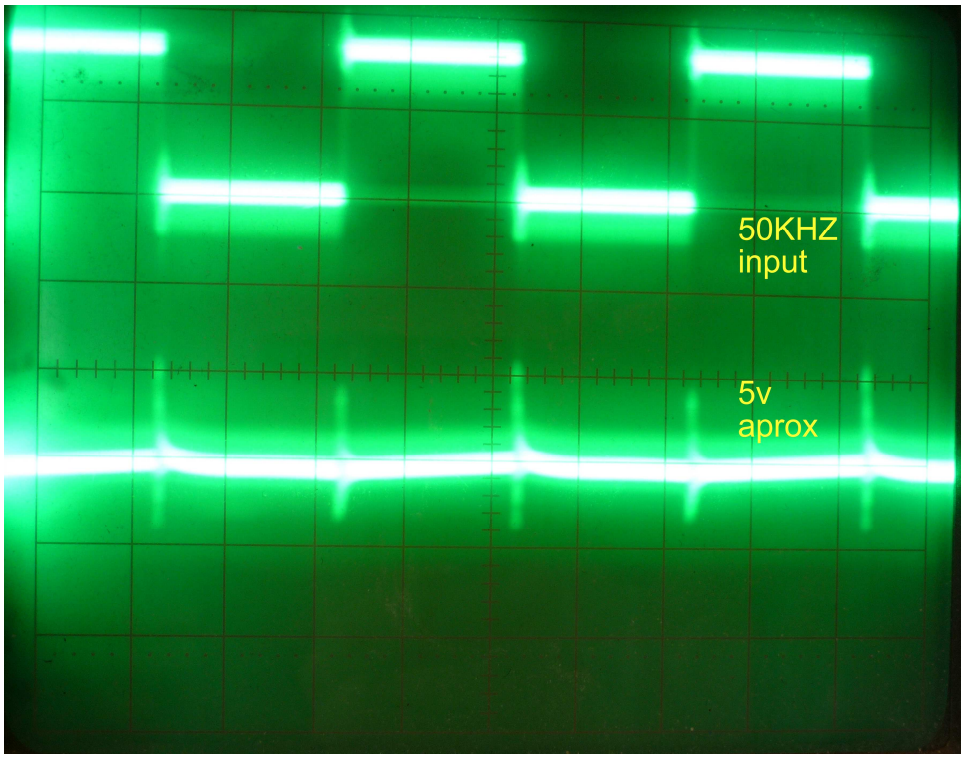
Test results: Vin=11v, Duty cycle=50%

We found that the circuit generates the maximum output voltage at about 70 Khz, however, that frequency is consider high for switching circuits and it produce heat in the mosfet driver and the mosfets themselves, also, such a high frequency produces noise and the resulting output is not so constant

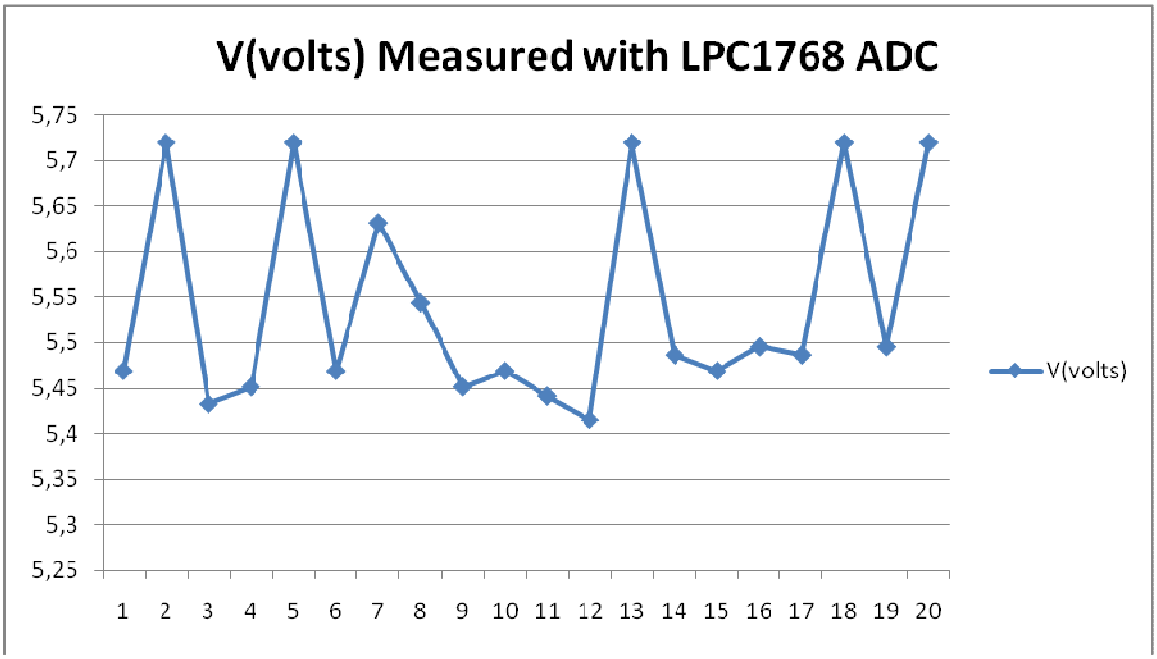
We used a 200uH inductor, we got the following results, having a 1K Ohm load:



F=50Khz, Rl=1K, Vin=11v



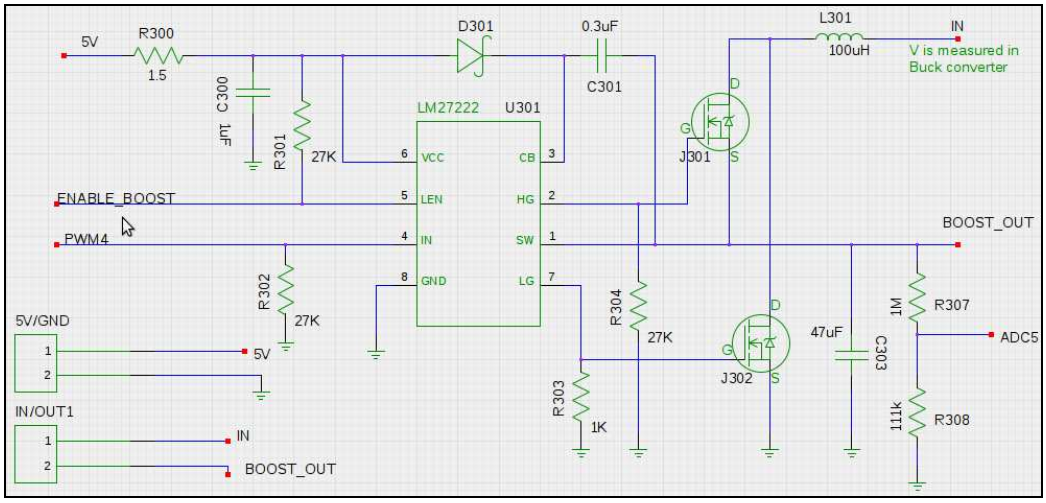
50Khz, 50% duty cycle, 200mH, Rl=1Kohm



Voltage measurements in the converter output, taken with the microcontroller ADC batch option

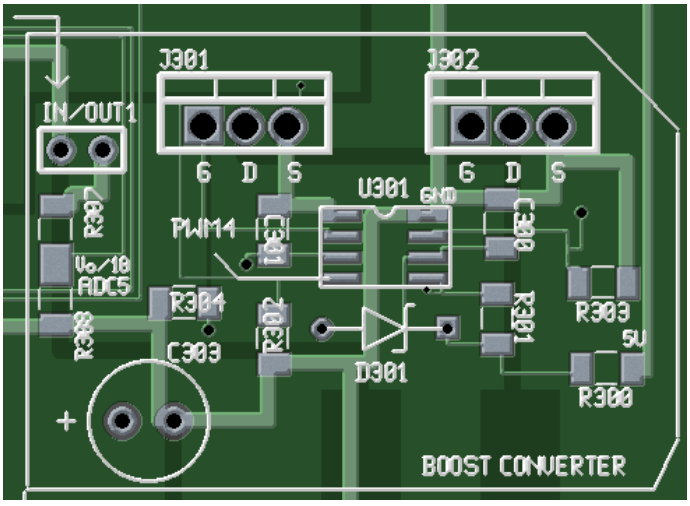
3.8.3. Boost Converter

Similar to the buck converter, the boost converter design was taken from Minne3, improving some elements values and adding the output capacitor, always following high power switching circuits designing guides, documented in the buck/boost converter deliverables. We mounted the circuit and made test, without good results, we consider that this tests could not be done in the bread board but in the printed PCB, so we expect that test in the Minne4 PCB go better.



Minne4 Boost converter design

The final boost converter circuit is shown in the following image:



Minne 4 Boost converter design (v2011-09-06)

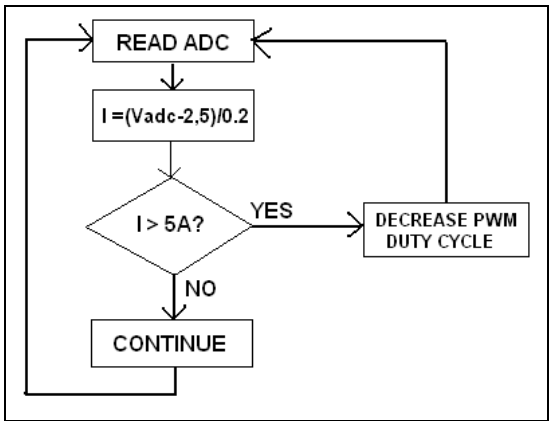
In the above PCB design, Minne4 tried to implement a good design, also including the ADC scaling, needed to scale to 0-3.3v the voltages for the ADC reader in the microcontroller, and we also include good labelling and an additional IN/OUT connector to make tests easier.

Integration with the microcontroller:

In the tests, as well as in the PCB, voltages were measured with the microcontroller's ADC, by using the "ADC [channel] batch [times] delay [delay ms]" we could take batch results and produce time graphs. The driver PWM was also produced and its duty cycle and frequency varied using UART user commands, such as "PWM [channel] setdc [DC in %]" or "PWM khz [new freq in KHZ]".

3.8.4. Current Limiter

Current must be limited to do not destroy the circuitry. We have calculated a maximum of 5A maximum current to keep the elements in a non dangerous temperature. Minne4 measures the charging current through the output Hall sensor, we tested the AC715 Hall sensor and included it into the Minne4 PCB, we documented it and wrote the needed code for the microcontroller to read the current through the ADC channel. The microcontroller controls the buck/boost circuits through the PWM, but the proper current limiter algorithm has not been programmed or tested yet, it should keep the circuit bellow a certain settable value.



Minne4 Current limiter algorithm

More information about the current measurement in Minne4 can be found in the document "Minne4 Hall sensor.pdf", from the deliverables list.

ACS712 5A Hall sensor

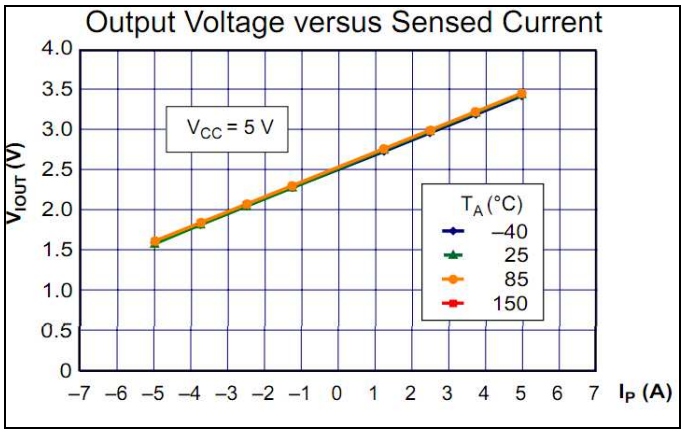
The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a

proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging. The output of the device has a positive slope (>VIOUT(Q)) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 mΩ typical, providing low power loss. The thickness of the copper conductor allows survival of the device at up to 5× over current conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques. Taken from [36].

Features[36]:

- 5 µs output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at TA = 25°C
- Small footprint, low-profile SOIC8 package
- 1.2 mΩ internal conductor resistance
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis

The ACS712 is provided in a small, surface mount SOIC8 package. The lead frame is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.



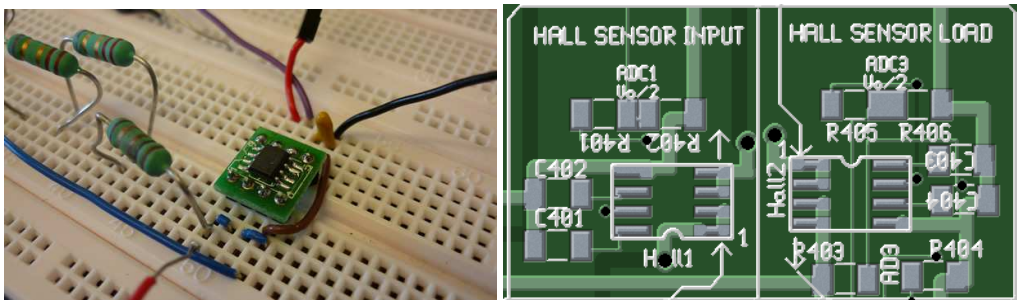
Main input Vs output graph.

The theoretical formulas are:

Vout=2,5+(0.2) Imeasured

Imeasured =(Vout-2,5)/0.2

The tests were performed in the bread board, and the final design was included in Minne4 PCB,



Hall sensor testing and PCB design, V2011-09-06

For the calculations, we took the following measurements and found that the approximately formula is:

Vin	Rin(Ohm)	Real I (calculated)	Vout(measured)
11	1000	0,011	2,54
11	500	0,022	2,54
10,5	4,8	2,1875	2,96
10,75	7	1,53571429	2,81
9,7	2	4,85	3,68

Imeasured=2,52+(0,21)Vout

We consider that we can use the Theoretical formula for the calculations in the micro controller.

The maximum expected Vout=3.5, for 5A. The ADC can measure up to 3.3v, so the ADC directly connected to Vout could register currents up to 4A (3.3v). We then need a voltage divider for measuring currents above 4A. If we use a 50% voltage divider, then we could measure 3.5v, or even 7v.

3.8.5. Over Discharge protection

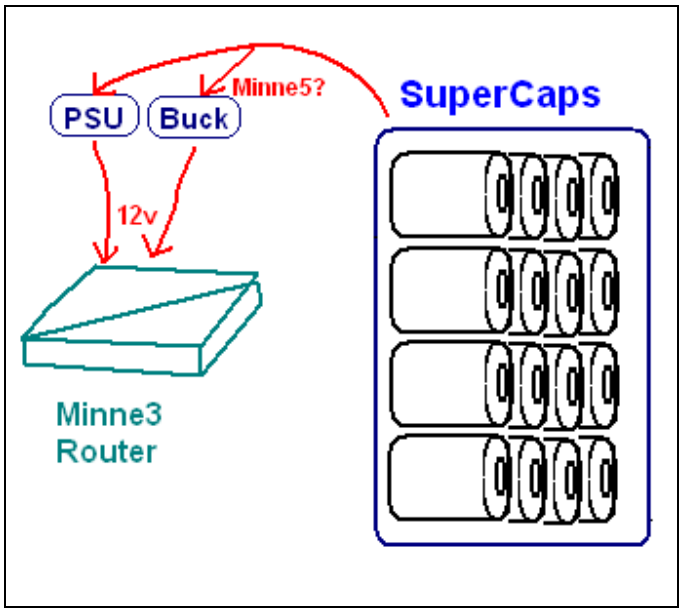
SuperCapacitors have a very low ESR (Eq. Series resistance), BCA3000P have 0.29 mΩ!! [17], when the charger is connected they draw a high charging current (6.3A according to [24]), similar to shortcut the circuit. This can cause the circuitry to burn or damage. The consumed charging current is decreased if the superCapacitors are not completely discharged when we start charging them. Minne2 proposed a discrete circuit composed of a comparator, it kept the capacitors bank above 4x0.65v=2.7V (Minne2 had 4x2.7v =10.8v). Minne3 developed scripts in Bifrost that query the microcontroller for data, if the superCaps voltage is too low (a settable value), Minne3 router will hibernate [25], keeping safe the routing data and releasing the load in the superCapacitors. In Minne4, we included in the designs and the PCB the needed circuitry to measure the voltage in the superCaps so the microcontroller can report reliable data to the router. Minne4 can read voltages and also change the duty cycle, a missing needed development is to program the code to join those modules and keep the superCaps with the proper voltage.

3.9. Routing module

The routing module is composed of the router Hardware,including the picoPSU, the serial communications system and the Bifrost OS [website in 21]. As said in the module description, Minne4 only proposes a picoPSU replacement by a buck converter and Bifrost O&M improvement.

3.9.1. PicoPSU

Inside the router itself, the PicoPSU regulates a wide different voltage range for the router, 3.3v, 5v, 12v, etc . If the mother board is replaced by one requiring 12v only, this function could be replaced by a buck converter in the superCaps output. This is proposed as future work, after the buck converter circuit is re-tested and approved.



If the Motherboard used a unique input voltage, a Buck converter could replace the PSU

3.10. Bifrost OS

In Minne4 project, we needed to install Bifrost, main website in [21], as a virtual machine and in a USB memory as well. We tested the serial communication to the microcontroller and we also tested the firmware flashing from Bifrost by deploying the software LPC21ISP, as mentioned in the "firmware flashing module". While using Bifrost, we realized that there is no installation guide or any starting guide, so we wrote the document "Minne4 deploying Bifrost into virtual machine or memory stick", meant to future developers or Bifrost users. Bifrost 7-0-4 is the latest distribution to date 2011-06-07, available from: ftp://ftp.sunet.se/pub/Linux/distributions/bifrost/. The Bifrost installation procedure can be summarized as:

Plug the USB to the computer, then find the USB name in the system...

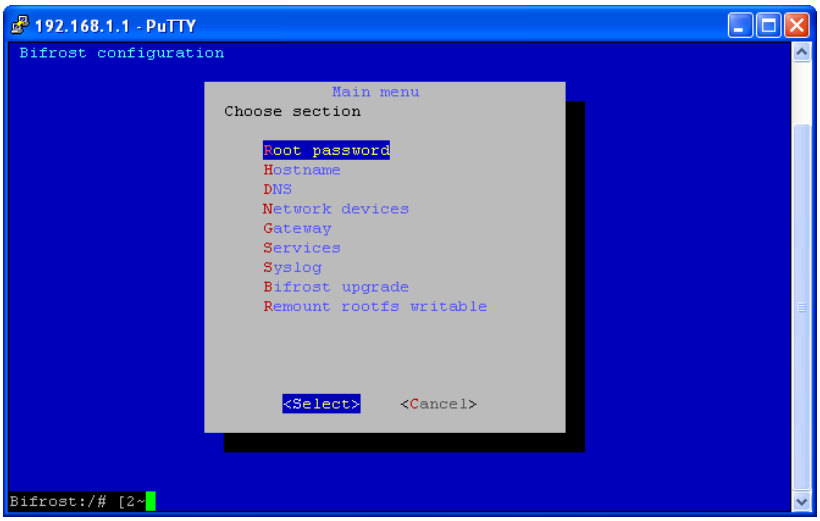
```
root@bt:~# fdisk -l |grep Disk
Disk /dev/sda: 1073 MB, 1073741824 bytes
Disk identifier: 0x00085bd3
Disk /dev/sdb: 21.5 GB, 21474836480 bytes
Disk identifier: 0x000e8eac
Disk /dev/sde: 1030 MB, 1030750208 bytes
Disk identifier: 0x00085bd3
```

Use that name to run the script from the directory containing the files.
*For bifrost7 use i.e. : **./make_bifrost sdc 0 boot_image.gz bifrost-beta-7.0-4.tar.gz**

```
#./make_bifrost sde 0 boot_image.gz bifrost-6.1-beta3.tgz
/dev/sde contains mounted file systems. Trying to umount them...
Trying to umount filesystem /dev/sde1

Will erase /dev/sde and load
/mnt/hgfs/KTH/MINNE3/bifrost/from_minne3_site/make_bifrost.tar__0_FILES/boot_image.gz onto it
Is this correct (yes is affirmative)?
yes
dd:ing the image to the device
2048+0 records in
2048+0 records out
8388608 bytes (8.4 MB) copied, 1.9707 s, 4.3 MB/s
Rebuilding partition table on /dev/sde...
Making filesystem with only one partition.
e2fsck 1.41.11 (14-Mar-2010)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
bifrost: 24/2000 files (0.0% non-contiguous), 479/8000 blocks
resize2fs 1.41.11 (14-Mar-2010)
Resizing the filesystem on /dev/sde1 to 1004028 (1k) blocks.
The filesystem on /dev/sde1 is now 1004028 blocks long.


Disk /dev/sde: 1030 MB, 1030750208 bytes
255 heads, 63 sectors/track, 125 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00085bd3

   Device Boot      Start         End      Blocks   Id  System
/dev/sde1   *           1         125     1004031   83  Linux
Mounting /dev/sde1 on /tmp/makeusbmem-mp
Adding optional package bifrost-6.1-beta3.tgz
Done - cleaning up.
```

3.10.1.1. Bifrost 7 used in Minne4 project

When launching Bifrost 7 for the first time, an optional configuration menu will appear. You can skip it or configure basic settings. The partition is read only by default, select “Remount rootfs writable” for making the partition writable.



A basic configuration includes:

- Change the root password
- Set an IP address and mask
- Set a Hostname

When setting the IP address, if you don't find your interface card, a restart maybe needed. After the configuration, a restart maybe needed.

Bifrost 7 useful commands

remount w	Remount the partition as writable
ifconfig	Shows the able interfaces and settings
ethtool [interface_name]	Shows the configuration and physical state for an interface. Example #ethtool eth0
`/etc/rd.c/rc.interfaces <interface> <up|down|show>`	Shows, brings up or down an interface
(file) /etc/config.data/nic.eth0	To manualy change parameters.
ip address	To add a secondary ip address. example ip address add 1.1.1.1/30 broadcast 255.255.255.255 dev eth6
netstat	Show the active tcp connections
ip route show	Shows the routing table

ps –ef |grep ssh	Shows if sshd process is up
more /etc/ssh/sshd_config	To see the ssh configuration
sshd	Enables ssh
/etc/init.d/sshd restart	Restarts SSH
cat /etc/ssh/sshd_config |grep ftp	Checks if sftp is enabled.
halt	If no arguments, then shutdowns.
reboot	reboots

4. Results

Minne4 team has continued the IT4D KTH and URJC initiative, delivered the following main achievements:

- provide guides and documentation

  We have developed a web site [10] containing the project description and all deliverables, also, we wrote guides and tutorials for every tested module.

- Mount, test, identify bugs and provide new designs for all the modules in the Minne3 PCB.

- Design a new Printed Board with the new circuitry, including the following features:

  - Simple design
  - Standard board for any other project
  - Easily flasheable thought ISP
  - Powered by USB or external source
  - Pins match the Blueboard, Blueboard works.
  - Leds showing the state, power ON, reset, test, etc.
  - Modularity. Modules work separately.
  - Double side Euroboard size 100x160mm
  - Well labeled and documented

- Give support for the incoming Minne5 support students.

5. Final conclusions

As Minne1 team shown, to solve the energy needing in Tanzania, or a similar network, to keep powered a load such the Bifrost router, it is possible to replace the typical acid battery by a super capacitors bank, the needed circuitry includes a charging circuit based on the solar panel energy, a voltage balancer and, included since Minne3, a monitoring and controlling system based on a LPC1768 microcontroller and a user interface through serial communication. Minne4 gave improvements, guides and new approaches to the design.

6. Future work

It has been found the following needed or suggested work for future project holders:

- To manufacture the designed Minne4 PCB, check the printed board, mount the elements and test the different modules in the PCB.
- To write the needed code, based on the developed ADC and PWM module, for choosing from buck or boost converter and decide the right charging voltage for the superCaps.
- Using the PCB's buck/boost converters, the hall sensors and the developed ADC and PWM modules, to write the code for the auto-adjustable buck and boost converter, the over discharge protection and the current limiter in the input and the output.
- To keep optimizing the microcontroller firmware, by removing unnecessary code and rewriting bad written code as necessary.
- After test and approve the PCB's buck converter, implement it as replacement for the router's picoPSU.
- To test the routers hibernation system designed, and tested by Minne3 on 2010.
- To keep tracking of the GNU flashing software LPC21ISP to get and test the latest versions.
- To include LPC21ISP as part of the Bifrost distribution.
- To develop new features in the serial communication Router-power system, such as an alarm system, a log system or a power O&M system.
- To develop a user friendly O&M interface for Bifrost
- To test the solar panel as power supply in Minne4 PCB.
- To design a buck/boost converter dual circuit.

7. References

[1] Minne1 final report
http://www.tslab.ssvl.kth.se/csd/projects/0921191/sites/default/files/Final%20Report.pdf

[2] Minne2 MDDC30 regulator testing
http://www.tslab.ssvl.kth.se/csd/projects/0921191/sites/default/files/stepdown%20regulator%20report_0_4.pdf

[3]MAXWELL charging of Ultracapacitors
http://www.maxwell.com/products/ultracapacitors/docs/AN-010_CHARGING_ULTRACAPS.PDF

[4] Minne2 final report
http://www.tslab.ssvl.kth.se/csd/projects/1031350/sites/default/files/Final%20report_MinNE%20II.pdf

[5] Minne2 Enegy module design report
http://www.tslab.ssvl.kth.se/csd/projects/1031350/sites/default/files/Energy_Module_Design_Report_v1.1.pdf

[6] Minne3 final report
http://csd.xen.ssvl.kth.se/csdlive/sites/default/files/projects/MinNE3FinalReportv1.0.pdf

[7] Minne3 SoC final report
http://csd.xen.ssvl.kth.se/csdlive/sites/default/files/projects/IL2213_Final_Report_v2.pdf

[8] LP2951 5v regulator datasheet http://www.national.com/ds/LP/LP2950.pdf

[9] Minne3 website http://csd.xen.ssvl.kth.se/csdlive/content/minne3

[10] Minne4 website http://web.ict.kth.se/~cjcm

[11] Farnell electronics providers, Sweden http://se.farnell.com

[12] OpenOCD-USB dongle website (German) http://www.embedded-projects.net/index.php?page_id=256

[13] NGX LPC1768_H website http://shop.ngxtechnologies.com/product_info.php?products_id=65

[14] OpenOCD website http://openocd.berlios.de/web/

[15] MBED website http://mbed.org/

[16] FTDI cable http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf

[17] Maxwell super capacitors datasheet
http://www.glyn.co.nz/downloads/documents/Maxwell/DATASHEET_K2_SERIES_1015370.pdf

[18] Mini-box picoPSU Manual http://resources.mini-box.com/online/PWR-PICOPSU-120/PWR-PICOPSU-120-manual-engl.pdf

[19] Minne3 gracefull shutdown system
http://csd.xen.ssvl.kth.se/csdlive/sites/default/files/projects/Graceful%20shutdown%20for%20Minne%201.0.pdf

[20] LPC21ISP website http://sourceforge.net/projects/lpc21isp/

[21] Main Bifrost website http://bifrost.slu.se/

[22] JTAG Wikipedia, http://en.wikipedia.org/wiki/Joint_Test_Action_Group

[23] Internal Software Program ISP

[24] Minne2 superCaps charging profile
http://www.tslab.ssvl.kth.se/csd/projects/0921191/sites/default/files/Charging%20the%201500F%20supercaps.pdf

[25] Graceful shutdown for Minne3 Router
http://csd.xen.ssvl.kth.se/csdlive/sites/default/files/projects/Graceful%20shutdown%20for%20Minne%201.0.pdf

[26] Master COMPAD, URJC Madrid-Spain http://www.tsc.urjc.es/Master/COMPAD/

[27] GDB GNU Project Debugger website http://www.gnu.org/software/gdb/

[28] UART definition http://whatis.techtarget.com/definition/0,,sid9_gci213237,00.html

[29] LPC1768 datasheet
http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf

[30] LPC1768 user manual
http://ics.nxp.com/support/documents/microcontrollers/pdf/user.manual.lpc17xx.pdf

[31] JTAG Wikipedia, http://en.wikipedia.org/wiki/Joint_Test_Action_Group

[32] Minne1 final report
http://www.tslab.ssvl.kth.se/csd/projects/0921191/sites/default/files/Final%20Report.pdf

[33] Minne2 Enegy module design report
http://www.tslab.ssvl.kth.se/csd/projects/1031350/sites/default/files/Energy_Module_Design_Report_v1.1.pdf

[34] MAXWELL super capacitors user guide
http://maxwell.com/products/ultracapacitors/docs/1014627_BOOSTCAP_PRODUCT_GUIDE.PDF

[35] US energy department Thermal analasys for SuperCaps
http://www.nrel.gov/vehiclesandfuels/energystorage/pdfs/43564.pdf

[36] ACS712 datasheet

http://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf

[37] BURUCA project site http://vm-199.xen.ssvl.kth.se/csdlive/content/buruca

[38] ICT project site http://www.tslab.ssvl.kth.se/csd/projects/0605/