MCU Controlled DC-DC Buck/Boost Converter for Supercapacitors

Jorge Querol Borrès
June 29, 2012

Master of Science Thesis
Stockholm, Sweden 2012
TRITA-ICT-EX-2012:108
Abstract

This work is focused on DC to DC conversion, what is a crucial function to enable the use of supercapacitors for energy storage. A theoretical study and comparison of methods, algorithms and techniques for software controlled DC-DC converters have been used to develop a system what can step up or down a DC variable voltage and transform it into a steady state voltage.

As a result a new control theory based on Bang-Bang control has been developed with an ARM LPC1768 processor. It was implemented to solve the commercial converters problems because they cannot work with supercapacitors due to their low internal resistance.

The outcome is a device what can provide a programmable voltage between 4.5 V and 25 V, hardware can support up to 6 A and it is able to control the operating current flowing through the converter. It can be used with the supercapacitors as shown in this work but it can also be used as a general platform for voltage and energy conversion.

Furthermore, the designed hardware has the potential to work with smart grids via Ethernet connector, solar panels with MPPT algorithms and, at last, manage energy between different kinds of DC voltage sources and devices.
Sammanfattning

Detta arbete är inriktat på DC till DC konvertering, vad är en viktig funktion för att möjliggöra användningen av superkondensatorer för lagring av energi. En teoretisk studie och jämförelse av metoder, algoritmer och tekniker för program styrs DC-DC omvandlare har använts för att utveckla ett system vad som kan stega upp eller ner en DC variabel spänning och omvandla det till ett stabilt tillstånd spänning.

Som ett resultat av en ny kontroll teori bygger på Bang-Bang kontroll har utvecklats med en ARM LPC1768 processor. Det genomfördes för att lösa de kommersiella omformare problemen eftersom de inte kan arbeta med superkondensatorer på grund av deras låga inre motstånd.


Dessutom har hårdvara möjlighet att arbeta med smarta nät via ethernet-uttag, solpaneler med MPPT algoritmer och äntligen, hantera energi mellan olika typer av DC spänningskällor och enheter.
Resum

Aquest document està centrat en la implementació d’un convertidor DC/DC, el qual és un element fonamental en l’ús de supercondensadors per a l’emmagatzenament d’energia. S’ha realitzat un estudi teòric, així com l’anàlisi de diferents mètodes, algorismes i tècniques per al disseny de convertidors DC/DC controlats per software amb la finalitat de desenvolupar un sistema capaç d’elevar o reduir un voltatge DC de valor variable i transformar-lo en una tensió constant.

Així mateix, s’ha desenvolupat una nova teoria de control basada en el control Bang-Bang sobre un processador ARM LPC1768. Fou especialment implementada per a resoldre els problemes dels convertidors comercials ja que aquests no poden treballar amb els supercondensadors degut a la seva baixa resistència interna.

El resultat és un dispositiu que pot proporcionar una tensió programable des de 4,5 V fins a 25 V, amb un hardware que suporta fins a 6 A i que és capaç de controlar el corrent que circula través del convertidor. Pot ser utilitzat amb els supercondensadors, com es mostra en aquest document, però també com una plataforma per la conversió de tensió i la gestió d’energia.

A més a més, el hardware dissenyat té el potencial per a funcionar amb xarxes de potencia intel·ligents mitjançant un connector Ethernet, panells solars controlats amb algorismes MPPT i, al capdavall, gestionar la transferència d’energia entre diferents tipus de fonts i dispositius que operen amb corrent continu.
Resumen

Este documento se centra en la implementación de un conversor DC/DC, el cual es un elemento clave en el uso de supercondensadores para el almacenamiento de energía. Se ha realizado un estudio teórico, así como el análisis de diferentes métodos, algoritmos y técnicas para el diseño de conversores DC/DC controlados por software con el objetivo de desarrollar un sistema capaz de elevar o reducir un voltaje DC de valor variable e transformarlo en una tensión constante.

Asimismo, se ha desarrollado una nueva teoría de control basada en el control Bang-Bang sobre un microprocesador ARM LPC1768. Fue especialmente implementada para resolver los problemas de los conversores comerciales ya que estos no pueden trabajar con los supercondensadores debido a su baja resistencia interna.

El resultado es un dispositivo que puede proporcionar una tensión programable desde 4,5 V hasta 25 V, con un hardware que soporta hasta 6 A y que es capaz de controlar la corriente que fluye a través del convertidor. Puede ser utilizado con los supercondensadores, como se muestra en este documento, pero también como una plataforma para la conversión de tensión y la gestión de energía.

Además, el hardware diseñado tiene el potencial para funcionar con redes de potencia inteligentes mediante un conector Ethernet, paneles solares controlados con algoritmos MPPT y, a fin de cuentas, gestionar la transferencia de energía entre diferentes tipos de fuentes y dispositivos que operan con corriente continua.
Keywords

List of keywords from IEEE taxonomy:

- Energy storage
- Supercapacitors
- DC-DC power converters
- Voltage control
- Current limiters
- Bang-bang control
- Energy management
- Smart grids
# Contents

Abstract I  
Sammanfattning III  
Resum V  
Resumen VII  
Keywords IX  
Contents XI  
List of Figures XIII  
List of Tables XV  

1 Introduction 1  
  1.1 Problem statement ................................................................. 1  
  1.2 Related work ........................................................................... 1  
  1.3 Outline .................................................................................... 5  

2 Bang-Bang control 7  
  2.1 Differences with Sliding Mode Control ........................................... 7  
  2.2 Considerations about the converter ............................................... 8  
  2.3 Bang-Bang control theory ............................................................ 8  
  2.4 Bang-Bang controller ................................................................. 11  

3 Hardware design 15  
  3.1 Converter core .......................................................................... 15  
  3.2 Control logic ............................................................................. 17  
  3.3 Microcontroller .......................................................................... 19  
  3.4 Overvoltage protection ............................................................... 22  
  3.5 Hardware implementation .......................................................... 23  

4 Control software design 25  
  4.1 Implemented features ................................................................. 25  
  4.2 Initialization ............................................................................... 25  
  4.3 Main loop .................................................................................. 28  
  4.4 Bang-Bang Algorithm ............................................................... 29  
  4.5 Switch management ................................................................. 30  
  4.6 Serial bus commands ............................................................... 33  

XI
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Prototype evaluation</td>
<td>35</td>
</tr>
<tr>
<td>5.1</td>
<td>General behavior</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Efficiency considerations</td>
<td>37</td>
</tr>
<tr>
<td>5.3</td>
<td>Ultracapacitors handling</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Future work</td>
<td>45</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>47</td>
</tr>
<tr>
<td>A</td>
<td>Comparative C pseudo-code between Bang-Bang and Sliding Mode Control approaches</td>
<td>51</td>
</tr>
<tr>
<td>B</td>
<td>Bang-Bang controller implemented with MATLAB Simulink blocks</td>
<td>53</td>
</tr>
<tr>
<td>C</td>
<td>Bang-Bang controller implemented in C pseudo-code</td>
<td>55</td>
</tr>
<tr>
<td>D</td>
<td>Schematics</td>
<td>57</td>
</tr>
<tr>
<td>D.1</td>
<td>Overvoltage circuit schematics</td>
<td>57</td>
</tr>
<tr>
<td>D.2</td>
<td>Main circuits schematics</td>
<td>58</td>
</tr>
<tr>
<td>D.3</td>
<td>Ethernet controller schematics</td>
<td>59</td>
</tr>
<tr>
<td>E</td>
<td>Source code</td>
<td>61</td>
</tr>
<tr>
<td>E.1</td>
<td>main.c</td>
<td>61</td>
</tr>
<tr>
<td>E.2</td>
<td>control.c</td>
<td>63</td>
</tr>
<tr>
<td>E.3</td>
<td>bang.c</td>
<td>66</td>
</tr>
<tr>
<td>E.4</td>
<td>adc.c</td>
<td>70</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 Universal programmable DC/DC converter .................................................. 2
1.2 MinNE teams operation system ................................................................. 3
1.3 MinNE teams designed converter ............................................................... 3
1.4 Bidirectional buck/boost DC/DC converter .................................................. 5
1.5 Bidirectional converter with constant output and variable input .................. 5

2.1 Bidirectional non-inverting buck/boost converter ...................................... 8
2.2 Regions of convergence for each kind of converter ..................................... 10
2.3 Evolution of Bang-Bang converter with variable input voltage .................... 11
2.4 Evolution of Bang-Bang converter efficiency with variable input voltage ...... 12
2.5 State plane variation of the converter ....................................................... 12
2.6 Buck operation mode in phase plane ......................................................... 13
2.7 Boost operation mode in phase plane ......................................................... 13

3.1 Bidirectional non-inverting buck/boost converter with MOSFET switches ...... 15
3.2 Conversion efficiency for single buck and boost converters ......................... 16
3.3 MOSFET gate drivers ................................................................................. 18
3.4 Voltage divisors for adapt ADC range ....................................................... 19
3.5 Bidirectional current sensor ....................................................................... 19
3.6 Power supply circuit ................................................................................... 20
3.7 Microcontroller main connections ............................................................... 20
3.8 Ethernet physical circuit .............................................................................. 21
3.9 Push buttons circuit ..................................................................................... 22
3.10 Overvoltage circuit to protect supercapacitors .......................................... 23
3.11 Top view of the designed PCB ................................................................. 24

4.1 Core clock structure ................................................................................... 26
4.2 PLL structure ............................................................................................. 26
4.3 Main loop controlled by ADC interruptions ................................................ 28
4.4 Bang-Bang control algorithm ....................................................................... 29
4.5 Shoot-through effect .................................................................................. 31
4.6 Shoot-through protection ............................................................................. 31
4.7 Reverse Current effect ................................................................................ 32
4.8 Reverse current and shoot-through protection .......................................... 32

5.1 Converter working in buck mode ............................................................... 36
5.2 Converter working in boost mode ............................................................... 37
5.3 Converter working under current limitation ................................................. 38
5.4 Performance of the converter under light load conditions ......................... 38
5.5 Efficiency of the converter with high load .................................................. 39
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6 Voltage-Current characteristic of the overvoltage circuit</td>
<td>40</td>
</tr>
<tr>
<td>5.7 Supercapacitors voltage limited by the overvoltage circuit</td>
<td>40</td>
</tr>
<tr>
<td>5.8 Efficiency of the converter working with the supercapacitors battery</td>
<td>41</td>
</tr>
<tr>
<td>B.1 Converter Bang-Bang controller</td>
<td>53</td>
</tr>
<tr>
<td>B.2 Operating state makers</td>
<td>53</td>
</tr>
<tr>
<td>B.3 Proposed sub-controller to decide between buck or boost converter</td>
<td>54</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Phase portraits for each combination of switches ........................................ 9

4.1 List of programmed features ..................................................................... 25
4.2 State of MOSFET transistors for each state of the converter ..................... 30
4.3 Serial commands to configure the converter ............................................. 33
Chapter 1

Introduction

1.1 Problem statement

Concern about the environment and rising of renewable energies set new scenarios to replace the old sources of energy. Therefore, it seems logical that new kinds of energy require new kinds of storage for them. Traditional lead-acid batteries are the main support to storage electrical energy but their characteristics have weaknesses, including a high internal resistance, limited lifetime depending on operating temperature and deep cycling. They also include polluting elements, such as lead, that are less desirable from an environmental point of view. Batteries based on supercapacitors can take the place of traditional batteries with chemical elements in the energy world because they meet the requirements to be compatible with renewable energies. Nevertheless, they have an important disadvantage; their output voltage is not constant and depends on the amount of energy stored.

This project has the aim of provide a solution to the main problem of the supercapacitors batteries and make them able to supply whatever device that works with DC (Direct Current) voltage. The designed system in this document is based on a DC/DC converter that allows using these supercapacitors to supply other devices as well as storing energy from DC sources (see Figure 1.1). The Universal programmable DC/DC converter must be able to transform a variable DC voltage into a programmable steady voltage without take care about its output load. This mechanism can solve the problem of the commercial converters.

They cannot work with supercapacitors because of their low internal resistance. For this reason, they break down and then they are useless if it is desired to limit the current going into the battery. This capacitor can take up to 60 A and, therefore, it has to be limited in order to preserve the converter. It requires a controller that works regardless of the load such a Bang-Bang controller.

Moreover, this converter is primarily conceived for working with renewable energies so it have to be able to work with solar panels or smart power grids as instance.

1.2 Related work

Supercapacitors battery

The supercapacitors battery is composed by a group of high value capacitors that can store a wide amount of energy. These supercapacitors have low internal resistance, long lifetime, high power performance and they are compatible with the environment.

The chosen capacitors are made by MAXWELL TECHNOLOGIES [1]. They are 3000 F and their maximum operating voltage is 2.7 V. The battery is composed by 16 supercapacitors,
8 groups of 2 in parallel connected in serial, so the equivalent capacitance of the battery is 750 F.

\[
C_{\text{battery}} = \frac{1}{C + C} + \frac{1}{C + C} + \frac{1}{C + C} + \frac{1}{C + C} + \frac{1}{C + C} + \frac{1}{C + C} + \frac{1}{C + C} = \frac{C}{4} = 750 \text{ F} \quad (1.1)
\]

The maximum voltage of the battery is 21.6 V and it can store 116160 J or 32.3 Wh, an amount about 7 % of a small car battery (BOSCH S3001 12V can store 1771200 J).

\[
E_{\text{battery}} = \frac{1}{2} C_{\text{battery}} (V_{\text{max}} - V_{\text{min}})^2 = 116160 \text{ J} = 32.3 \text{ Wh} \quad (1.2)
\]

**Power source for a low consumption router**

MinNE teams [2][3][4][5] were projects composed by KTH students and their work was dedicated to provide communication networks with equipment which have low energy consumption, support power failures and operate with renewable energy supply. They, in collaboration with SoC group [6], developed a power supply system with solar panel source and super capacitors to store the energy obtained.

After they, GreeNet was another student team that went on with MinNE project. They fixed bugs and checked the performance of previous designs. Their software, components, documents, design and equipment were a start point to the new converter.

**Hardware**

MinNE teams implemented a balancing circuit for the supercapacitors. It forces that all the supercapacitors have a similar voltage and then, to prevent the overvoltage on every one controlling the total voltage between the battery terminals. This balancing circuit is supplied by the same supercapacitors and it can be considered a component of the battery.

Otherwise, MinNE teams proposed a DC/DC converter with independent buck and boost single modules in order to manage the variable voltage of the supercapacitors battery and charge it with a solar panel. The block diagram in Figure 1.2 represents the power system for the MinNE router which is supplied with the supercapacitors battery due to PicoPSU.
The MinNE DC/DC converter with independent buck and boost modules have to regulate the amount of energy that comes to battery and step up or down the input voltage to get the desired value at its output. In doing so, the current going into battery and its maximum voltage charge can be controlled.

This converter is controlled with PWM (Pulse Width Modulation) signals generated by a microcontroller (ARM Cortex M3 LPC1768) [7] supplied by a 3.3 V regulator (LP2950-33) that can take its power from the input voltage or from the super capacitors battery. PWM signals are connected to drivers (LM27222) to provide the necessary current to make work the converter and are supplied by a 5 V regulator (LP2950-50). Moreover, current (ACS712) and voltage sensors make possible to monitor the converter variables by the microcontroller. In their design (see Figure 1.3), there are some problems:

- The microcontroller is able to change the duty cycle of the PWM signals to modify the obtained voltage but it cannot switch between buck or boost power inputs. It is possible to have reverse current in the converter that can cause a malfunctions or damages.
- The boost converter has a design problem with the current driver and it does not work (N-channel power transistor cannot drive current because the voltage in its bulk is higher than in its drain).
- The microcontroller can operate at 3.3 V but drivers and current sensors only from 5 V, so control logic cannot operate if power voltage is lower than 5 V.

GreeNet team could solve some of these problems and has done some test charging the supercapacitors. Their main improvements over the hardware were the followings:
• Correct and test the right performance of super capacitors balancing circuit.

• Correct the converter design and improve it with a switch to select the power input between buck and boost converter.

Software

MinNE teams developed the necessary software to control their converter with the microcontroller. They implemented the necessary libraries to modify the duty cycle of PWM peripheral, to get the values of current and voltage sensors and a serial module which it is possible to manage the configurable options. Moreover, they wrote some guides to install the required software for programming [8] and flashing the LPC1768 microcontroller [9].

In order to control the converter voltage, MinNE teams implemented a closed loop control algorithm based on monitoring the desired current and output voltage and then increment or decrement the duty cycle of PWM signal. It also checks if the supercapacitors battery has reached the maximum voltage and then turn off the converter. This closed loop algorithm can be improved with others that are faster, optimal, independents in relation to the load and simpler in the hardware.

GreeNet team developed other improvements in software and they did many tests to their converter and charging the supercapacitors. This work was entirely considered in the design of the new converter.

Bidirectional DC/DC converter for an EV application

Magnus Hedlund was a student in Uppsala Universitet and his Master Thesis [10] in Electrical Engineering is related to voltage conversion for electric vehicles. He designed a bidirectional buck/boost converter with a Sliding Mode Control to get the desired voltage at its output.

This converter (see Figure 1.4) is controlled by the microcontroller dsPIC30F2010, which is supplied by a 5 V regulator; current drivers (HAL50S) supplied by ±15 V regulators; and switch drivers (IR2110) that are independent each other and make possible to implement all possible configurations by the microcontroller.

In this work, it was designed a SMC (Sliding Mode Control) [11][12][13][14] based in a constant voltage and maximum current error switching surface. This method approach is recognized as an efficient tool to design robust controllers for complex high-order non linear dynamic plant operating under uncertain conditions and offers several advantages: stability even for large supply and load variations, robustness, good dynamic response and simple implementation.

However, as it is shown in [11] and [14], the direct output control used to make the closed loop control is not possible to implement following the Sliding Mode Control design rules. In this project, it was implemented a method based in Bang-Bang control [15][16] with voltage and current control. This control offers similar advantages as SMC but it requires a wide study of system dynamics based in phase portrait convergence and avoiding possible unstable states. It will be the base of the designed control algorithm to get the desired voltage at the output of the converter.

This converter is able to transform a variable input voltage into a steady output voltage controlling the maximum current like we want and its main behavior is shown in Figure 1.5.

It has some advantages compared with MinNE teams design:

• Buck and boost converter with the same components, it allows to have a reduced hardware and fast switching between each other.

• No PWM control signals implies faster and simpler software.
1.3 Outline

The structure of this document is the following. In first place, there have been stated the benefits and problems of the supercapacitors and how the designed converter might solve them. There have also been explained the previous related work what was used as a initial start point.

Later on, the control theory based on a Bang-Bang controller is designed and described. It will be necessary to implement a converter what can control voltage and current regardless con the output load.

In the next chapter, there are detailed explanations and calculations about how the converter has been designed to accomplish its function trying to maximize its efficiency. Some prototypes has been built to test if the hardware design, and also the Bang-Bang Control theory developed meets the specifications and forecasts.

Hereafter, the control software will be programmed following GreeNet guides and recommendations. It will implement the software necessary to make work the control algorithm over the designed hardware.

Eventually, they will be shown the results of a mounted prototype and what are the operating conditions of the designed hardware and software. After that, it will be stated the conclusions.

Figure 1.4: Bidirectional buck/boost DC/DC converter

Figure 1.5: Bidirectional converter with constant output and variable input
and some future work what may be implemented to improve the converter.
Chapter 2

Bang-Bang control

Bang-Bang controllers are VSC (Variable Structure Controllers) and they are specifically designed to control signals in discontinuous systems as in DC/DC converters. They are used in applications that need optimal control, solve in minimum-time problems and are also often implemented because of simplicity or convenience.

The advantages of Bang-Bang control are: efficiency, robustness in front of uncertain operating conditions, stability even for wide input voltage range, good dynamic response, minimal hardware implementation and independence with load variations [14][15][16].

Its correct performance regardless of load variations makes possible to work with supercapacitors battery. For example, a linear PID (Proportional Integral Derivative) control changes its performance when there is a change from resistive to complex load and produces a systematic error at the output [17], so it cannot work with the supercapacitors.

Bang-Bang control consists in switch on or off knowing the system behavior and then, to make it evolve in order to reach the desired stability point. The system behavior can be obtained by modeling it with a differential equations system for each combination of two switches and then analyze it calculating its phase portrait.

2.1 Differences with Sliding Mode Control

In control theory, Sliding Mode Control, or SMC, is a nonlinear control method, as well as Bang-Bang control, that alters the dynamics of a nonlinear system by application of a discontinuous control signal. It forces the system to "slide" along a cross-section of the system’s normal behavior called “switching surface”.

Summarizing, these are the main differences between proposed Bang-Bang control and Sliding Mode Control applied to DC/DC converters:

- Bang-Bang control does not control how current and voltage vary, it only checks for the final desired voltage.

- In contrast, SMC defines a “switching surface” where the dynamics have some convergence and a stability point.

- The constant voltage surface, defined in Bang-Bang, has not convergence but it has a stability point. For this reason, it cannot be implemented in SMC.

- First order SMC control is based in defining the output error and its derivative as the state variables. It provides a faster control than the Bang-Bang one but it is depending on the converter load.
It would be possible to implement the converter with SMC defining a limited range for the complex load. However, making it with a Bang-Bang control will become simpler. In Appendix A is shown this difference for a buck converter in C pseudo-code.

2.2 Considerations about the converter

There are some different configurations of DC-DC converters which can be used as buck and boost DC voltage transformers. However, there is one of them what provides the minimal hardware and most simple alternative without changing the polarity of the input voltage. Moreover, it can work in different ways and allows several configurations as well as they will be described below.

For these reasons, it has taken the DC/DC converter described in Section 1.2 as a starting point for the hardware and as a base to develop the Bang-Bang control theory.

It is a bidirectional non-inverting DC/DC buck/boost converter (see Figure 2.1) that sets up the hardware core and provides the main required features. This configuration has the ability to step up or down the input voltage by controlling the system switches. As the saying goes in [18], this kind of converter is composed by a buck cascaded by boost, and it can work as a single buck, a single boost or a buck-boost converter depending on the switches behavior.

![Figure 2.1: Bidirectional non-inverting buck/boost converter](image)

Thanks to its symmetrical design, it is possible to manage power bidirectionally using a set of sensors and switch drivers to control the energy flow in both directions as it was implemented in [10].

2.3 Bang-Bang control theory

As has been noted before, in Bang-Bang control is essential to know the system behavior and evolution in order to reach the desired stability point. It can be done through studying the phase portraits in CCM (Continuous Conduction Mode) for each combination of switches of the converter. CCM means that the current in the energy transfer inductor never goes to zero between switching cycles. From [10], the differential equations that model the non-inverting buck/boost converter are the stated below.

\[
\begin{align*}
\begin{pmatrix}
i_L' \\
v_{out}'
\end{pmatrix} &= \begin{pmatrix}
0 & -\frac{S_2}{L R_{load}} \\
\frac{S_2}{C} & -\frac{S_1 v_{in}}{L}
\end{pmatrix} \begin{pmatrix}
i_L \\
v_{out}
\end{pmatrix} + \begin{pmatrix}
\frac{S_1 v_{in}}{L} \\
0
\end{pmatrix} \\
S_n &= \begin{cases} 1 \text{ when position 1 (ON mode)} \\ 0 \text{ when position 2 (OFF mode)} \end{cases}
\end{align*}
\]

A phase portrait is a geometric representation of the trajectories of a dynamical system in the phase plane. In the converter case output voltage and inductor current are the variables what compose the phase plane. Also, differential equations what model the behavior of the converter are only valid under the assumption of CCM. The combinations of switch states and their corresponding phase portraits are summarized in Table 2.1.
Table 2.1: Phase portraits for each combination of switches
These normalized phase portraits represent the evolution of inductor current (x axis) and output voltage (y axis) in each one of converter states. States 1, 2 and 3 combined in couples define three different operation modes of converter. Buck single converter for 1 and 2, boost single converter for 2 and 3, and buck/boost converter for 1 and 3.

Moreover, for each operation mode of converter, there is a convergence region which includes the stable operating points. These regions are drawn in Figure 2.2 above the three first phase portraits. Blue region corresponds to buck single converter, green to boost and red to buck/boost.

These regions of convergence or stable operating points are placed where angle between vectors, corresponding to voltage and current derivative for each couple of combination of switches, are 180 degrees. From Figure 2.2, it follows that single buck and single boost converter are more efficient because it is possible to reach the desired voltage with less amount of current from the input.

The width of each region depends on the sample rate for the sensors and, inductor and capacitor values. Sample rate will determine the ability of the microcontroller to change the switch control signals in order to select the correct operation mode. Besides, both inductor and capacitor values will enforce the speed of the changes in inductor current and output voltage. All of them will be important parameters in the hardware design to get a correct performance of the converter.

Combination of switches number 4 is a state where inductor is discharging itself and losing energy and there is not any interchange of energy from the input power. For this reason, its phase portrait has not been represented in the convergence graphic. However, it can be an useful state in case of overload of the inductor, fast switching off of the converter or reducing the output voltage without charging the inductor.

From all this theory related to Bang-Bang control, it follows three important points to design the controller:

- In Bang-Bang control, boost converter cannot work without current limitation. State number 3 becomes unstable if converter does not change to another state. This fact does not appear in lineal controllers with PWM signals.
• A priori, current limitation only can be implemented by turning off the buck switch cutting off the input current (state 1). Studying the dynamics, we can assure that if the output voltage is higher than the input one, the current can be controlled by switching to state 2 (buck switch turned on) because the derivative of the current in this region is always negative. It increments the conversion efficiency respects doing the same with state 1.

• It is possible to rectify the lost of efficiency in buck converter (it cannot achieve values near input voltage) changing the operating mode to the boost one because they have in common the state 2 (buck converter always turned on is the same that boost converter always turned off). So, its performance is continuous between the maximum voltage reached by the buck converter and the minimum reached by the boost one. It will be necessary a sub-controller to get the optimal voltage for changing between buck and boost single converters.

2.4 Bang-Bang controller

A preliminary Bang-Bang controller has been designed and simulated. It switches between the four states depending on six parameters: output voltage, inductor current, input voltage, voltage reference, maximum voltage and maximum current.

The universal programmable buck/boost converter with Bang-Bang control has been simulated in MATLAB Simulink. The controller used in these simulations was implemented with Simulink functions (see Appendix B) and C pseudo-code has been programmed to carry out the same function (see Appendix C).

In this simulation, the input voltage varies from 21.6 V to 4 V so the converter has to work in both modes (single buck and single boost) to get the programmed output voltage value. In the Figure 2.3 is shown how voltage and current of the converter evolve with a voltage reference of 12 V and operating current of 5 A (in average). Furthermore, it has been simulated with a complex load of 10 Ω and 0.5 F.

![Figure 2.3: Evolution of Bang-Bang converter with variable input voltage](image)

It also has been represented in Figure 2.4 the time averaged efficiency and input and output power. In the graph, first 1.5 seconds correspond to the time charging the load capacitor and,
at 2.5 there is change in the operation mode from single buck to single boost and at 7.5 seconds the same change but in the opposite order.

Figure 2.4: Evolution of Bang-Bang converter efficiency with variable input voltage

It also interesting to see how the output voltage varies in front of the inductor current. Figure 2.5 shows how the converter evolves in state variable plane.

Figure 2.5: State plane variation of the converter

The performance of this Bang-Bang controller can be summarized in the following points:

1. If output voltage is higher than maximum voltage reference, set state 4 to reduce both inductor current and output voltage (see section 2.3).

2. If inductor current is higher than maximum current reference, set state 1 or 2 (depending on the output and input relationship) to reduce current passing through the converter (see section 2.3).
3. Decide which converter (single buck or boost) must be used according to the relationship between input voltage, output voltage and output reference voltage.

4. Decide which mode has to be used to move up or down the output voltage according to the relationship between output and reference voltage.

Once they are stated the main points to design the Bang-Bang controller, its performance can be represented over the phase portrait. In Figure 2.6 and Figure 2.7, it has been represented the combination of switches which has to be active depending on the instant values of output voltage and inductor current depending on the operating mode, buck or boost, which has been selected.

Figure 2.6: Buck operation mode in phase plane

Figure 2.7: Boost operation mode in phase plane
It can be appreciated that the trajectories (arrows) points to the desired output voltage ($v_{\text{ref}}$) and they are set up to reach the corresponding stability points. If $v_{\text{out}}$ is higher than $v_{\text{max}}$, combination of switches number 4 is active and then both $v_{\text{out}}$ and $i_L$ are reducing their values. Moreover, for a $v_{\text{out}}$ below $v_{\text{ref}}$, the derivative of the $i_L$ is positive if its value is lower than $i_{\text{max}}$, and it is negative on the contrary. This allows to control the maximum current flowing through the converter.
Chapter 3

Hardware design

The hardware of the bidirectional DC/DC buck/boost converter has been designed to accomplish some requirements in order to become an universal platform to develop software or architectures that incorporate elements related to green networks, microgrids, supercapacitors batteries and solar panels.

The design has been made following a process based in match the general theory of converters and a set of experimental tests through several prototypes. By this way it has been possible to check that this converter will be able to work with the Bang-Bang controller and provide a circuit capable to work with the supercapacitors battery. However, it can be also used by researchers or teachers as a generic powerful tool to develop any application related to DC power transfer, smart grids or green energy.

This converter has the aim of being able to transform an input DC voltage in a range of 5 V to 25 V, to an output DC voltage in a range of 0 V to 25 V, with a maximum input and output current of 6 A.

3.1 Converter core

A serial combination of a simple buck and a simple boost converter becomes in a step up and step down converter that can be used bidirectionally depending on the state of the switches [18]. It allows to the current going in to the converter to reach any voltage in the capacitance of one of the ends of the converter. This is the structure showed in the Figure 3.1 that is called non-inverting buck and boost converter and it is the simplest way to get a circuit that is able to step up or down the DC input voltage without change its polarity at the output.

![Bidirectional non-inverting buck/boost converter with MOSFET switches](image)

First important issue to consider is that all the components must have a low serial resistance because it decreases the conversion efficiency [19][20][18]. Figure 3.2, it is represented the relationship between output and input voltage depending on the duty cycle. It can be appreciate that the parasitic serial resistance will limit the output-input voltage ratio conversion and then it will reduce the converter capability to reach the desired output voltage.
Conversion Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Buck Mode</th>
<th>Boost Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ideal inductor</strong></td>
<td><img src="image" alt="Ideal inductor (R_L=0)" /></td>
<td><img src="image" alt="Boost Mode" /></td>
</tr>
<tr>
<td><strong>Non-ideal inductor (R_L=100)</strong></td>
<td><img src="image" alt="Non-ideal inductor (R_L=100)" /></td>
<td></td>
</tr>
<tr>
<td><strong>Non-ideal inductor (R_L=10)</strong></td>
<td><img src="image" alt="Non-ideal inductor (R_L=10)" /></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.2: Conversion efficiency for single buck and boost converters

**Inductor**

Relationships between values of the main components of the converter are well known and there are widely developed in power electronics literature. As it can be noted in [18], [19] and [20], the DC to DC converters can work in two modes. These are CCM (Continuous Conduction Mode) and DCM (Discontinuous Conduction Mode), and, in the case of this converter, the CCM gives better benefits such as higher efficiency, lower output impedance and less ripple at the output. Besides, the Bang-Bang controller has been designed to operate in CCM.

Therefore, the boundary between these modes establishes the lower value for the inductor that allows to work in CCM and is given by the following relationships depending on the mode that is working.

- **Buck mode:**

  \[
  L \geq \frac{(1 - D) R_{\text{load}}}{2 f_{\text{switch}}}  
  \]

  with \( 0 \leq D \leq 1 \) and \( \frac{R_{\text{load}}}{2 f_{\text{switch}}} \geq R_{\text{load}} \).

- **Boost mode:**

  \[
  L \geq \frac{(1 - D)^2 D R_{\text{load}}}{2 f_{\text{switch}}}  
  \]

  with \( 0 \leq D \leq 1 \) and \( \frac{4 R_{\text{load}}}{27 f_{\text{switch}}^2} \geq R_{\text{load}} \).

- **Buck-Boost mode:**

  \[
  L \geq \frac{(1 - D)^2 R_{\text{load}}}{2 f_{\text{switch}}}  
  \]

  with \( 0 \leq D \leq 1 \) and \( \frac{R_{\text{load}}}{2 f_{\text{switch}}} \geq R_{\text{load}} \).

For this design, it was assumed that, in steady state and in the worst case, \( R_{\text{load}} \leq 10 \, \Omega \) and \( f_{\text{switch}} \geq 50 \, \text{kHz} \) (Section 3.3). So, the value for the inductor must be \( L \geq 100 \, \mu\text{H} \). It has been taken the minimum value because it reduces size, cost and internal DCR (Direct Current Resistance).

The chosen inductor was the 2312-RC from BOURNS JW MILLER with \( L = 100 \, \mu\text{H} \), \( I_{\text{DC max}} = 7 \, \text{A} \) and \( \text{DCR}_{\text{max}} = 37 \, \text{m}\Omega \).

**Capacitors**

A pair of capacitors has been put in parallel in each ends of the converter to filter its voltage in order to get a DC voltage at the output. They are in parallel to reduce the ESR (Equivalent
and then increase its efficiency. Their capacitance is related to the maximum voltage ripple at the output and it is defined by the following expressions depending on the working mode of the converter.

- **Buck mode:**
  \[
  C_{\text{parallel}} \geq \frac{(1 - D)}{8L_f^2} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)^{0 \leq D \leq 1} \geq \frac{1}{8L_f^2} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)
  \]  
  \[ (3.4) \]

- **Boost mode:**
  \[
  C_{\text{parallel}} \geq \frac{D}{R_f} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)^{0 \leq D \leq 1} \geq \frac{1}{R_f} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)
  \]  
  \[ (3.5) \]

- **Buck-Boost mode:**
  \[
  C_{\text{parallel}} \geq \frac{D}{R_f} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)^{0 \leq D \leq 1} \geq \frac{1}{R_f} \left( \frac{V_{\text{out}}}{V_{\text{ripple}}} \right)
  \]  
  \[ (3.6) \]

Taking the same parameters than in the inductor case and considering that \(V_{\text{ripple}}/V_{\text{out}} \leq 1\) %, it follows that \(C_{\text{parallel}} \geq 50 \mu F\) from (4) and \(C_{\text{parallel}} \geq 200 \mu F\) from (5) and (6). It has been taken \(C_{\text{parallel}} = 2000 \mu F\) for each end of the converter because then it will become more robust in front of noise, input voltage variations and DCM operation.

Chosen capacitors were the 108CKS050M from ILLINOIS CAPACITOR with \(C = 1000 \mu F\), \(V_{\text{DC max}} = 50 V\) and \(ESR_{\text{max}} = 166 m\Omega\).

**Switches**

Converter switches have been made with MOSFET transistors because of their low on resistance and their robustness driving high power. A pair of P-channel and N-channel MOSFET provides synchronous switching in order to reduce the losses compared with the use of a diode instead and allows the converter to work bidirectionally.

In synchronous converters is usual to find a double N-channel MOSFET because they provide a lower on resistance. However, in this converter has been used a pair of P-channel and N-channel MOSFET instead because the P-channel MOSFET is able to drive current permanently between the inductor and the outer nodes and this fact is required to make it work with the Bang-Bang controller and to provide a bidirectional behavior. The non-ideal behavior of the MOSFET as switches as well as the internal resistance of the inductor and the capacitors have influence over the efficiency of the converter.

Chosen MOSFET transistors were the IRF4905 and IRF3205 from INTERNATIONAL RECTIFIER. IRF4905 is a P-channel MOSFET with \(V_{GS(th)} = 4 V\), \(I_{DS max} = -74 A\) and \(R_{DS(on)} = 20 m\Omega\), and IRF3205 is a N-channel MOSFET with \(V_{GS(th)} = 4 V\), \(I_{DS max} = 110 A\) and \(R_{DS(on)} = 6.5 m\Omega\).

### 3.2 Control logic

**MOSFET drivers**

As it is stated in [21] and [22], Power MOSFET have great potential as switches for high speed high voltage applications like pulsed power. Although the MOSFET is intrinsically capable of fast switching, the packaging inductance and the internal capacitances largely degrade the
performance of the device. These parasitic impedances become in a negative effect on the efficiency of the switch and hence it demotes the efficiency of the converter. To fight against this fact, there are MOSFET drivers that allow pumping high current in the MOSFET gate and then forcing a quick transition between switch states such as FAN3278.

The FAN3278 [23] is a dual 1.5 A gate driver optimized to drive a high-side P-channel MOSFET and a low-side N-channel MOSFET in switching control applications operating from a voltage rail up to 27V. Internal circuitry limits the voltage applied to the gates of the external MOSFETs to 13V maximum. This component incorporates MOSFET devices for the final output stage, providing high current throughout the MOSFET turn-on / turn-off transition to minimize switching losses.

In Figure 3.3 is shown how the FAN3278 drivers have been connected. Moreover, a 5.1 V zener diode has been connected to each one of its inputs in order to protect the microcontroller against any overvoltage.

Sensors

Sensors are needed to monitor several variables of the converter in order to apply the appropriate response through the MOSFET drivers. In Bang-Bang control, three variables must be monitored: input voltage, output voltage and inductor current (see Section 2.3).

Values of input and output voltage are obtained through a voltage divider and then it is read with an ADC (Analog to Digital Converter) of the microcontroller. The relationship between the input and the output of the voltage divider is the following:

\[ V_{ADC} = \frac{12}{100 + 12} V = \frac{3}{28} V \]  

Therefore, the microcontroller can read a voltage value in a range from 0 V to 30.8.

Moreover, a 1 nF capacitor has been added to filter noise and it provides a cut-off frequency of:

\[ f_{filter} = \frac{1}{2\pi RC} \approx 72 kHz \]  

Furthermore, a 5.1 V was connected to each divider zener diode whereby the microcontroller is protected against overvoltages. Figure 3.4 shows the circuit for each voltage that has to be measured.

A bidirectional sensor is required to get the value of the inductor current as well as it is necessary that it has a lower serial resistance in order to maximize the converter efficiency. The Allegro ACS710 [24] current sensor provides economical and precise means for current sensing and it fits perfectly to the converter requirements.
ACS710 is a precision linear Hall sensor integrated circuit with a copper conduction path located near the surface of the silicon die. Applied current flows through the copper conduction path, and the analog output voltage from the Hall sensor linearly tracks the magnetic field generated by the applied current. This output voltage proportional to the current and a zero current reference are read with two ADC of the microcontroller and then it can be obtained the value of the current flowing through the inductor. Figure 3.5 shows how the current sensor was connected to the microcontroller.

Attending to the ACS710 specifications, a 100 pF capacitor connected to its FILTER pin makes a bandwidth for the output signal of 120 kHz. Moreover, some capacitors are connected to the power supply input and signal outputs to reduce noise.

The relationship between the inductor current and the voltage read by the microcontroller is given by the following relationship:

\[ I_l (A) = \frac{2(V_{ADC3} - V_{ADC2})}{0.151} \]  \hspace{1cm} (3.9)

This is valid for \(-7.5 \ A \leq I_l \leq 7.5 \ A\) with a total output error of 1.6 % over the reading.

### Power supply

All the control logic and the microcontroller are supplied using both input and output power from the converter. As it is shown in Figure 3.6, a double anode diode with common cathode (MBR2425CT) provides the power necessary to supply the MOSFET drivers and a 3.3 V regulator (NCV4274A).

The NCV4274A is a low dropout voltage regulator that gives 3.3 V at its output with a input range from 4.5 V to 40 V, and a maximum output current of 400 mA. It is used to supply the current sensor, the microcontroller and the Ethernet port logic.

### 3.3 Microcontroller

The chosen microcontroller for controlling the converter was the LPC1768 because it was used in the previous related work (see Section 1.2). The software implemented by the previous teams has been used as a start point for the control algorithm.
The LPC1768 [7] is an ARM Cortex-M3 based microcontroller for embedded applications featuring a high level of integration and low power consumption. The ARM Cortex-M3 is a next generation core that offers system enhancements such as enhanced debug features and a higher level of support block integration.

The LPC1768 is supplied by the 3.3 V signal from the voltage regulator NCV4274A. A set of capacitors and inductors has been connected to the power supply pins in order to filter noise. Moreover, a 12 MHz crystal oscillator provides an external clock signal for the microcontroller and a push button is connected to the reset signal in order to make a hard reset. In Figure 3.7 are shown the main connections of the microcontroller.

The LPC1768 peripherals that can be used with the designed hardware are the following:

- In-System Programming (ISP) module.
- Ethernet MAC with RMII interface.
- Two UARTs with fractional baud rate generation.

![Figure 3.6: Power supply circuit](image)

Figure 3.6: Power supply circuit

![Figure 3.7: Microcontroller main connections](image)

Figure 3.7: Microcontroller main connections
- 6 GPIO (General Purpose I/O) pins with configurable pull-up/down resistors.
- 4 12-bit Analog-to-Digital Converter (ADC) with conversion rates up to 200 kHz.

**Analog to Digital Converters (ADC)**

Four Analog to Digital Converters (ADC) of the LPC1768 are connected to the sensors and their measures are used to know the state of the main variables of the converter. These ADC take 12 bit precision readings at 200 kHz working frequency and hence, the microcontroller can check the value of the state variables with a frequency of 50 kHz without losing precision. ADC0 and ADC1 are used to measure the voltages at the ends of the converter and, ADC2 and ADC3 are connected to the current sensor outputs.

**General Output Pins**

Four pins with digital input and output behavior are connected to the inputs of the MOSFET drivers. This allows to the microcontroller to control the state of the four MOSFET and then to decide how is working the whole converter. Moreover, other two general output pins are connected both to LED diodes in order to provide a visual signal for test programs.

**Ethernet connection**

The Ethernet connection is based in a very used architecture called RMII (Reduced Media Independent Interface). RMII is a standard that addresses the connection of Ethernet physical layer transceivers (PHY) to Ethernet switches or the MAC portion of an end-device Ethernet interface. It reduces the number of signals/pins required for connecting to the PHY from 16 (for an MII-compliant interface) to between 6 and 10. The designed interface (see Figure 3.8) is based in a general construction for a RMII working at 10 MHz speed.

![Ethernet physical circuit](image)

**Figure 3.8: Ethernet physical circuit**

This Ethernet interface, along with the bidirectional behavior of the converter core, provides the capability to integrate the converter in smart grids. Therefore, it is possible to handle the flow of energy in a microgrid [25] between some elements and provide the voltage and current that they need. Moreover, this configuration can provide a tool to develop communication methods for remote control or motorization of the converter.
Serial communication and program flashing

Two serial UART (Universal Asynchronous Receiver-Transmitter) are reserved to used them communication ports to change the converter parameters. These are UART0 and UART1. Moreover, the UART0 serial port can be used for programming or reprogramming the on-chip flash memory, using the boot loader software. This method is called ISP (In-System Programming).

The flash boot loader code is executed every time the part is powered on or reset. The loader can execute the ISP command handler or the user application code. A low level after reset at pin P2.10 is considered an external hardware request to start the ISP command handler. Assuming that power supply pins are on their nominal levels when the rising edge on RESET pin is generated, it may take up to 3 ms before P2.10 is sampled and the decision on whether to continue with user code or ISP handler is made. If P2.10 is sampled low and the watchdog overflow flag is set, the external hardware request to start the ISP command handler is ignored. If there is no request for the ISP command handler execution (P2.10 is sampled high after reset), a search is made for a valid user program. If a valid user program is found then the execution control is transferred to it. If a valid user program is not found, the auto-baud routine is invoked.

![Figure 3.9: Push buttons circuit](image)

Pin P2.10 is used as a hardware request signal for ISP and it can be generated with a push button connected to the P2.10. P2.10 is in high impedance mode after reset, after that, a pull-up resistor provides an external high level to put the pin in a defined state. This configuration and also the reset button one, are shown in the Figure 3.9.

3.4 Overvoltage protection

One of the applications of the converter is to manage the power stored in a supercapacitors battery. The BCAP3000 [1] from MAXWELL TECHNOLOGIES with 3000F were used to build the battery and they allow a maximum voltage between their terminals of 2.7 V (see Section 1.2). To exceed this voltage can damage permanently the supercapacitors. For this reason, a overvoltage circuit has been designed to protect them and it replaces the balancing circuit designed by the previous teams (see Section 1.2).

They are some alternatives to built a overvoltage circuit. A balancing circuit between all pairs can be used, this keeps all the pairs at the same voltage and the 2.7 V maximum voltage can be controlled monitoring the voltage of the battery. This option have low efficiency and much energy is lost during the process of balancing.

Another alternative could be a circuit that switch the energy stored between one pair of capacitors or another with the same principle as a DC converter. This would be the most
efficient alternative but it requires a lot of hardware to make it work. In order to focus the hardware development on the converter, it has been chosen an option which meets the efficiency and the minimal hardware requirements.

![Overvoltage circuit to protect supercapacitors](image)

In Figure 3.10 are shown the overvoltage circuit that can act over the eight pairs of supercapacitors. Each one of them has built with a NPN transistor base-controlled by a 2.7 zener diode. The zener diodes begin conducting when the voltage between the ultracapacitors terminals is close to 2.7 V. Then, the current flowing through them goes into the base terminal of the NPN transistors which start to drive current between the terminals of the supercapacitors.

### 3.5 Hardware implementation

The designed hardware has been implemented in a 160 x 100 mm PCB (*Printed Circuit Board*) and it has been installed in a metallic case to protect the circuit and provide a cooling system for the transistors. In Figure 3.11 is shown a top view of the PCB.

This PCB has been designed using the free software gEDA. The gEDA [26] project has produced and continues working on a full GPL’d suite and toolkit of Electronic Design Automation tools. These tools are used for electrical circuit design, schematic capture, simulation, prototyping, and production. Currently, the gEDA project offers a mature suite of free software applications for electronics design, including schematic capture, attribute management, bill of materials (BOM) generation, netlisting into over 20 netlist formats, analog and digital simulation, and printed circuit board (PCB) layout.

In Appendix ?? is shown the list of components used into the PCB. In Appendix D is shown the complete schematics which include all the designed circuits, jumpers, test probes and holes that the designed PCB has.
Figure 3.11: Top view of the designed PCB
Chapter 4

Control software design

4.1 Implemented features

The Bang-Bang controller has been implemented to work with the designed hardware and pro-
grammed in the LPC1768 microcontroller. The main source code that is shown in Appendix E
added to some functions made by the previous teams, has been programmed to build a converter
that is able to work with the supercapacitors battery. Table 4.1 summarizes the implemented
features and others that have been considered in the hardware design but they have not been
programmed.

<table>
<thead>
<tr>
<th>Achieved</th>
<th>Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unidirectional buck converter</td>
<td>Bidirectional converter</td>
</tr>
<tr>
<td>Unidirectional boost converter</td>
<td>MPPT algorithm</td>
</tr>
<tr>
<td>Current limitation</td>
<td>Ethernet communication</td>
</tr>
<tr>
<td>Techniques for maximize efficiency</td>
<td>Efficiency auto-maximized</td>
</tr>
</tbody>
</table>

Table 4.1: List of programmed features

4.2 Initialization

System clock

The main oscillator is used as the clock source for the CPU, with using a PLL (Phase Locked
Loop). The main oscillator is a crystal oscillator which operates at 12 MHz. This frequency is
boosted to the maximum CPU operating frequency, by the Main PLL (PLL0). The oscillator
output is called osc_clk and the microprocessor clock frequency is referred to as cclk as it is
shown on Figure 4.1.

The PLL0 input (sysclk), what must be in the range of 32 kHz to 50 MHz, is connected the
oscillator output osc_clk and thus it has a frequency of 12 MHz. It is initially divided down
by a value ”N”, which may be in the range of 1 to 256. This input division provides a greater
number of possibilities in providing a wide range of output frequencies from the same input
frequency. In this case it has taken 3 as ”N” value.

Following the PLL0 input divider is the PLL multiplier. This can multiply the input divider
output through the use of a Current Controlled Oscillator (CCO) by a value ”M”, in the range
of 6 through 512. The resulting frequency must be in the range of 275 MHz to 550 MHz. The
multiplier works by dividing the CCO output or pllclk by the value of M, then using a phase-
frequency detector to compare the pllclk to the multiplier input. The error value is used to
adjust the \( pllclk \) frequency. The chosen value for "M" was 50, that gives a \( pllclk \) frequency of 400 MHz. The PLL structure is shown in Figure 4.2.

There are additional dividers at the output of PLL0 to bring the frequency down to what is needed for the CPU and peripherals such as the CCD (CPU Clock Divider). It has taken a value of 4 for the CCD. It divide the \( pllclk \) frequency of the PLL0 and provide a \( cclk \) of 100 MHz to the system. Equation 4.1 summarizes how is obtained the 100 MHz main clock:

\[
\begin{align*}
    f_{osc,clk} &= 12 \text{ MHz} \\
    M &= 50 \\
    N &= 3 \\
    CCD &= 4
    \end{align*}
\]

\[
f_{cclk} = f_{PLL0} = \frac{2 \times M \times f_{osc,clk}}{N \times CCD} = 100 \text{ MHz}
\]  

ADC converters

Four ADC (Analog to Digital Converters) are used to monitor the main variables of the converter. These are:

- ADC0 for output voltage \( V_{out} \).
- ADC1 for input voltage \( V_{in} \).
- ADC2 for zero reference current \( I_{ref} \).
- ADC3 for inductor current \( I_{L} \).
The input clock for each one of the ADC (\(pclk\)) has a frequency of 25 MHz because it was selected the divider by four over the \(cclk\) (100 MHz). This programmable divider is included in each converter to scale this clock to the clock needed by the successive approximation process of the ADC. According to the specification, the maximum frequency for a 12 bit conversion precision is 13 MHz, hence with a working frequency of 25 MHz the precision of the analog to digital conversion is about 10 bit. Moreover, fully accurate conversion requires 65 of these clocks, an then the sampling frequency of each ADC is:

\[
f_{ADC} = \frac{f_{clk, ADC}}{65} = \frac{f_{cclk}}{4 \times 65} \approx 385\ \text{ksamples/s}
\]  

(4.2)

Given that the four state variables have to be known for every decision of the Bang-Bang controller, the maximum control frequency will be:

\[
f_{\text{control, max}} = \frac{f_{ADC}}{4} \approx 96\ kHz
\]  

(4.3)

Moreover the ADC interruption was programmed to read sequentially the ADC from 0 to 3 making a interruption loop.

**Serial communication**

UART0 was configured to provide serial communication and thus the capability of change the converter parameters such as desired output voltage or maximum inductor current, without reprogram the microcontroller firmware.

The baud-rate of UART0 was set to 115200 bps configuring the corresponding registers and following the relationship of the Equation 4.4.

\[
\{f_{cclk} = 100\ MHz,\ UnDLM = 0,\ UnDLL = 37,\ DivAddVal = 7,\ MultiVal = 15\}
\]

\[
\text{baud}_{UART0} = \frac{f_{cclk}}{16 \times (256 \times UnDLM + UnDLL) \times (1 + \frac{DivAddVal}{MultiVal})} \approx 115200\ bps
\]  

(4.4)

Besides, there was set eight bits of data, one stop bit and no parity bit. Therefore the serial configuration is 115200 8N1.

**Other configurations**

Other initial configurations that are done at the beginning of the code are the following:

- Set as digital outputs the control signals. These are connected to the MOSFET drivers and then the microcontroller can control the state of the switches with them.
- Initialize the *Repetitive Interrupt Timer* (RIT). It is used to implement the shoot-through protection and limit the reverse current (Section 4.5).
- Set as digital outputs the pins that are connected to the two LEDs. They can be used as test pins or visual control.
- Initialize the internal variables. They are used to store the state variables and the parameters of the converter \((V_{out},\ V_{in},\ I_{l},\ I_{ref},\ V_{max},\ I_{max},\ V_{ref})\).
4.3 Main loop

After the initialization process, the LPC1768 starts a infinite loop where the required operations to execute the Bang-Bang control take place. In this main loop, the LPC1768 executes continually a function (handle_command) that checks if any command has been received through the UART0. If so, the received command is attended and the corresponding routine is executed (see Section 4.6). This process is executed until a flag (UpdateChannel) is set up by the ADC interruption. This flag indicates that the four ADC have been read and then the Bang-Bang controller function can be executed. In Figure 4.3 is showed the flow diagram of the main loop.

![Flow Diagram of Main Loop](image)

Figure 4.3: Main loop controlled by ADC interruptions

Once the ADC flag has been checked, the obtained values for output and input voltage, and inductor current are low-pass filtered in order to remove noise from the readings. It also can discard some samples detected as glitches caused by the ADC peripheral. This fact was noticed and contrasted with some users of the LPC1768 [27]. Another important factor that has been considered is that both the filtering and the stored values of the state variables are in integer format because it allows working faster than float formats.

After that, another function (MeanValues) is the responsible of calculate the mean values of the state variables every 100000 readings. These values are used to report the state of the converter through the serial port (see Section 4.6) and can be used for MPPT control or other improvements.

Eventually, the Bang-Bang control algorithm is executed and then the microcontroller starts a new iteration of the main loop.
4.4 Bang-Bang Algorithm

The Bang-Bang controller algorithm that is executed in the LPC1768 is shown in the Figure 4.4. This is the final implemented version and it completes the algorithm described in Section 2.4 and Appendix C. It is executed every time that the state variables have been read with the ADC and it decides the state of the MOSFET transistors of the converter.

The steps of the algorithm are the following:

1. If $V_{ref} = 0$ the converter goes to `ALL_OFF` state. Then all the transistors are cut-off and no current is flowing through the inductor. $V_{ref}$ is the desired steady voltage at the output, it can be configured in a range of $0 \leq V_{ref} \leq V_{max}$ and its initial value is $V_{ref} = 0$.

2. If $V_{out} > V_{max}$ the converter goes to `DISCHARGE` state. On one hand, the output capacitors are disconnected of the inductor and they start discharging through the load. On the other hand, the inductor terminals are short-circuited and starts discharging too. This is a security state to avoid overvoltage at the output of the converter. $V_{max}$ determines this maximum voltage, it can be configured in a range of $0 \leq V_{max} \leq 30$ and its initial value is $V_{max} = 25$.

3. If $I_l > I_{max}$ the current flowing through the inductor must be reduced. $I_{max}$ sets the maximum current at the inductor, it can be configured in a range of $0 \leq I_{max} \leq 6$ and its initial value is $I_{max} = 1$. There are two alternatives depending on the input and output voltage.
   
   a) If $V_{out} > V_{in}$ the converter goes to `BOOST_OFF` state. As it has been noted in Section 2.3, in this case the current can be reduced without disconnecting the inductor from the input supply.
b) Otherwise the converter goes to \textit{BUCK.OFF} state and the inductor is disconnected from the input supply.

4. If $V_{\text{ref}} > V_{\text{in}}$ the converter must work in boost mode.

a) If $V_{\text{out}} > V_{\text{ref}}$ the converter must reduce the voltage at its output. It is possible going to \textit{BOOST.OFF} or \textit{BOOST.SOFT} state depending on the current flowing through the inductor. The choice between one or other provides the capability to protect the converter against the shoot-through and reverse current phenomenons (see Section 4.5).

b) Otherwise the converter goes to \textit{BOOST.ON} state. Then the current necessary to increase the output voltage until the $V_{\text{ref}}$ value is provided by the input supply.

5. Eventually, if $V_{\text{ref}} \leq V_{\text{in}}$ the converter must work in buck mode. As well as in boost mode, the converter must go to the proper state depending on the relationship between $V_{\text{out}}$ and $V_{\text{ref}}$, and the current flowing through the inductor. \textit{BUCK.OFF}, \textit{BUCK.SOFT} and \textit{BUCK.ON} are the equivalent states for buck mode.

### 4.5 Switch management

Every state of the converter corresponds to a combination of cut-off or active state of the MOSFET transistors. Table 4.2 summarizes these relationships.

<table>
<thead>
<tr>
<th>State</th>
<th>N-channel 1</th>
<th>P-channel 1</th>
<th>N-channel 2</th>
<th>P-channel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUCK.OFF</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>BUCK.SOFT</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>BUCK.ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>BOOST.OFF</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>BOOST.SOFT</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>BOOST.ON</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>DISCHARGE</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>ALL.OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Table 4.2: State of MOSFET transistors for each state of the converter

The transition between these states can induce a negative effect over the efficiency of the converter. It is called shoot-through and there has been implemented a mechanism to minimize its effects. Moreover, two more states have been added to the ones showed in Table 2.1 and they are necessary to implement the Bang-Bang control in order to avoid another negative effect that is called reverse current. Both effects appear in synchronous converters and they have to be minimized to get a higher efficiency than in the non-synchronous converters.

**Shoot-through**

Shoot-through is defined as the condition when both MOSFET transistors (P-channel and N-channel) are either fully or partially turned on, providing a path for current to “shoot through” from $V_{\text{in}}$ or $V_{\text{out}}$ to GND (see Figure 4.5) [28].

To minimize shoot-through, synchronous buck or boost regulators employ one of two techniques to ensure “break before make” operation of the transistors:
1. Fixed “dead-time”. A MOSFET is turned off, then a fixed delay is provided before the low-side is turned on. This circuit is simple and usually effective, but suffers from its lack of flexibility if a wide range of MOSFET gate capacitances are to be used with a given controller. Too long a dead-time means high conduction losses. Too short a dead time can cause shoot-through. A fixed dead-time typically must err on the “too long” side to allow MOSFET transistors to fully discharge before turning on the complementary MOSFET.

2. Adaptive gate drive. This method looks at the gate voltage of the MOSFET that is being driven off to determine when to turn on the complementary MOSFET. Theoretically, adaptive gate drives produce the shortest possible dead-time for a given MOSFET without producing shoot-through.

The fixed “dead-time” method was chosen because it can be implemented easily without adding more hardware to the converter (an additional ADC is needed for each transistor in method 2). It has been set 2 µs as ”dead-time” because it is enough to let discharging the gate of the active transistor. In Figure 4.6 are shown the behavior of the control signals in the shoot-through protection.

The ”dead-time” is controlled by the RIT peripheral and it is set every time that the state of the converter changes. However, if the shoot-through effect is not present in the current state transition, the microcontroller ignores this fixed ”dead-time” and changes immediately to the corresponding state (see Figure 4.8).
Reverse current

Reverse conduction of synchronous rectifiers is a major factor in converter efficiency [29]. In some cases, the reverse conduction can even cause the destruction of the converter. When the converter is operating in full heavy load (like with the ultracapacitors battery), the synchronous rectifiers are always more efficient than the standard Schottky diode solution (assuming the converter is operating at a reasonable frequency of less than 500 kHz).

![Figure 4.7: Reverse Current effect](image)

The implemented method to avoid this effect (see Figure 4.7) is to let the transistors that allow flowing the reverse current to be active only during a short period of time, in this case, it is about 8 $\mu$s. This happens during the **BUCK_OFF** and **BOOST_OFF** states. The rest of time until the next execution of the Bang-Bang algorithm, these transistors (N-channel 1 or P-channel 2) are in cut-off state and the current can flow through its body diodes. This corresponds to **BUCK_SOFT** and **BOOST_SOFT** states.

After that, in the next decision of the Bang-Bang controller, it is checked if the inductor current is positive. If so, the microcontroller repeats the previous process again. If not, the converter remains in a "SOFT" state. In Figure 4.8 are shown how both reverse current and shoot-through protection work.

![Figure 4.8: Reverse current and shoot-through protection](image)

1. Switch in **ON** state
2. Shoot-through protection about 2$\mu$s duration
3. Allow N-channel MOSFET conducting for 8$\mu$s
4. N-channel MOSFET acts as a diode
5. Shoot-through protection is not necessary
6. Switch is **ON** and no shoot-through protection

32
4.6 Serial bus commands

A set of basic commands has been programmed in order to configure the parameters of the controller without flashing the microcontroller and monitor the evolution of the state variables through a serial bus.

The serial communication software protocol was designed by MinNE 3 team and its described in [30]. This code implements a serial interface that can be connected to another device with a Future Technology Devices International (FTDI) interface. A simply serial communication software such as Minicom is required in the device to send and receive data from the converter. The parameters of the serial communications are 115200 bps and 8N1 as it was established in Section 4.2.

Table 4.3 summarizes the commands implemented to handle the internal parameters of the converter.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Shows a list of supported commands.</td>
</tr>
<tr>
<td>vref x</td>
<td>Sets the reference voltage to x. This is the desired output voltage at the output of the converter.</td>
</tr>
<tr>
<td>imax x</td>
<td>Sets the maximum current inductor to x. Determines how much current can flow through the inductor and it has effect over how efficient is the converter.</td>
</tr>
<tr>
<td>vmax x</td>
<td>Sets the maximum voltage at the output to x. This provide a safe mechanism against overvoltage.</td>
</tr>
<tr>
<td>par</td>
<td>Shows the value in float format of the three configurable parameters ($V_{ref}$, $I_{max}$ and $V_{max}$).</td>
</tr>
<tr>
<td>val</td>
<td>Shows the value in float format of the three state variables ($V_{out}$, $V_{in}$ and $I_l$) of the converter.</td>
</tr>
<tr>
<td>↑</td>
<td>When this key is pressed, the last command appears in the command prompt and the previous ones if it is pressed successively.</td>
</tr>
</tbody>
</table>

Table 4.3: Serial commands to configure the converter

The commands that are related with parameters and variables also execute functions that have to change the format of the values that they carry. As it was mentioned before, variables and parameters are handled in integer format because the LPC1768 can carry out the operations faster than if it uses the float format. Otherwise, float format is better to show the values to the user of the converter and to configure the parameters too. It is easier to handle than the integer format. Therefore, $vref$, $imax$ and $vmax$ commands transform float values into stored integer values, and $par$ and $val$ transforms integer into float values.
Chapter 5

Prototype evaluation

A prototype has been built in order to check if the Bang-Bang controller satisfies the requirements exposed at the beginning of the document. This prototype has the same structure than the designed hardware except the Ethernet interface. Some tests have been carried out with a resistor as load at the output and with the supercapacitors battery later.

5.1 General behavior

In order to check the correct performance of the prototype, it has been done the following setup:

- A DC power source sets the input voltage and provides the power to supply the prototype.
- A 18.5 Ω resistor acts as the output load.
- A multimeter is used to measure the DC component of the output voltage.
- Two multimeters more are used to measure the input and the output voltage.
- A mixed signal oscilloscope is used to:
  - Measure the AC component of the output voltage with its analog channel 1 (A1).
  - Measure the output of the current sensor with its analog channel 2 (A2).
  - Check the operating frequency of the converter with its digital channel 0 (D0).
  - Monitor the control signal of the N-MOS 1 with its digital channel 2 (D2).
  - Monitor the control signal of the P-MOS 1 with its digital channel 3 (D3).
  - Monitor the control signal of the N-MOS 2 with its digital channel 4 (D4).
  - Monitor the control signal of the P-MOS 2 with its digital channel 5 (D5).

Several previous considerations must be taken in account before the evaluation of the converter. First one is that it can be calibrated in order to reduce the systematic error produced in the ADC readings. A sample taken by the ADC peripheral has to be scaled using the value of the supply voltage to know the value of the voltage that wants to be measured. If it is assumed that this voltage is exactly $V_{DD} = 3.3 \, V$, a systematic error in the measure will be produced. The second consideration is that an error smaller than the 5 % of the measure can be considered within the acceptable margin.

In first place, it has been checked the performance of the converter working in buck mode. The converter parameters have been set up to $V_{ref} = 5 \, V$, $I_{max} = 1 \, A$ and $V_{max} = 25 \, V$ with
the corresponding serial commands (see Section 4.6). Moreover, the input voltage have been set to 10 V.

The measured DC voltage at the output was $V_{out} = 4.978 \text{ V}$. It means a error smaller than 0.5 % so it is within the acceptable error margin.

![Figure 5.1: Converter working in buck mode](image)

Furthermore, the following observations can be made looking at Figure 5.1:

1. There are some glitches in voltage and current. It is a common effect in all switched converters and it is caused by the reverse recovery charge of the transistors and other parasitic elements. The analysis of the results will ignore this effect as it was done in [31].

2. The output ripple is about $V_{ripple} = 50 \text{ mV}$ and it means 1 % of the DC voltage. It fits the specifications of the hardware.

3. About the inductor current, it is positive so the reverse current protection accomplishes its function and the maximum value is about 1.3 A (from Equation 9). It means an error of 30 % and it does not fit to the theoretical study. However, it can be observed that the average of the current does not exceed 1 A.

4. The working frequency moves into a range (the value showed in the figure was not constant) but it has been observed that its average is around 71 kHz.

5. It is also shown how change the control signals of the transistors to avoid the reverse current and the delay between these signals that provides the shoot-through protection.

The same procedure has been followed to check the boost performance. The configured parameters have been $V_{ref} = 12 \text{ V}$, $I_{max} = 1 \text{ A}$ and $V_{max} = 25 \text{ V}$ and the input voltage has
been set up to 8 V. The measured output voltage with the multimeter was 12.402 V and it means an acceptable error of 3 % over the desired value.

![Figure 5.2: Converter working in boost mode](image)

From the Figure 5.2 it follows:

1. It was expected that there are glitches in voltage and current waveforms too.
2. The output ripple is about $V_{\text{ripple}} = 40 \, \text{mV}$ and it means less than 1 % of the DC voltage. It also fits the specifications of the hardware.
3. The inductor current is also positive and the maximum value is almost 2 A. It does not fit to the theoretical study neither. However, it can be observed that the average current is about 1 A.
4. The working frequency and reverse current performance, as well as shoot-through protection, have a similar behavior compared with the buck converter mode.

Eventually, Figure 5.3 shows the converter working under current limitation. This mode works when the required current at the output is higher than the maximum current configured such as it happens when the converter is connected to the supercapacitors battery. The results are similar to the ones showed before and the control signals work is as it was described in the Bang-Bang theory.

### 5.2 Efficiency considerations

The efficiency of the converter will be analyzed first under the conditions of light load. These include low resistances such as supercapacitors battery or the 18.5 Ω probe resistor connected before.
In Figure 5.4 is represented the evolution of the efficiency of the converter in front of the reference voltage. It can be appreciate that the efficiency is higher than 90% for high output voltages or, in other way, for high input-output power transfer. On the contrary, efficiency is worse for low power transfer.
This is a expected behavior because of the high idle power consumption. All the control logic devices have a quiescent current what causes this degradation (or at least a part of it) on the converter efficiency.

The main sources of this idle consumption (theoretically from the information in their data sheets) are:

- LPC1768 microcontroller: 42 mA (Active mode with clock at 100 MHz and PLL enabled) [32].
- CWX823 50 MHz crystal controlled oscillator: 30 mA [33].
- DP83848 Ethernet controller: 14 mA (Power Down Mode) [34].
- ACS710 sensor: 9 mA [24].
- FAN3278 MOSFET drivers: 1.8 mA each one [23].
- Others: NCV4274A 3.3 V Regulator, LEDs, voltage divisors and leakage current of all passive components.

Practically, the measured idle current is 90 mA.

Another negative effect over the converter efficiency are the losses on MOSFET body diodes. As is has noted before, they are used to block the reverse current but, when they are in conduction, their high forward voltage become in another source of power losses.

Over high load conditions, it can be appreciated the same effect as before. In the case of this converter, a high load is when the load at the output needs a current lower or with the same magnitude than the idle current. In Figure 5.5, there is represented the converter efficiency with a 110 Ω resistor connected at the output.

![Figure 5.5: Efficiency of the converter with high load](image)

This resistor takes 228 mA at 25 V and if the previous idle current is considered, it can explain the efficiency what is shown in the Figure 5.5.

In Chapter 7, the alternatives to reduce the idle power consumption will be discussed.
5.3 Ultracapacitors handling

One of the main goals of this project is to manage a supercapacitors battery. Hardware and software have been designed to charge and discharge the supercapacitors and also it has to protect it against overvoltages and overcurrents.

The overvoltage protection circuit has to be active when one of the supercapacitors reaches 2.7 V between its terminals. In Figure 5.6, it is shown the voltage-current characteristic of this circuit for every capacitor. It has been noted that when the voltage is near to 2.7 V, the current starts flowing to the transistor instead through the capacitor, avoiding to overcharge it.

![Figure 5.6: Voltage-Current characteristic of the overvoltage circuit](image)

In Figure 5.7 is shown the evolution of the voltage of the eight pairs of supercapacitors. The battery was charged from 0 V to 19 V in 3 hours and 45 minutes. The input voltage was 10 V generated with the power source and the average charging current 1.25 A.

![Figure 5.7: Supercapacitors voltage limited by the overvoltage circuit](image)

The supercapacitors battery acts as a light load and, as it was discussed in previous section, and in Figure 5.8 it can be observed the same behavior as with the 18.5 Ω resistor.
Figure 5.8: Efficiency of the converter working with the supercapacitors battery
Chapter 6

Conclusions

Once the converter has been checked and tested with different loads at its output, it is time to analyze what is the result of all the control theory, hardware and software design.

The DC/DC converter is able to step up and down the input voltage in the defined range. It can manage voltages from 0 V to 25 V at both sides, input and output, and it works regardless of the output load. The designed controller based on Bang-Bang control has been perfectly adapted to abrupt changes on load, input or reference voltage. However, there is one issue what was not expected: one of the voltages, input or output, must be higher than 7 V because of the MOSFET drivers. According to the drivers data sheet [23], they can work with a minimum voltage of 4.5 V but it seems that practically, this value is 7 V.

Apart from that, the hardware fits the initial specifications about absolute ratings. It can support up to 27 V and 6 A. Moreover, it can work with the supercapacitors battery because it can avoid the overvoltage on the capacitors and control the current going into them in order to avoid damaging the converter. Nevertheless, the idle power consumption is high and it must be reduced.

About the implemented software, the average current of the converter can be controlled but not the maximum. This difference between the Bang-Bang controller simulations and the prototype evaluation is produced by the existing and inescapable delay between the measures of the state variables and the change of the control signal of the MOSFET transistors due to the needed microcontroller operations. This fact produces that the value of \( i_{\text{max}} \) is proportional to the average current flowing through the converter but this relationship still has to be quantified.

It has been noted that the designed hardware has a lot of potential so it can be used in many applications which involves energy management. It was designed to fit a control based in the developed Bang-Bang algorithm but, besides, it can be used with other kinds of control, with digital signals or, also, with PWM signals. The digital control output pins which have been taken in the design, have PWM peripherals what can be programmed in software.

The "MCU controlled DC-DC buck/boost converter for supercapacitors" is a device what can be used in many applications. Manage a supercapacitors battery is its main application but it can be used by researchers or companies in the field of the power and energy management. Used components of the converter are low cost and non polluting. The designed and built PCB with the components cost about 800 SEK (90 EUR) and it can be reduced changing some components with others with the similar characteristics.

The supercapacitors batteries can replace the other kins of batteries some day. The evolution of the capacitors technology are evolving day after day and, in the future, all indications are that supercapacitors can store as much energy as the other batteries. They would be able to supply electric vehicles, making them totally "green" because right now they include batteries with polluting elements. Then a converter such as the described in this document will be necessary
to manage the power in this kind of vehicles.

As it has been noted before, the converter can be used also to manage energy in smart grids. This is one of the fields related to power management which is growing up at the moment. The LPC1768 microcontroller has a Ethernet controller what can be programmed by researchers to implement a protocol what can be used to communicate with other nodes in a smart grid. This grids will have the function of distributing the energy from places where renewable energy is produced in small amounts to the regions where it is needed or store them. This mechanism will be able to manage the energy more efficiently.
Chapter 7

Future work

There are some issues what may be improved in the converter. In first place, some changes may be it more efficient, reducing the idle power consumption. These are the following:

- Avoid using LEDs or, at least, do not make them working all the time.
- Make the microcontroller go into sleep mode when it is not doing some calculations and there are only working the peripherals.
- Include PMOS transistors to connect or disconnect the power supply of some components (MOSFET drivers, current sensor, Ethernet driver, ...) making them working only when they are required.

Moreover, Schottky diodes with low forward voltage should be connected in parallel with the four MOSFETS. They will improve the efficiency of the hardware when it is working to avoid the reverse current. Other MOSFET drivers with similar characteristics might be used instead of the FAN3278 in order to avoid the restriction of at least 7 V in one of the sides of the converter.

In addition, another microcontroller with faster ADC peripherals would provide a less noise, faster response and more accurate current control. Also, bidirectional power management may be easily implemented changing the corresponding control signals from one switch to the other one in the current source code.

Otherwise, the Ethernet peripheral of the LPC1768 microprosessor should be programmed to implement the network communication. Besides, a smart grid management protocol such as DNP3 should also be included to manage the power between other grid nodes.

Eventually, a MPPT (Maximum Power Point Tracking) algorithm may be implemented to get the maximum power from a solar panel. There are some different kinds of algorithms but which best fits the behavior of the converter is the Incremental Conductance method (IncCon). This mechanism consists in vary the input current of the converter in order to find the maximum point of power supplied from the solar panel. This variation can be performed using the programmable maximum current limitation; it can be varied from zero to the maximum current that supports the components of the converter.
Bibliography


Appendix A

Comparative C pseudo-code between Bang-Bang and Sliding Mode Control approaches

```c
//*** BANG-BANG CONTROL VS SLIDING MODE CONTROL ***/
//*** ONLY FOR BUCK CONVERTER ***/

/*
   Inputs and constants used:
   * Vout – Measured output voltage
   * Il – Measured inductor current
   * Vref – Desired output voltage
*/

/* 0 for switch OFF, 1 for switch ON */
bool buck = 0:

//*** BANG-BANG CONTROL ***/
if (Vout > Vref)
   buck = 0;
else
   buck = 1;

//*** SLIDING MODE CONTROL ***/
/*
   Constant that controls the stability and speed of
   the converter (typically 2/(R*C)) (depends on the load)
*/
#define c (2/(R*C));

/* Voltage Error */
int e = Vout - Vref;

/* Derivative voltage error \(d = e' = vout' = ic/C = (il - Ir)/C\) (depends on the load)
*/
int d = highpassfilter(Il)/C;
```
bool switchingsurface = e*c + d;
if(switchingsurface > 0)
    buck = 0;
else
    buck = 1;
Appendix B

Bang-Bang controller implemented with MATLAB Simulink blocks

Figure B.1: Converter Bang-Bang controller

Figure B.2: Operating state makers
Figure B.3: Proposed sub-controller to decide between buck or boost converter
Appendix C

Bang-Bang controller implemented in C pseudo-code

/* This C pseudo-code implements a Bang–Bang controller with current limitation for a non-inverting buck/boost converter */

/* Inputs and constants used:
* Vout – Measured output voltage
* Il – Measured inductor current
* Vin – Measured input voltage
* Vref – Desired output voltage
* Vmax – Configured maximum output voltage
* Imax – Configured maximum inductor current */

/* 0 for switch OFF, 1 for switch ON */
bool buck = 0;
bool boost = 0;

/* Definitions for each one of the switch combinations */

#define State1() (buck = 0; boost = 0)
#define State2() (buck = 1; boost = 0)
#define State3() (buck = 1; boost = 1)
#define State4() (buck = 0; boost = 1)

/* Safe maximum voltage limitation (it uses configuration number 4) */
if (Vout > Vmax)
    State4();
/ * Current limitation */
else if (I_l > I_{\text{max}})
{
    /* If output is over input it’s possible to limit the current with state 2 */
    if (V_{\text{out}} > V_{\text{in}})
        State2();
    /* else it must be limited with state 1 */
    else
        State1();
}
else
{
    if (V_{\text{ref}} > V_{\text{in}})
        {
            /* OFF mode (state 2 for boost converter) */
            if (V_{\text{out}} > V_{\text{ref}})
                State2();
            /* ON mode (state 3 for boost converter) */
            else
                State3();
        /* Buck single converter */
    else
        {
            /* OFF mode (state 1 for buck converter) */
            if (V_{\text{out}} > V_{\text{ref}})
                State1();
            /* ON mode (state 2 for buck converter) */
            else
                State2();
        }
Appendix D

Schematics

D.1 Overvoltage circuit schematics

Ultracapacitors Overvoltage Protection Circuit
D.2 Main circuits schematics
D.3 Ethernet controller schematics
Appendix E

Source code

These are the main source files of the implemented software. The rest of the code can be downloaded from the SVN repository (https://repo0.svn.ict.kth.se/csd/smartgrid).

E.1 main.c

/*@main.c
Created on: 14/12/2011
Author: Jorge Querol*/

#include "../LPC17xx.h"
#include "../adc/adc.h"
#include "../bang/bang.h"
#include "../control/control.h"

extern void SystemInit(void);
extern void comm_init(void);
extern void handle_command(void);
extern void welcomeMsg(void);
extern void set_echo(void);
extern uint8_t printchar(const char *charBuf);

extern int UpdateChannel;
extern int Vout, Vin, Il;
int Iref;

int main(void)
{
    /* Start led connected to P1.29 and P1.18 */
    LPC_GPIO1->FIODIR |= 0x20040000;
    LPC_GPIO1->FIOSET |= (1 << 29);

    /* Start main program */
    SystemInit();
    comm_init();
    handle_command();
    welcomeMsg();
    set_echo();
    printchar("Hello World!
");

    /* Sleep forever */
    while (1) {
        printchar("... ");
    }

    return 0;
}
LPC_GPIO1->FIOLR |= (1 << 18);

SystemInit(); // lpc1768_startup.c

GPIOInit();
TimerInit();
ValueInit();
ADCInit();

comm_init();
// welcomeMsg();
set_echo();

while (1)
{
    handle_command();
    if (UpdateChannel >= 3)
    {
        UpdateChannel = -1;
        ADCRead(0);
        Vout = ADCValues(0);
        Vin = ADCValues(1);
        Iref = ADCValues(2);
        Il = ADCValues(3) - Iref;
        MeanValues();
        BangBang();
        LPC_GPIO1->FIOPIN ^= (1 << 29);
    }

    return 0;
}
E.2 control.c

/*
 * control.cimax 6
 *
 * Created on: 08/03/2012
 * Author: Jorge Querol
 */

#include "./LPC17xx.h"
#include "./bang/bang.h"
#include "./control/control.h"

#ifdef ShowPeriodically
#include "./comm/comm.h"
#endif

int Vref, Imax, Vmax;
int Vout, Vin, Il;
int Vo, Vi, Io, Ii;
int i, Vom, Vim, Iom, Iim;

extern int ADC[4];
extern state_t CurrentState;

void ValueInit(void)
{
    Vref = 0;
    Imax = (int)(I_IMAX * 4095 * 0.151) /(Vdd * 2);
    Vmax = (int)(I_VMAX * 4095 * 3) /(Vdd * 28);
    Vo = 0;
    Vi = 0;
    Io = 0;
    Ii = 0;
    i = 1;
    Vom = 0;
    Vim = 0;
    Iom = 0;
    Iim = 0;
}

void BangBang(void)
{
    if (Vref > 0)
    {
        /* Safe maximum voltage limitation
           (it uses configuration number4) */
        if (Vout > Vmax)
SetState(DISCHARGE);
/* Current limitation */
else if(II > Imax)
{
    /* If output is over input it's possible to
     * limit the current with state 2 */
    if(Vout > Vin)
        SetState(BOOST_OFF);
    /* else it must be limited with state 1 */
    else
        SetState(BUCK_OFF);
}
else
{
    /* Boost single converter */
    //
    if(operatingzone(Vin, Vref, Vout))
    if(Vref > Vin)
    {
        /* OFF mode (state 2 for boost converter) */
        if (Vout > Vref)
        {
            if(II > 10)
                SetState(BOOST_OFF);
            else
                SetState(BOOST_SOFT);
        }
        /* ON mode (state 3 for boost converter) */
        else
            SetState(BOOST_ON);
    }
    /* Buck single converter */
    else
    {
        /* OFF mode (state 1 for buck converter) */
        if (Vout > Vref)
        {
            if(II > 10)
                SetState(BUCK_OFF);
            else
                SetState(BUCK_SOFT);
        }
        /* ON mode (state 2 for buck converter) */
        else
            SetState(BUCK_ON);
    }
}
else
    SetState(ALL_OFF);
```c
void MeanValues(void)
{
    float aux;

    i++;
    Vom += Vout;
    Vim += Vin;
    if(((LPC_GPIO2->FIOPIN & 0x0008) || (CurrentState == BOOST_SOFT))
        Iom += Il;
    if(LPC_GPIO2->FIOPIN & 0x0002)
        Iim += Il;
    if((i % 100000) == 0)
    {
        Vo = Vom / 100000;
        Vi = Vim / 100000;
        Io = Iom / 100000;
        Ii = Iim / 100000;
        Vom = 0;
        Vim = 0;
        Iom = 0;
        Iim = 0;
    }

    #ifdef ShowPeriodically
    if(i >= 100000)
    {
        i = 1;
        printf("\nVo: ");
        aux = (Vo * Vdd * 28) / (4095 * 3);
        printf_float(&aux);
        printf("Vin: ");
        aux = (Vi * Vdd * 28) / (4095 * 3);
        printf_float(&aux);
        printf("Io: ");
        aux = (Io * Vdd * 2) / (4095 * 0.151);
        printf_float(&aux);
        printf("Ii: ");
        aux = (Ii * Vdd * 2) / (4095 * 0.151);
        printf_float(&aux);
    }
    #endif
}
```
# E.3 

```c
/*
 * bang.c
 * Created on: 14/12/2011
 * Author: Jorge Querol
 */

#include "../LPC17xx.h"
#include "../bang/bang.h"

state_t CurrentState ;

void GPIOInit ( void )
{
    LPC_PINCON->PINSEL4 &= ~0x0000000FF ;
    // Set GPIO control as digital outputs
    LPC_GPIO2->FIODIRL |= 0x000F ;
    // Set P2.0, P2.1, P2.2, P2.3 as outputs
    LPC_GPIO2->FIOPINL &= ~0x000F ;
    // Set OFF all Buck and Boost switches
    CurrentState = ALL_OFF;
}

void TimerInit ( void )
{
    LPC_SC->PCONP |= (1 << 16);  
    // Set up Power for RIT
    LPC_SC->PCLKSEL1 |= (1 << 26);  
    // Peripheral Clock Enabled
    LPC_RIT->RICTRL &= ~0x0C ;  
    // Disable Repetitive Interrupt Timer (RIT)
    LPC_RIT->RICTRL |= 0x01 ;  
    // Clear Repetitive Interrupt Timer (RIT)
    NVIC_EnableIRQ( RIT_IRQn ) ;
    NVIC_SetPriority( RIT_IRQn , 1 ) ;
}

void RIT_IRQHandler ( void )
{
    LPC_RIT->RICTRL &= ~0x08 ;  
    // Disable Repetitive Interrupt Timer (RIT)

    // Shoot-through & Reserve current protection
    if ( CurrentState == BUCK_OFF )  
    // Buck L ON Buck H OFF − Boost L OFF Boost H ON
    {
        LPC_GPIO2->FIOSETL = 0x0009 ;
    }
```
LPC_RIT->RICTRL |= 0x09;
   // Enable and clear interrupt
LPC_RIT->RICOMPVAL = LPC_RIT->RICOUNTER + 700;
   // 8us delay (100 cycles +)
CurrentState = BUCK_SOFT;
}
else if (CurrentState == BUCK_SOFT)
   // Buck L OFF Buck H OFF - Boost L OFF Boost H ON
{
   LPC_GPIO2->FIOCLRL = 0x0001;
   LPC_GPIO2->FIOSETL = 0x0008;
}
else if (CurrentState == BUCK_ON)
   // Buck L OFF Buck H ON - Boost L OFF Boost H ON
   LPC_GPIO2->FIOSETL = 0x000A;
else if (CurrentState == BOOST_OFF)
   // Buck L OFF Buck H ON - Boost L OFF Boost H ON
{
   LPC_GPIO2->FIOCLRL = 0x000A;
   LPC_RIT->RICTRL |= 0x09;
   // Enable and clear interrupt
   LPC_RIT->RICOMPVAL = LPC_RIT->RICOUNTER + 700;
   // 8us delay (100 cycles +)
   CurrentState = BOOST_SOFT;
}
else if (CurrentState == BOOST_SOFT)
   // Buck L OFF Buck H ON - Boost L OFF Boost H OFF
{
   LPC_GPIO2->FIOCLRL = 0x0008;
   LPC_GPIO2->FIOSETL = 0x0002;
}
else if (CurrentState == BOOST_ON)
   // Buck L OFF Buck H ON - Boost L ON Boost H OFF
   LPC_GPIO2->FIOSETL = 0x0006;
else if (CurrentState == DISCHARGE)
   // Buck L ON Buck H OFF - Boost L ON Boost H OFF
   LPC_GPIO2->FIOSETL = 0x0005;
}

void SetState(state_t state)
{
   if (state == BUCK_OFF)
      // Buck L ON Buck H OFF - Boost L OFF Boost H ON
   {
      if (LPC_GPIO2->FIOPINL & 0x0002)
         LPC_GPIO2->FIOCLRL = 0x0002;
      else
         LPC_GPIO2->FIOSETL = 0x0001;
   }
if(LPC_GPIO2->FIOPINL & 0x0004)  
    LPC_GPIO2->FIOCLRL = 0x0004;
else
    LPC_GPIO2->FIOSETL = 0x0008;
}
else if(state == BUCK_SOFT)  
    // Buck L OFF Buck H OFF – Boost L OFF Boost H ON
    {
        if(LPC_GPIO2->FIOPINL & 0x0001)  
            LPC_GPIO2->FIOCLRL = 0x0001;
        if(LPC_GPIO2->FIOPINL & 0x0002)  
            LPC_GPIO2->FIOCLRL = 0x0002;
        if(LPC_GPIO2->FIOPINL & 0x0004)  
            LPC_GPIO2->FIOCLRL = 0x0004;
        else
            LPC_GPIO2->FIOSETL = 0x0008;
    }
else if((state == BUCK_ON) || (state == BOOST_OFF))  
    // Buck L OFF Buck H ON – Boost L OFF Boost H ON
    {
        if(LPC_GPIO2->FIOPINL & 0x0001)  
            LPC_GPIO2->FIOCLRL = 0x0001;
        else
            LPC_GPIO2->FIOSETL = 0x0002;
        if(LPC_GPIO2->FIOPINL & 0x0004)  
            LPC_GPIO2->FIOCLRL = 0x0004;
        else
            LPC_GPIO2->FIOSETL = 0x0008;
    }
else if(state == BOOST_SOFT)  
    // Buck L OFF Buck H ON – Boost L OFF Boost H OFF
    {
        if(LPC_GPIO2->FIOPINL & 0x0001)  
            LPC_GPIO2->FIOCLRL = 0x0001;
        else
            LPC_GPIO2->FIOSETL = 0x0002;
        if(LPC_GPIO2->FIOPINL & 0x0004)  
            LPC_GPIO2->FIOCLRL = 0x0004;
        if(LPC_GPIO2->FIOPINL & 0x0008)  
            LPC_GPIO2->FIOCLRL = 0x0008;
    }
else if(state == BOOST_ON)  
    // Buck L OFF Buck H ON – Boost L ON Boost H OFF
    {

68
if(LPC_GPIO2->FIOPINL & 0x0001)
    LPC_GPIO2->FIOCLRL = 0x0001;
else
    LPC_GPIO2->FIOSETL = 0x0002;

if(LPC_GPIO2->FIOPINL & 0x0008)
    LPC_GPIO2->FIOCLRL = 0x0008;
else
    LPC_GPIO2->FIOSETL = 0x0004;
}
else if(state == DISCHARGE)
    // Buck L ON Buck H OFF - Boost L ON Boost H OFF
{
    if(LPC_GPIO2->FIOPINL & 0x0002)
        LPC_GPIO2->FIOCLRL = 0x0002;
    else
        LPC_GPIO2->FIOSETL = 0x0001;

    if(LPC_GPIO2->FIOPINL & 0x0008)
        LPC_GPIO2->FIOCLRL = 0x0008;
    else
        LPC_GPIO2->FIOSETL = 0x0004;
}
else if(state == ALL_OFF)
    // Buck L OFF Buck H OFF - Boost L OFF Boost H OFF
    LPC_GPIO2->FIOCLRL = 0x000F;

if(state != CurrentState)
{
    LPC_RIT->RICTRL |= 0x09;
    // Enable and clear interrupt
    LPC_RIT->RICOMPVAL = LPC_RIT->RICOUNTER + 100;
    // 2us delay (100 cycles +)
    CurrentState = state;
}
}
#include "../LPC17xx.h"
#include "../adc/adc.h"

int ADC[ADCChannels];
int UpdateChannel;

int pos[ADCChannels];
int glitches[ADCChannels];
int Buffer[2][GlitchBuffer][ADCChannels];

void ADCvaluesInit(void);

void ADCInit(void)
{
    LPC_PINCON->PINSEL1 = 0x00154000;
    // Select ADC function for pins (ADC0−3)
    LPC_PINCON->PINSEL3 = 0xF0000000;
    // Select ADC function for pins (ADC4−5)
    LPC_PINCON->PINMODE1 = 0x0002A8000;
    // Neither pull−up nor pull−down resistor
    LPC_PINCON->PINMODE3 = 0xA00000000;
    // Neither pull−up nor pull−down resistor
    LPC_SC->PCONP |= (1 << 12);
    // Set up bit PCADC
    LPC_SC->PCLKSEL0 = (01 << 24);
    // PCLK = CCLK (102 MHz)
    /* 7−0 SEL
        * 15−8 CLKDIV = 7 (12.25 MHz)
        * 15−8 CLKDIV = 3 (25.5 MHz)
        * 16 BURST software
        * 21 PDN on (ADC on)
        */
    LPC_ADC->ADCR = 0x00200300;
    LPC_ADC->ADINTEN = 0x00000100;
    // ADC Interrupt Enabled
    NVIC_EnableIRQ(ADC_IRQn);
    NVIC_SetPriority(ADC_IRQn, 2);
    ADCvaluesInit();
    // Init ADC glitch filter
    UpdateChannel = −1;
ADCRead(0);
    // Start first conversion
}

void ADCRead(unsigned char ADC)
{
    LPC_ADC->ADCR &= ~(0x000000FF);
    // Remove ADC selected
    LPC_ADC->ADCR |= (1 << ADC);
    // Select ADC
    LPC_ADC->ADCR |= (1 << 24);
    // Start conversion
}

void ADC_IRQHandler(void)
{
    int Channel;
    Channel = (LPC_ADC->ADGDR >> 24) & 0x00000007;
    ADC[Channel] = (LPC_ADC->ADGDR >> 4) & 0x00000FFF;
    UpdateChannel = Channel;
    Channel++;
    if (Channel < ADCChannels)
        ADCRead(Channel);
    else
    {
        LPC_GPIO1->FIOPIN ^= (1 << 29);
        //ADCRead(0);
        /**<
    }
}

void ADCvaluesInit(void)
{
    int i, j, k;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < GlitchBuffer; j++)
        {
            for (k = 0; k < ADCChannels; k++)
            {
                Buffer[i][j][k] = 0;
            }
        }
    }
    for (i = 0; i < ADCChannels; i++)
    {
        pos[i] = 0;
        glitches[i] = 0;
    }
}
```c
int ADCValues(int Channel)
{
    #ifdef PREFILTER
        static int i, match;
    #endif

    static int sample;
    sample = ADC[Channel];

    #ifdef PREFILTER
        match = 0;
        while(((sample - Buffer [0][pos[Channel]][Channel]) > (GlitchHyst)) ||
            ((Buffer [0][pos[Channel]][Channel] - sample) > (GlitchHyst)))
        {
            pos[Channel]++;  
            if(pos[Channel] >= GlitchBuffer)  
                pos[Channel] = 0;
            match++;
            if(match >= GlitchBuffer)
                break;
        }

        if(match >= GlitchBuffer)  
            // Possible glitch detected
            {
                glitches[Channel]++;
                if(glitches[Channel] > GlitchBuffer)  
                    // Previous samples maybe were not glitches
                    {
                        for(i=0; i<GlitchBuffer; i++)
                            Buffer [0][i][Channel] = Buffer [1][i][Channel];
                        glitches[Channel] = 0;
                    }
                else  
                    // Store suspicious samples
                    Buffer [1][glitches[Channel]-1][Channel] = sample;
                    sample = 0;
                    for(i=0; i<GlitchBuffer; i++)
                        sample += Buffer [0][i][Channel];
                    sample /= GlitchBuffer;
            }
    else
        {
            glitches[Channel] = 0;
            pos[Channel]--;  
            if(pos[Channel] < 0)
```
pos[Channel] = GlitchBuffer - 1;
Buffer [0][pos[Channel]][Channel] = sample;

#endif

    return(sample);

}