# Design and Implementation of an IoT-controlled DC-DC Converter

Voravit Tanyingyong, Robert Olsson, Markus Hidell, Peter Sjödin, Björn Pehrson
School of Information and Communication Technology
KTH Royal Institute of Technology
Kista, Sweden
Email: {voravit, roolss, mahidell, psj, bpehrson}@kth.se

***Abstract*—In line with the shift towards renewable energy, small-scale solar panels have become commonly available. Solar panels are intermittent energy sources producing direct current, and DC-DC converters are needed to convert between different voltage levels, both for different power loads and for storing energy. DC-DC converters typically have a very limited functionality and are statically configured for specific voltage levels. In this paper, we propose a new generation of flexible DC-DC converters with software and communication support (through Contiki and CoAP) for remote power monitoring and control. We present a prototype design and implementation of a DC-DC converter including a microprocessor, a lean operating system, and networking support. With such a DC-DC converter, controlled over the Internet, we can address various types of power and energy related issues and advance the state-of-the-art of green networked applications.**

## I. Introduction

Renewable energy and flexible utilization of energy have gained much interest during the latest years. In line with the shift towards renewable energy, numerous small-scale solar panels emerge as affordable alternative energy solutions, many of which can be easily found in local hardware stores. Since solar energy is an intermittent energy source available during daytime only, the generated energy is usually stored in energy storage devices, such as batteries.

Battery cells are traditionally based on chemical processes. Recent development of supercapacitors provides alternatives, which offer both strengths and weaknesses when compared to the chemical cells. On the strength side, they can stand an almost unlimited number of recharging cycles and consequently have a longer lifetime. They have very low internal resistances allowing them to have much shorter charge/discharge cycles, which is also a challenge to be considered when designing systems. They also contain fewer substances harmful to the environment. On the weakness side, they have lower energy density. Large-capacity supercapacitor batteries become bulky and also more expensive than traditional alternatives. They are, however, already competitive in low-power systems.

In a scenario with different kinds of energy storage devices and various types of loads, DC-DC converters are needed to convert DC power to the correct voltage levels for devices consuming the stored energy. In such a setting, it is also desirable to be able to monitor power consumption and to control and direct the power supply. However, traditional DC-DC converters are statically configured and lack the ability to be remotely accessed for monitoring and control. This calls for a new generation of DC-DC converters that are flexible in terms of supporting multiple types of energy sources and power loads, and that have support for communication – networked DC-DC converters.

In earlier work of ours, we have designed a flexible DC-DC converter that is able to transform varying input DC voltage into a programmable steady output DC voltage as described in [1]. The design is based on a bang-bang controller [2] for its simplicity and independence of load variations. Although the design focus was for supercapacitors, it can also be used as a generic platform for DC voltage conversion. We have implemented this in the form of a prototype printed circuit board (PCB).

In this work, we propose a new generation of DC-DC converters for power monitoring and control. We use the prototype board mentioned above as our hardware platform and introduce programmable control and monitoring capabilities as well as support for communication. This is realized by adding an operating system and networking support to the DC-DC converter. For the operating system, we use Contiki, "an open source operating system for the *Internet of Things* (IoT)" [3], to provide an extensible platform suitable for further development. For networking support, we use the constrained application protocol (CoAP) [4] as the main communication protocol since it is specifically designed for devices with constrained resources. We use IPv6 as the underlying network protocol since it is well suited thanks to the stateless address autoconfiguration and large address space needed for future expansion according to the vision of IoT.

The prototype board with integrated software components according to the above can be programmatically controlled over the Internet. Various parameters on the board can be monitored and configured remotely. This opens up possibilities for various types of novel applications. For instance, this DC-DC converter can be used as a component in a smart DC grid to monitor and control power consumption in real-time. The collected information can be used as an input to an optimization process for efficient power distribution and energy storage.

978-3-901882-56-2 

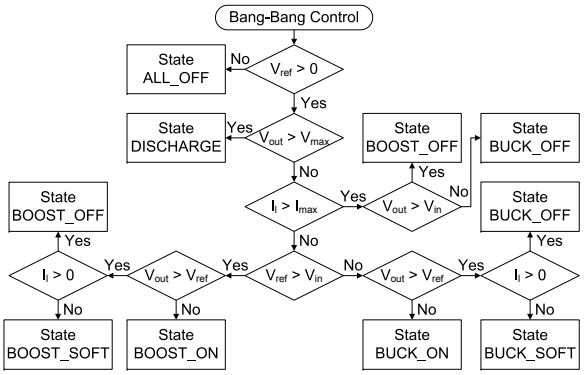


Fig. 1. The Bang-Bang Control Algorithm

TABLE I
Converter States with the corresponding N-Channel (NMOS) and P-Channel (PMOS) switches state

| State | NMOS1 | PMOS1 | NMOS2 | PMOS2 |
|---|---|---|---|---|
| BUCK_OFF | On | Off | Off | On |
| BUCK_SOFT | Off | Off | Off | On |
| BUCK_ON | Off | On | Off | On |
| BOOST_OFF | Off | On | Off | On |
| BOOST_SOFT | Off | On | Off | Off |
| BOOST_ON | Off | On | On | Off |
| DISCHARGE | On | Off | On | Off |
| ALL_OFF | Off | Off | Off | Off |

## II. The bidirectional DC-DC Buck/Boost Converter Core Design for Supercapacitors

To master the challenges that the output voltage of supercapacitor cells is not constant and that the internal resistance is very low, we designed a system that is able to transform variable DC voltage into a steady DC voltage and to control the currents involved.

### A. Bang-Bang Control Algorithm

The control software on the prototype board runs as an infinite loop to continuously monitor several parameters of the converter and switches its state according to the bang-bang control algorithm depicted in Fig. 1.

The values of input voltage ($V_{in}$), output voltage ($V_{out}$), and inductor current ($I_l$) are obtained from the reading sensors while the values of desired output voltage ($V_{ref}$), maximum output voltage allowed ($V_{max}$), and maximum inductor current allowed ($I_{max}$) are configured by the user. These values are used to decide the converter states, which are a combination of on/off states of N-channel and P-channel MOSFETs as shown in Table I.

### B. Converter Core Hardware Design

The converter core is shown in Fig. 2. It is designed as a symmetrical circuit consisting of a combination of buck and boost converters, which can step down and step up the output voltage respectively. Two pairs of NMOS and PMOS are used to provide synchronous switching and ability to work in bidirectional. The MOSFETs are controlled by a microcontroller unit (MCU).

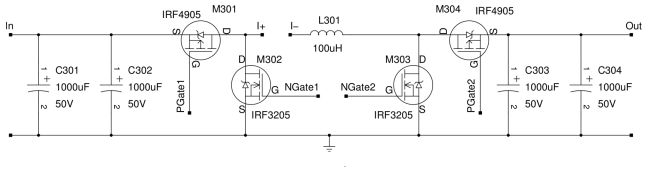


Fig. 2. The Converter Core Design

In order to apply appropriate actions through the MOSFETs, various parameters on the converter are monitored. For instance, our bang-bang controller requires three parameters ($V_{in}$, $V_{out}$, and $I_l$) to be monitored as described in Section II-A. The reading sensors (current sensors and voltage dividers) are connected to the analog-to-digital converters (ADCs) on the MCU, in which the control algorithm is being implemented. More details on the design can be found in [1] (Chapter 3).

## III. Architecture for Prototype Board

Our prototype of the bidirectional DC-DC buck/boost converter can transform a varying input DC voltage from the supercapacitors to a steady output DC voltage in a range of 0-25 V with a maximum input and output current of 6 A. Although the design focus was for supercapacitors, it can also be used as a generic platform for voltage and energy conversion. We propose software extensions with an operating system and an Ethernet-based communication architecture to facilitate remote power monitoring and control for next generation DC-DC converters.

### A. Communication Architecture

*1) Link Level – Ethernet:* Ethernet is the most dominant link layer technology. Many MCUs have a built-in Ethernet controller making it relatively easy to adopt.

*2) Network Level – IPv6:* IP is a natural choice since it is widely supported. We choose IPv6 since it offers a large address space suitable for expansion. It also has stateless address autoconfiguration, which simplifies address assignment. More information about IPv6 and its usage in low-power devices can be found in [5].

*3) Application/Transport Level – CoAP/UDP:* We use the current IETF Internet-Draft constrained application protocol (CoAP) with UDP transport. CoAP is a specialized protocol designed for use with constrained nodes and constrained networks [4]. It targets small electronic devices, which have a simple microcontroller and limited memory resources. They are typically operated in a network environment that has high packet error rates (often referred to as *lossy network*). It has low overhead and is optimized for machine-to-machine (M2M) applications.

### B. Hardware Architecture

The prototype board of the DC-DC converter includes an LPC1768 microcontroller. This microcontroller is an ARM Cortex-M3 based microcontroller for embedded applications featuring a high level of integration with low power consumption [6]. It has an Ethernet controller,

which could be used to connect the board to any IP network. More details on the design, development, and evaluation of the prototype board (before an operating system and networking support are introduced) can be found in [1].

## IV. Software Design and Implementation

The main focus of this work is to port an operating system and bring network communication to the prototype board. An operating system provides an extensible platform that facilitates introduction of new capabilities and applications, while networking support gives the DC-DC converter an ability to communicate, which makes remote control and management possible.

To provide a high degree of flexibility, Contiki is used as an operating system on the prototype board. Contiki is an open source operating system for the IoT that allows tiny, low-power systems to communicate with the Internet [3]. It supports uIP, a small TCP/IP stack, that can be used on 8- and 16-bit microcontrollers. Contiki's development platform provides an abstraction from the low-level microcontroller-specific functions making it easier to work with.

We implement most parts of the software components required in our communication architecture on the prototype board. Nevertheless, there are some remaining challenges to be addressed. We highlight the key achievements as well as challenges in our software development process as follows.

*1) Porting the LPC1768 microcontroller library to Contiki:* Contiki supports various MCUs and hardware platforms. However, the LPC1768 microcontroller on the prototype board is not supported by Contiki, so we have developed support for this microcontroller in Contiki.

The ARM Cortex Microcontroller Software Interface Standard (CMSIS) driver library for the LPC1768 microcontroller is available on the MCU manufacturer (NXP) website [7]. We use this library as a basis for creating a set of files to interface between the library and Contiki. As a result, we successfully port the LPC1768 microcontroller to Contiki. Unfortunately, there is a copyright license conflict between ARM and Contiki, which prevents the code to be included in the Contiki main distribution. We are in contact with Contiki's maintainer and NXP to try to resolve this issue. Our fork of the Contiki distribution can be found in [8].

*2) Implementing the DC-DC Converter Control Function on Contiki:* Once Contiki can be run on the prototype board, it is straight-forward to implement the control function since both the existing control software source code (available in Appendix E of [1]) and Contiki are written in C.

*3) Defining Parameters and Messages in CoAP Context:* A CoAP message consists of a small header, resource identifiers, and an optional payload (which typically contains values of the requested resources). We use CoAP to control and collect data about the current status of the DC-DC converter. Thus, we need to define resources (in terms of parameters) and messages in CoAP according to our usage.

TABLE II
Control Parameters and the State Vector

| **Control Parmeters** (Identifier: dc-dc/controlParameters) | |
|---|---|
| **Parameter** | **Description** |
| $userAllowed$ | Switch on/off for output voltage |
| $V_{ref}$ | Desired output voltage |
| $V_{max}$ | Maximum output voltage allowed |
| $I_{max}$ | Maximum inductor current allowed |
| **State Vector** (Identifier: dc-dc/stateVector) | |
| **Parameter** | **Description** |
| $V_{in}$ | Measured input voltage |
| $V_{out}$ | Measured output voltage |
| $I_{in}$ | Measured input current |
| $I_{out}$ | Measured output current |
| $state$ | Current state of the bang-bang controller |

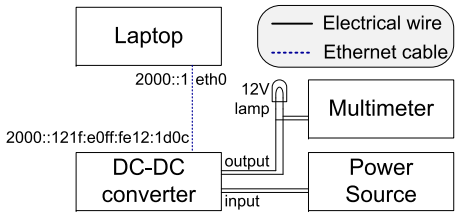


Fig. 3. Experiment Setup

Our CoAP implementation is based on the implementation from Erbium for Contiki [9]. We classify parameters used on our prototype board into two types; *control parameters*, which are writeable parameters that can be configured by the user, and *state vector*, which is a list of read-only parameters. Each parameter type has its own resource identifier. All parameters in our current implementation are listed in Table II. GET and POST methods are used to read and write these parameters respectively. For the message exchange, confirmable message is used in the request (for both GET and POST methods). This means that the requesting part always gets an acknowledgement message back. The prototype board also sends periodic status reports to the monitoring server (through a hard-coded IP address). We plan to replace this with the *Observe Option* extension [10] in the future.

## V. Evaluation of the Implementation

To verify our software implementation, we set up a simple experiment as shown in Fig. 3. The input and output of the prototype board (illustrated as *DC-DC converter* in Fig. 3) are connected to a power source and a load (12V lamp) respectively. An Ethernet port on the prototype board is connected to a laptop via an Ethernet cable. The output voltage is measured using a digital multimeter (UT-71B [11]).

We develop a CoAP client to communicate with the prototype board. It can also run as a background process to monitor all CoAP messages. We run this CoAP client on a laptop, which is used for monitoring and control. The source code of our CoAP client can be found in [12]. The

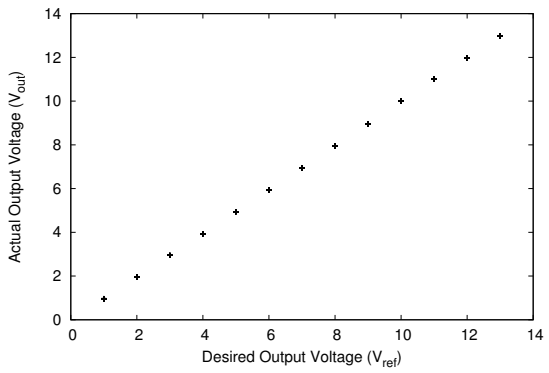


Fig. 4. Actual Output Voltage (with 10V Input)

current implementation supports sending a request with GET and POST methods through a simple command line interface. Alternatively, other CoAP implementations such as libcoap [13] and Copper (Cu) [14] can also be used for this purpose.

We use 10V input voltage on the power source. We enable the prototype board's output by setting *userAllowed* to "yes". We also set $I_{max}$ and $V_{max}$ to fixed values of 2A and 20V respectively. For the experiment, we simply set the value of $V_{ref}$ and measure the actual output voltage with the digital multimeter. We also vary $V_{ref}$ from 1V to 13V (The prototype board supports up to 25V output voltage). For each value of $V_{ref}$, we take 100 samples of the output voltage as our measurements.

We plot all samples in a graph as depicted in Fig. 4. For each output voltage measured, the variation in the samples is relatively low and hardly visible in Fig. 4. Thus, we conclude that the prototype board can correctly step up and step down the output voltage. However, in our experiments we have observed occasional significant deviations from the expected output voltage if we continuously generate CoAP requests while carrying out the experiment (we tested with 20 requests per second). This is likely to be a result of Contiki operating in a single thread context. When many CoAP messages must be processed, the bang-bang controller does not run frequently enough to maintain the steady output. This calls for a mechanism to give a higher priority to the bang-bang controller than to the rest of the processes to ensure stability of the converter. Further investigations and experiments are needed to address this issue.

## VI. Discussion

We successfully port Contiki to our prototype board and get the communication in place. Nevertheless, integrating our codes to Contiki main distribution is still a challenge remains to be solved. Contiki brings in flexibility and programmability to our prototype board. However, security is still a challenge for constrained systems in a constrained environment.

From the communication aspect, the fundamental messages needed for control and data collection purposes are developed as a proof-of-concept. Although CoAP supports built-in resource discovery, we need to extend our application further to increase usability. For example, we plan to provide a list of available DC-DC converters in the networks and their resources, give statistical information of the collected data, etc. In addition, further improvement on machine-to-machine communication between the DC-DC converters and the monitoring server would be very beneficial.

The network-enabled DC-DC converter has good potentials for various applications. One application is to use it as a component of a power grid. DC transmission is efficient and suitable for small-scale DC power grids. To control and direct the power, networked DC-DC converters are required. Nevertheless, our current DC-DC converter hardware design is still quite complex and has some limitations as described in [1]. Further improvement to resolve these issues would increase the usability and ease the adoption of the DC-DC converter.

## VII. Conclusion

Given advancements in today's technologies, it is possible to build a smart DC-DC converter that can be monitored and controlled via an IP network. To the best of our knowledge, this is the first DC-DC converter with CoAP support being implemented. We believe that such kind of devices will play a vital role in emerging DC power transmission networks. Communications will be the key to facilitate innovations and improvements in power distribution networks.

## References

[1] J. Querol Borròs, "MCU Controlled DC-DC Buck/Boost Converter for Supercapacitors," Master's thesis, KTH Royal Institute of Technology, 2012, http://kth.diva-portal.org/smash/get/diva2:546759/FULLTEXT01.pdf.

[2] D. N. Burghes and A. Graham, *Introduction to control theory, including optimal control*. Ellis Horwood Ltd, Dec. 1980.

[3] The Contiki project, "Contiki: The Operating System for the Internet of Things," http://www.contiki-os.org/index.html.

[4] Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)," IETF, Internet-Draft, Jun. 2013.

[5] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley, Dec. 2009.

[6] NXP Semiconductors, "LPC1769/68/67/66/65/64 - Product data sheet," Aug. 2012.

[7] ——, "LPC175x_6x CMSIS Firmware Driver Library," Apr. 2012, http://www.lpcware.com/system/files/lpc175x_6x_cmsis_driver_library_0.zip.

[8] CSD Fall 2012, Microgrid team, "Contiki fork for DC-DC converter," Dec. 2012, https://github.com/noiseoverip/contiki.git.

[9] Matthias Kovatsch, "Erbium (Er) REST Engine and CoAP Implementation for Contiki," http://people.inf.ethz.ch/mkovatsc/erbium.php.

[10] Z. Shelby, K. Hartke, and C. Bormann, "Observing Resources in CoAP," IETF, Internet-Draft, Jul. 2013.

[11] Uni-Trend Group Limited, "UT71B – Intelligent Digital Multimeters," http://www.uni-trend.com/UT71b.html.

[12] Robert Olsson, "IoT-grid," May 2013, https://github.com/herjulf/IoT-grid/tree/master/iotgrid.

[13] Olaf Bergmann, "libcoap: C-Implementation of CoAP," http://sourceforge.net/projects/libcoap/.

[14] Matthias Kovatsch, "Copper (Cu):: Add-ons for Firefox," https://addons.mozilla.org/en-US/firefox/addon/copper-270430/.