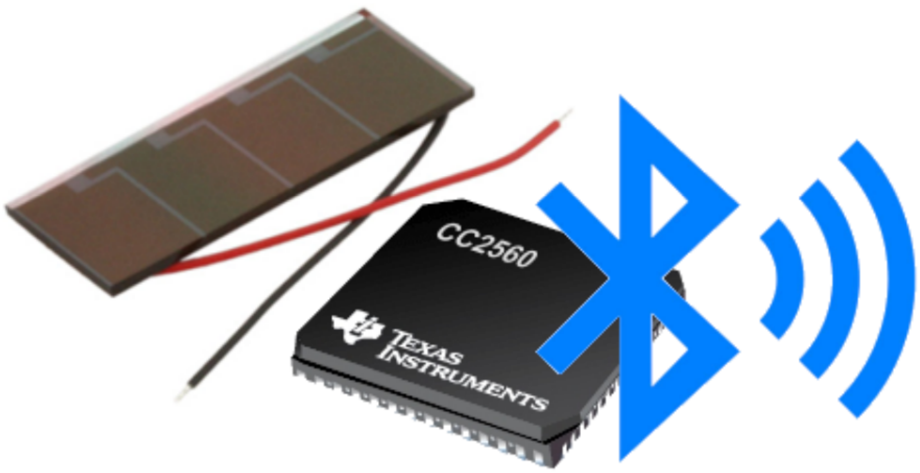
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik und Kommunikationsnetze

Carlo Signer

Batteryless Bluetooth Low Energy Communication



Bachelor Thesis (BA)-2017-07
February 2017 to June 2017

Supervisors: Lukas Sigrist sigristl@tik.ee.ethz.ch
Andres Gomez gomeza@tik.ee.ethz.ch
Professor: Prof. Dr. Lothar Thiele thiele@tik.ee.ethz.ch

Abstract

Typical harvesting applications are coupled with big energy storage devices to provide the system with power whenever needed. Batteryless energy harvesting systems have the advantage to work in long term scenarios without maintenance. This thesis explores the feasibility of wireless data communication in so called transient systems using the Bluetooth Low Energy (BLE) protocol. The data packets are sent via a connectionless channel which is often used in low power applications with beacons. Since a transient system totally relies on its energy consumption, this work evaluates the energy needed to send BLE packets. The evaluation shows that transmitting different Payloads has a linear behavior. The minimal energy to send a packet including a payload of 1 byte with a TX-Power of 5 dBm is about $13.3\,\mu J$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traditional energy harvesting application make use of batteries or supercapacitors to store high amounts of energy to provide the system whenever energy is demanded and can support the load even in times of energy unavailability. However, big storage devices are expensive in terms of area, have high discharge rates and are not easily integrated on board [2]. More recently, a different research line has evolved with its focus on batteryless sensor nodes. Without a battery to store energy for future use, a so called transient system is extremely dependent on its environment and the task execution is highly sporadic. There are different solutions to a transient system. In many of these the size is not a limiting factor, so the energy source can be relatively big. But in harvesting applications which have to be wearable, size becomes a very important factor [3]. The increasing number of devices in the Internet of Things (IoT) demands applications in a efficient, long term, low-cost, self sustainable fashion. The Energy Management Unit (EMU) presented in [4] can buffer energy even if the input power is much smaller than the output power needed to supply the load. The EMU allows the device to operate efficiently in a wide input power range without big and expensive storage devices (see Figure 1.1).

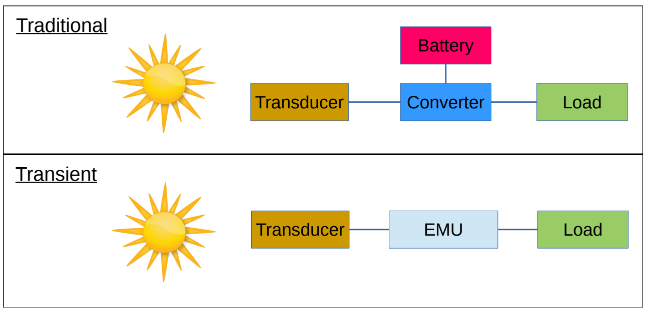


Figure 1.1: Traditional energy harvesting application with a battery to store energy for future use versus a transient system using an EMU

Batteryless harvesting scenarios have the advantage of being able to perform for a significantly long time in unaccessible locations where changing the battery is impractical. In many of these applications with sensor nodes, the gained sensor-information is not useful on board and needs to be transmitted to an external receiver. On request of the transient system to be power efficient, the wireless data communication shall consume as less power as possible. An obvious choice is the Bluetooth Low Energy (BLE) protocol. Compared to classic Bluetooth, it is designed to have a reduced power consumption while maintaining a similar communication range. BLE can operate in different modes. In applications with several cheap, simple sensor nodes or with beacons [5] it is preferable to have a connectionless communication. The user with the receiver can collect data from every node and is able to stay anonymous. BLE offers a connectionless communication where a receiver can act as a sniffer.

The focus of this thesis lies in exploring the feasibility of wireless data communication in energy harvesting applications with customary goods, using the Bluetooth Low Energy (BLE) protocol. The BLE mode used, is a connectionless way of communication which does not use multi hop. The sensor data sent is small enough that it fits completely into a packet in this mode, so the thesis investigates the communication of single packets sent on a single channel.
The thesis presents an embedded application that senses the temperature and humidity of the ambient air and transmits the read sensor data via BLE. The energy consumption of the node is measured and especially the part of transmission is analyzed. Finally, the gained information about energy consumption is used to determine the relevant parameters for the EMU and a real world energy harvesting application is presented.

1.1 Motivation

Many applications demand the ability of wireless data communication. Especially since the Internet of Things (IoT) has gained a lot of attention, and devices have to be portable, a wireless communication becomes very important. The implementation of a wireless data communication brings additional challenges to the system design of a batteryless energy harvesting application:

1. Higher power consumption: Instead of storing the sensing data, the information has to be sent out to a receiver, which comes with a higher power consumption for powering the Radio-core and providing energy for sending data-packets.
2. Initialization: There is additional power consumption with configuring the payload for sending out packets. Also the Radio always has to be powered up and shut down when it is not needed anymore.

There is a big variety of different communications but not all of them are very power efficient and suitable. Bluetooth Low Energy is constructed especially for devices which have to be very power efficient so it is an reasonable choice. Also, BLE is very common and it is easy to find a receiver (e.g. smartphone).
The goal of this thesis is to explore the feasibility of wireless data communication in batteryless energy harvesting applications, using the Bluetooth Low Energy protocol.

1.2 Contributions

To ascertain the use of BLE in transient systems, this thesis makes the following contributions:

1. Build an environmental sensor node: In a first step a simple wireless sensor node is built and configured, to read out the temperature and humidity of its environment and send the gained information to a BLE-sniffer.
2. Analyze BLE communication: BLE trade-offs of different communication parameters are used to characterize the use of data transmission via BLE.
3. Determine EMU parameters: The sensor node, built in step 1., will be used to analyze the different states of the system in terms of energy to calculate the required EMU parameters.
4. Finally the gained information is used to connect the sensor node to an EMU, connected to a solar cell, and a real world application is presented.

1.3 Related Work

The authors in [6] take a deeper look into powering the IoT with storage-less and converter-less applications unlike traditional harvesting scenarios, where the load and the transducer are directly coupled. But in scenarios where the load and the energy source have incompatible operating points a converter becomes necessary.
The work done in [4] shows a way to work with batteryless transient sensor nodes. The Energy Management Unit (EMU) is presented to harvest energy and minimize the storage capacitance. With Feedback-based Dynamic Energy Burst Scaling technique the EMU is able to track the load's optimal power point to optimize the harvesting process.
In [3], the authors concentrate on a wearable transient system to determine the user's walking speed. There is a high request to have a preferably small device. To fulfill this request, the EMU is used to harvest energy and the data is stored locally on an SD card.

Existing work in [7] and [8] describe the Bluetooth and BLE protocols and show the differences between traditional scenarios and the advantage of using BLE in low power applications. In general, BLE is just a specification of Bluetooth to make the communication less power consuming. In [5], the author shows ways for indoor localization by placing beacons using Bluetooth. In this scenario the receiver has to gain information from different devices to localize itself, using similar communication as in this thesis.

Chapter 2

Preliminaries

2.1 Transient System using the EMU

Instead of using a battery, in a transient system (built using the presented work in [4]) the Load is connected to the Energy Management Unit (EMU), in order to harvest power. The energy is stored in a small capacitor, which is loaded up to a certain level, to then supply a single energy burst for a task execution. Since the EMU triggers the load when there is enough energy stored , the task execution is highly sporadic. Although, for sensor nodes with low duty cycle, which are in sleep mode for a longer time before executing a task, this is absolutely suitable. The experience has shown, that transiently powered systems can still perform computationally expensive tasks. In such applications the capacitance can be reduced to the minimal size providing energy for one single burst, to reduce energy-expensive charge cycles. Typically, the EMU operatse with input powers that are significantly lower than the power needed by the load, which allows the development of small-sized, low-cost energy harvesting nodes, that can run without upkeeping services.

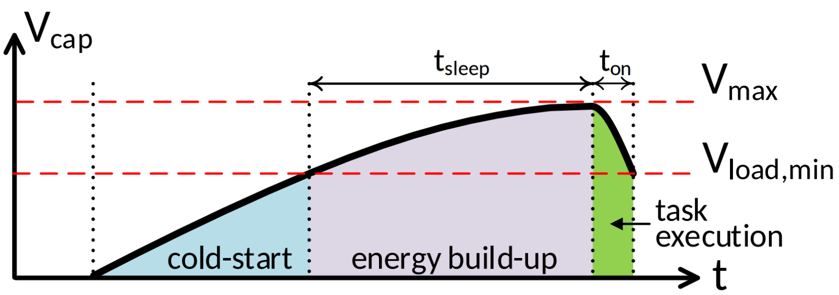


Figure 2.1: Loading sequence of the capacitance of an EMU Gomez et al. [1]

Figure 2.1 shows the voltage in the capacitance of the EMU. It starts with zero and when it reaches V_{max} the load is woken up (triggered) and a task is executed. The capacitor is dimensioned so that the voltage stored in it never falls under the limit $V_{load,min}$. If the voltage falls beneath this level during a task execution, a proper completion of the task is not guaranteed. After the load was on, it goes back to sleep so the EMU can again build-up energy to supply an energy burst.

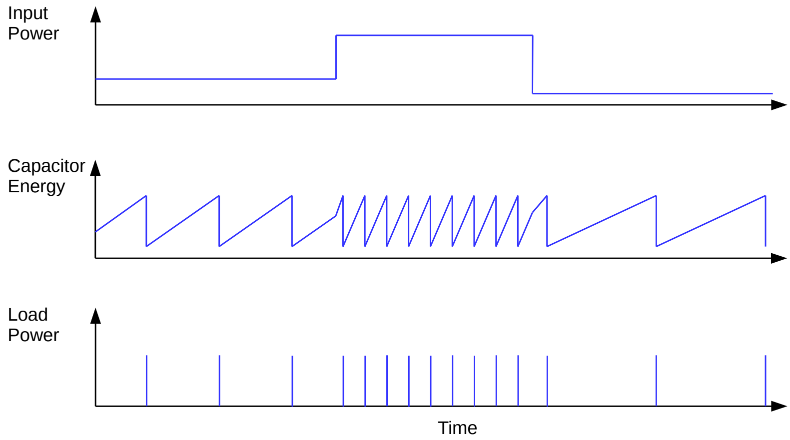


Figure 2.2: Environmental dependency of a transient system

A transient system is extremely dependent on its environment. For example, if the energy source is a solar cell. In the night (without artificial light) the system can not harvest energy and is not able execute tasks. At day, the higher the input Power gets, the faster the energy stored in the capacitor can be built up and the higher the resulting frequency of the task execution gets (see Figure 2.2).

2.1.1 Architecture of the Energy Management Unit

With the integrated Maximum Power Point Tracking (MPPT), the boost converter can adjust the input impedance such that the source always operates at its optimal power point to maximize the harvested energy. The EMU uses an Feedback-based load tracking algorithm called Dynamic Energy Burst Scaling (DEBS) that adjusts to the lowest operating voltage to minimize the load's energy per task.

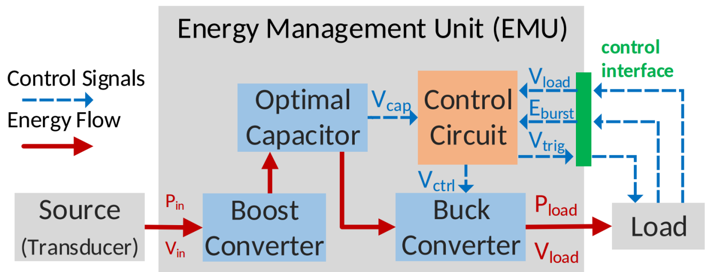


Figure 2.3: Architecture of a transient system using the the Energy Management Unit by Gomez et al. [1]

Chapter 3

BLE Sensing Node

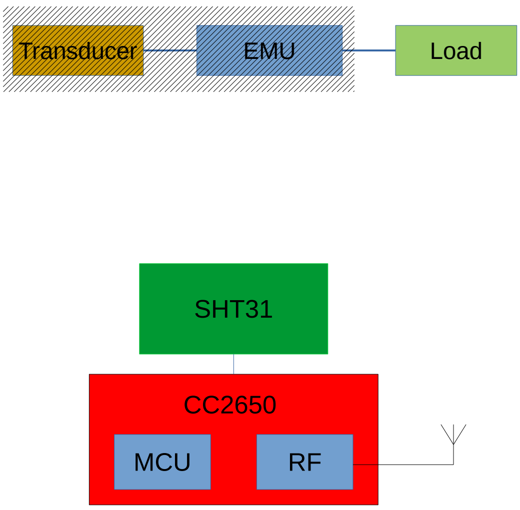


Figure 3.1: BLE sensing node Block diagram

Figure 3.1 shows a abstract Block diagram of the BLE sensing node used in this thesis. Chapter 3 focuses on the different parts and describes the essential of this sensor node.

3.1 CC2650 Launchpad

The CC2650 Launchpad[1] brings easy Bluetooth Low Energy handling and supports ultra-low power management. The main parts used in this thesis are the MCU that contains a 32-bit ARM®Cortex®-M3 processor, running at 48 MHz as the main microcontroller and a rich peripheral feature set that includes a unique ultra-low power sensor controller. The second important part of this board is the RF-core. As a operating system for the whole work Contiki Operating System was used, which provides a BLE-stack and makes it very easy to make use of the BLE protocol.

[1] http://www.ti.com/tool/launchxl-cc2650

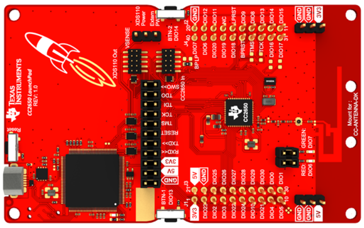


Figure 3.2: CC2650 Launchpad[1]

The CC2650 has different power modes to achieve a low-power application. The modes are listed and described in the table below:

Mode	Active	Idle	Standby	Shutdown
System CPU	Active	Off	Off	Off
RF-Core (radio)	Available	Available	Off	Off
Current	Application dependent	Application dependent	$1\,\mu A$	$0.1\,\mu A$
Wake up on edge pin	Available	Available	Available	Available
Wake up on reset pin	Available	Available	Available	Available

Table 3.1: CC2650: Power Modes

The EMU triggers (wakes up) the board with an edge pin so every sleep mode can be used. Since the Standby mode has the lowest current, this is the mode which will be used in this thesis. Although, waking up from standby (booting) consumes more energy than from other modes. After waking up the CC2650 will be in Active mode to be able to do sensor readings and use the radio to transmit BLE packets.

3.2 SHT31

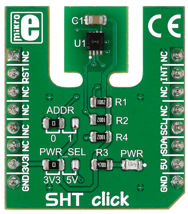


Figure 3.3: SHT Click with the SHT31 sensor on board[2]

Directly connected to the Board, the SHT31[2] measures the ambient temperature and humidity. It comes with a wide supply voltage range, from 2.4 V to 5.5 V. The sensor has a I2C Interface with communication speeds up to 1 MHz and a fast start-up and measurement time. The maximal Power-up time is 1 ms and the maximal measurement duration amounts to 4 ms in the mode used in this thesis. Whenever the sensor is powered up but not performing any measurement, it automatically enters idle state for energy saving where it consumes about up to $2\,\mu A$.

[1] http://www.ti.com/tool/launchxl-cc2650
[2] https://shop.mikroe.com/sht-click

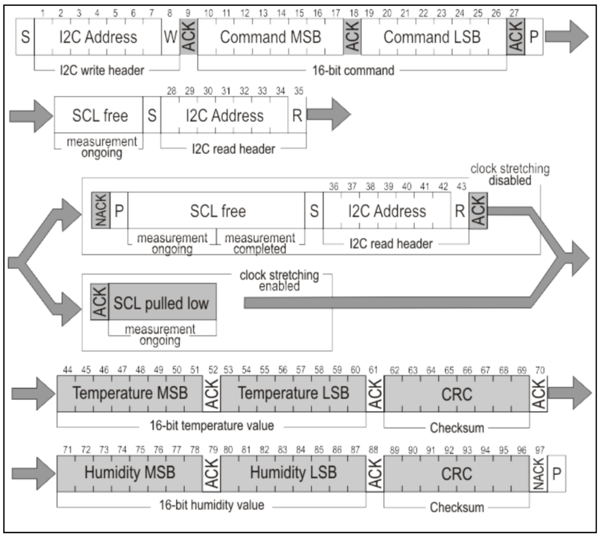


Figure 3.4: Single Shot Data Acquisition Mode of for the SHT31 sensor[3]

The SHT31 can either operate in Single Shot (single measurement) mode or in Periodic mode (periodic measuring). Since the task execution in a transient system is sporadic, Single Shot mode was used in this thesis. Figure 3.4 shows the way measurements are done using the Single Shot mode. The master (CC2650) first addresses the sensor over the I2C line and gives it a 16-bit command. The sensor will acknowledge (ACK) the reception and send two bytes (temperature) followed by a CRC checksum and another two bytes (humidity) followed by a CRC checksum. If no data is available, the sensor responds to a read header with a not acknowledge (NACK). To prevent this from happening, a method called Clock Stretching can be initialized in the command. When a command with Clock Stretching has been issued, the SCL line is pulled low until the sensor has its data collected. The sensor then releases the SCL line and sends the measurement results.

3.3 Bluetooth Low Energy Broadcasting

Bluetooth [7] is a universal radio interface, that has been developed for devices to communicate wirelessly via short-range ad hoc radio connections. In contrast to other ad hoc systems, Bluetooth is based on peer communications, where there is no difference between the radio units and it has no base station or terminal. The radio spectrum is placed in the Industrial, Scientific, Medical (ISM) band and has a range form 2402 to 2485 MHz. It contains 79 channels and the spacing between them is 1 MHz. The data rate is about 1 Mbit/s. In a traditional Bluetooth scenario, two devices are connected and communicate in a stateful mode.

	Traditional	BLE
Data Rate	1 Mbit/s	1 Mbit/s
Channels	79	40
Spacing	1 MHz	2 MHz

Table 3.2: Traditional Bluetooth versus BLE

In this work a slightly different protocol was used (see Table 3.2). Compared to the traditional Bluetooth protocol, Bluetooth Low Energy [8] only has 40 Channels each separated by 2 MHz.

[3]https://download.mikroe.com/documents/datasheets/Sensirion_Humidity_SHT3x_Datasheet_digital.pdf

There are two ways of communicating via BLE. In both, there is a transmitting device (peripheral) and one or more receiving devices. The first way is to use a connection. Here the slave sends out advertisements to tell that it wants to connect. The master can react on this and they can build a connection (e.g. connecting the music player to an audio box). In this stateful scenario all the information which is received properly, is confirmed. That means the receiving device sends a packet to the peripheral to tell that it can send the next packet. This connection makes sure that all the information is received properly. If not the packet can be sent again.
The second way is to use advertising channels as data channels. When it is not too important that 100% of the information is received, this unidirectional communication can satisfy the tasks. In this stateless scenario only the peripheral is sending without knowing if a device is listening. However, it is less power consuming since the peripheral does not have to wait for confirmation from a receiver and no connection has to be built up.
In this project data is sent from a simple sensor node which measures the temperature of the ambient air. This node is at sleep for most of the time and wakes up just for a very short amount of time, in an event triggered fashion. Always building up a connection is time and power consuming. Sending the information in a unidirectional, connectionless broadcasting way is the better choice. And even if some packets are not received it does not cause problems because the temperature does not change too fast.

As we can see in Figure 3.5, channels 37,38 and 39 are reserved for BLE advertisements. Normally, a BLE peripheral transmits packets on all of these channels one after another. Due to energy saving and analytical reasons, in this thesis only channel 37 is used to send advertising packets

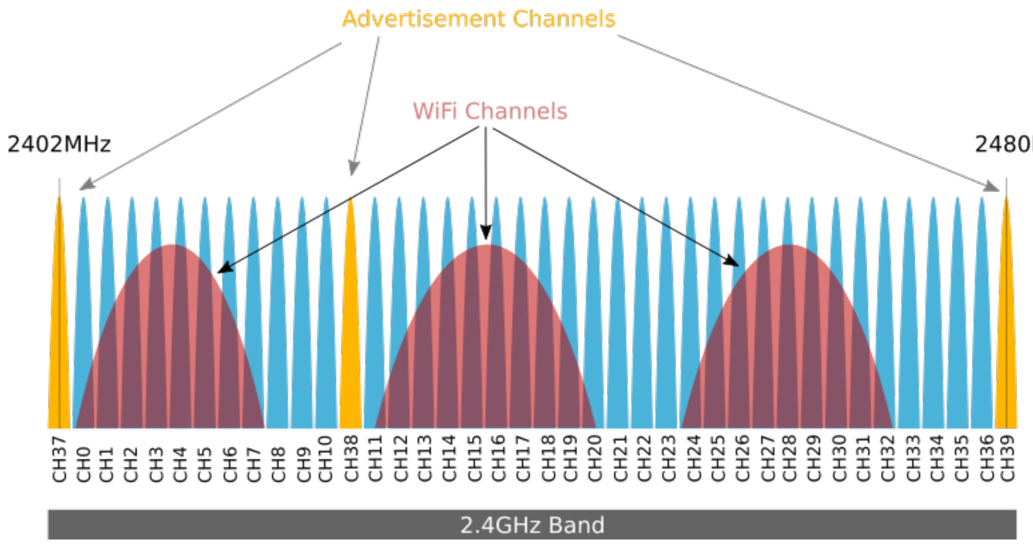


Figure 3.5: BLE advertising channels spectrum[4]

[4]http://www.argenox.com/a-ble-advertising-primer

3.3.1 BLE Advertising Packets

The Bluetooth Specification defines a top level advertising packet, that contains a preamble, an access address and as well a CRC Checksum.

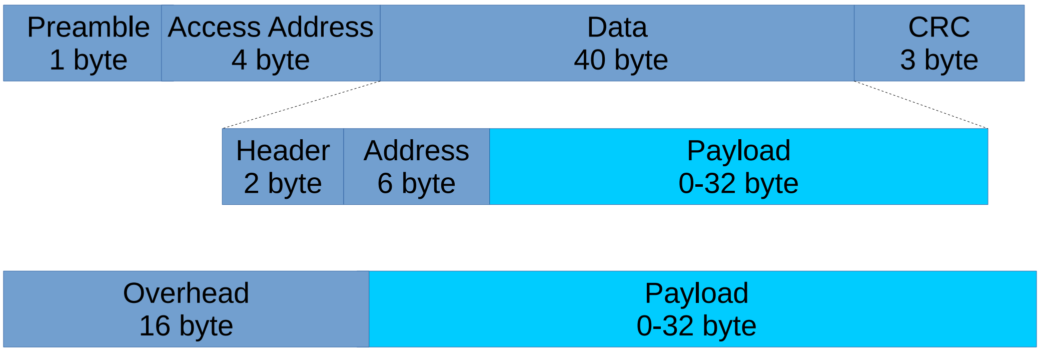


Figure 3.6: BLE advertising packet

The packet Data Unit (called Advertising Channel PDU) itself contains a 2 byte Header and a 6 byte address. The Advertising Channel PDU contains the length of the payload and also tells if the communication is connectible or not. All together there is a total overhead of 16 bytes and the payload can be filled up with 0-32 bytes.

Chapter 4

Evaluation

Since the BLE sensing node is directly connected to the EMU, the energy consumption of the task execution has to be determined. Figure 4.1 shows the cycle, which shall be executed in the end and integrated into a transiently powered system. In a next step, we are going to analyze the energy consumption of the different parts. First of all, the BLE communication is analyzed and especially the transmission part.

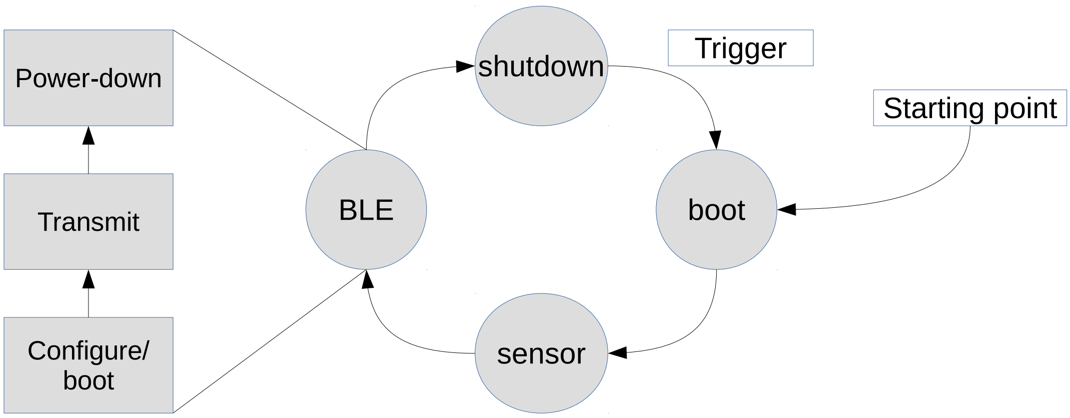
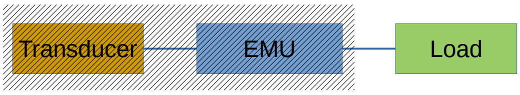


Figure 4.1: Cycle of the BLE sensing node in a transient system

4.1 BLE communication



In this section we take a closer look at the transmission part of the BLE communication and analyze the energy consumption of different trade-offs.

4.1.1 Experimental Set Up

For all the experiments, a constant power supply was used. Due to energy saving reasons, the provided voltage should be as small as possible. In this cycle the sensor is the limiting factor, which needs at least a supply of 2.4 V. Hence, all the measurements are done with a constant power supply of 2.4 V. A BLE-sniffer is used to receive the BLE packets.

For the measurements 2 things can be varied:

- The payload can be varied from 0 to 32 bytes with an increment of one.
- The power of the transmission is changed individually from -21 dBm to 5 dBm. 5 states are evaluated: -21 dBm, -12 dBm, -6 dBm, 0 dBm and 5 dBm

In this experiment the system first does 8 sensor measurements, which correspond to 8 temperature and 8 humidity measurements. This has a size of 32 byte and is exactly the maximal payload size sent. All these values are filled into an array and the board sends out different sizes of this array going from 0 to 32 bytes by an increment of 1 byte. After transmitting all sizes, the TX-Power is changed and the whole procedure repeats. Each individual measurement point has been taken a hundred times and averaged. For the measurements the RocketLogger [9] was used. It allows to sample with 64 kSPS. DC currents from $2\,mA$ to $500\,mA$ are measured with an accuracy of $0.03\% + 4\,nA$. DC currents from $1\,\mu A$ to $2mA$ have an accuracy of $0.09\% + 3\,\mu A$. Voltages from $6\,\mu V$ up to $5.5\,V$ can be measured with an accuracy of $0.02\% + 13\,\mu V$.

The Rocketlogger measures the voltage and current of the sensor node to calculate the power and finally the energy consumption. A configured GPIO (general purpose input/output) Pin is pulled high before the transmission and pulled low after a packet is sent in order to know which data points have to be averaged.

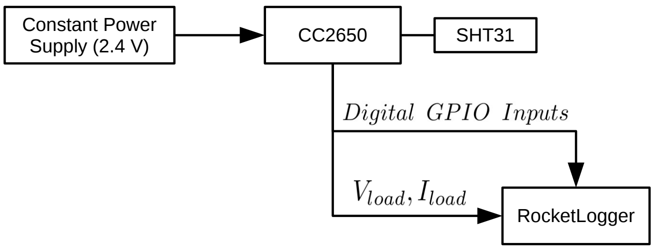


Figure 4.2: Experimental Set Up to evaluate BLE communication

4.1.2 Results

Figure 4.3 shows the energy per packet in where the payload in bytes is varied. As expected the behavior is linear which means that sending an additional byte takes the same amount of energy, independent of the packet size in total. Interesting are the points on the 0 byte payload line. These tell us how much energy is consumed for sending out the Overhead (see section 3.3.1) of the BLE advertising packets. This energy is the same for all measurement points of one TX-Power, because the Overhead always has the same size of 16 bytes. The second plot shows that the time is not dependent on the used TX-power.

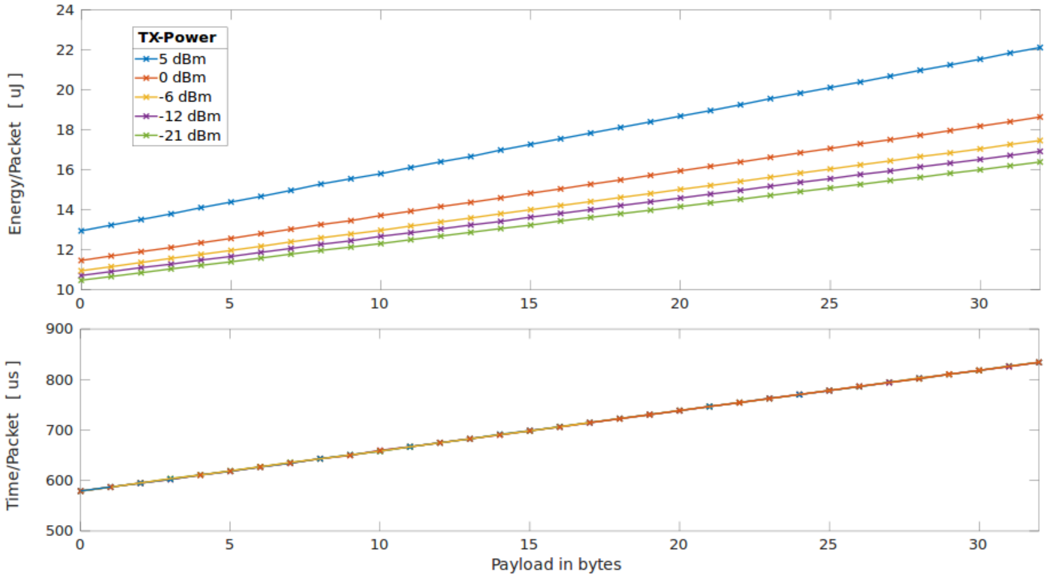


Figure 4.3: BLE communication measurement: Transmission part

If we have a closer look to the region shown in Figure 4.4, we can see that there is a region where we have some freedom of sending out different packages (black lines). From the lowest point of the 5 dBm line to the highest point of the -21 dBm line is a band where, with the same amount of energy, different decisions can be made about sending packets. For example, if an energy of $14\,\mu J$ (red line) is available, it could be used to send 3 bytes with a TX-power of 5 dBm or 17 bytes could be sent out with a TX-power of -12 dBm.

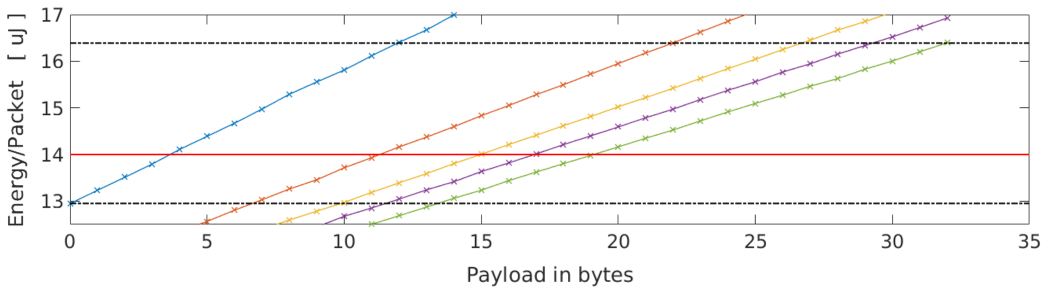


Figure 4.4: Bandwidth with some freedom of using available energy

In a next step the energy per packet has been normalized by the number of actual payload sent.

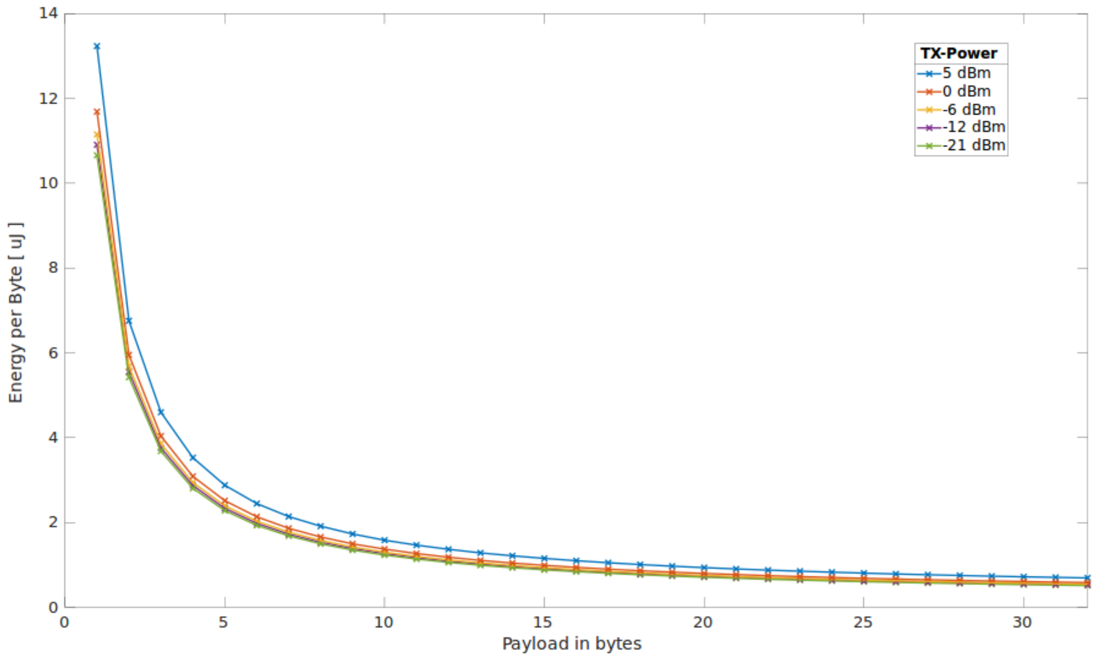


Figure 4.5: BLE communication measurement: Energy per Payload (normalized)

The plot shows the influence of the overhead in the packet. As the payload grows, the energy per byte falls and approaches asymptotically to some value, which corresponds to the slope of the lines shown in Figure 4.3. In this plot it makes no sense to show the points on the 0 payload line, because here the values go to infinity.

4.2 Real World application

To be able to integrate the system shown in Figure 4.1 all the energy consumption of the whole system. In this section all different parts and their energy consumptions are analyzed and in the end the sensor node is connected to a EMU and the system is powered by a solar cell.

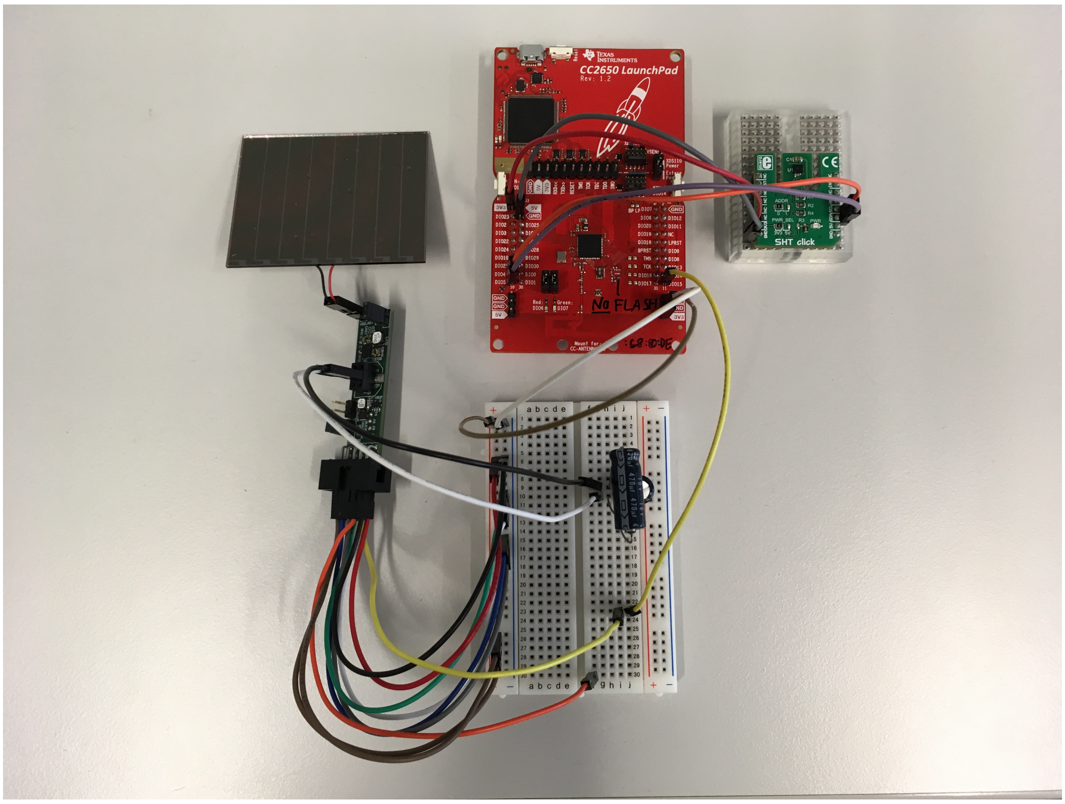


Figure 4.6: Picture of the transient system powered with a solar cell

4.2.1 Experimental Set Up

The sensor node sleeps while the capacitor is being loaded. When there is enough energy, the node is woken up, does a temperature/humidity measurement (4 bytes), sends the information and goes back to sleep (see Figure 4.1). Again, a BLE-sniffer is used to prove the transmission of the packets.
Four stages are analyzed:

- Energy consumption to boot the BLE sensing node.
- Energy consumption to do 1 sensor measurement (1 temperature and 1 humidity measurement (4 bytes))
- Energy consumption for transmitting 4 bytes, which means powering up RF-core, sending packet and powering down RF-core.
- Power consumption of the BLE sensing node if it is in shutdown mode.

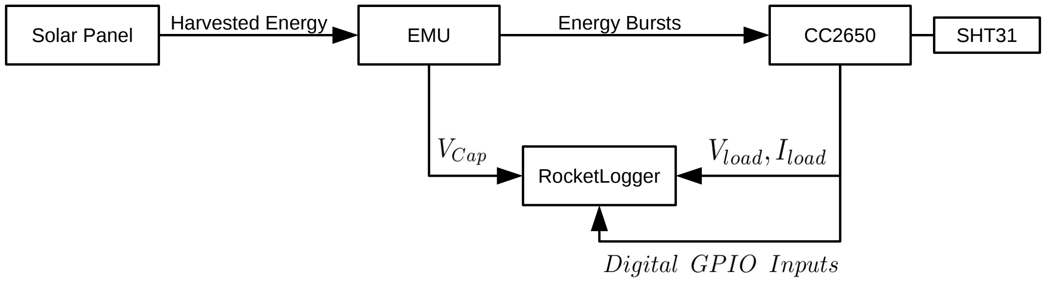


Figure 4.7: RocketLogger measures the voltage stored in the Ccpacitor (V_{Cap}), the voltage with which the sensor node is supplied (V_{load}) and the current on the board (I_{load}).

4.2.2 Results

In the table below the energy consumption of the different stages are presented and also the time it takes to execute them. If the system is in shutdown mode, it consumes $98\,\mu W$.

	boot	sensor reading	BLE
Energy	$4.36\,\mu J$	$70\,\mu J$	$30\,\mu J$
Time	$333.1\,ms$	$5.3\,ms$	$1.6\,ms$

Table 4.1: Energy consumption and time for the different parts of one task execution (reading/sending 4bytes)

All these parts summed up corresponds to one single task execution of the Application. It takes $4.46\,mJ$ and $340\,ms$ to execute a task. With the formula presented in [4] the minimal capacitance can be calculated.

$$C_{min} = \frac{2E_{load}}{\eta_{buck}(V_{max}^2 - V_{load}^2)} = 481\,\mu F \qquad with\ V_{max} = 5.25\,V,\, V_{load} = 2.4\,V$$

E_{load} is the energy consumed by the task, which is $4.46\,mJ$. V_{max} is voltage the capacitor will be loaded up to and at which the EMU triggers the sensing node. V_{load} corresponds to the minimal voltage with which the system has to be provided at least. η_{buck} is the buck converter efficiency and it is set to 0.85.

With the calculated EMU parameters we now can integrate the BLE sensing node into a transient system

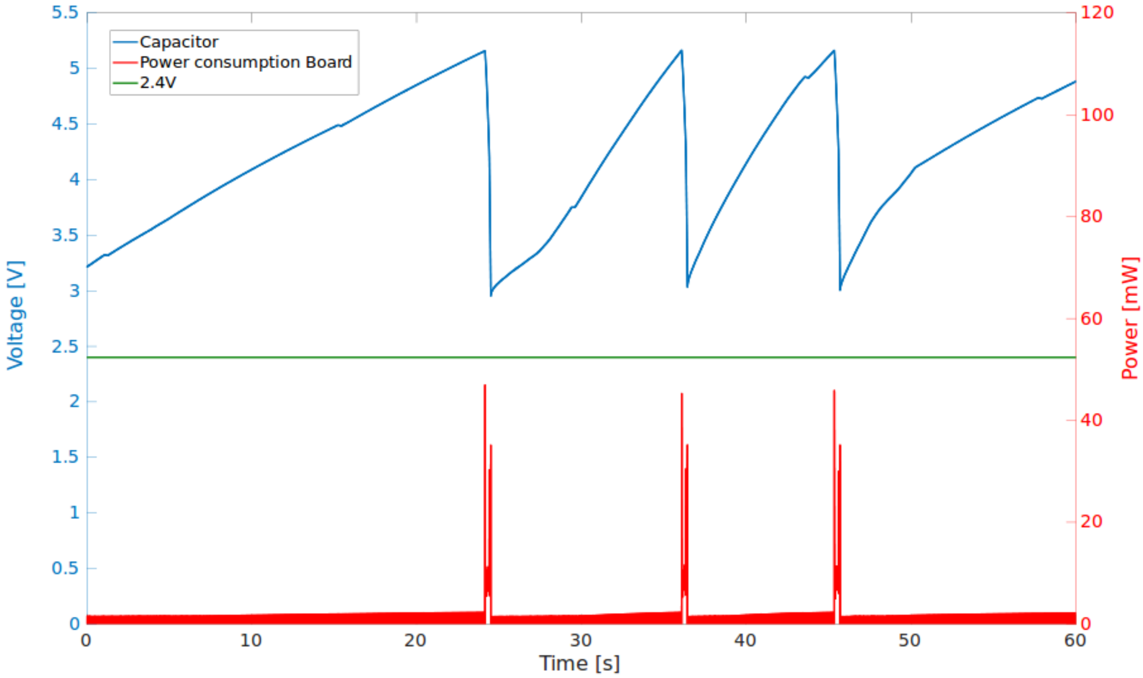


Figure 4.8: Real World application: Power consumption of the board in red and the Voltage buffered in the EMU in blue

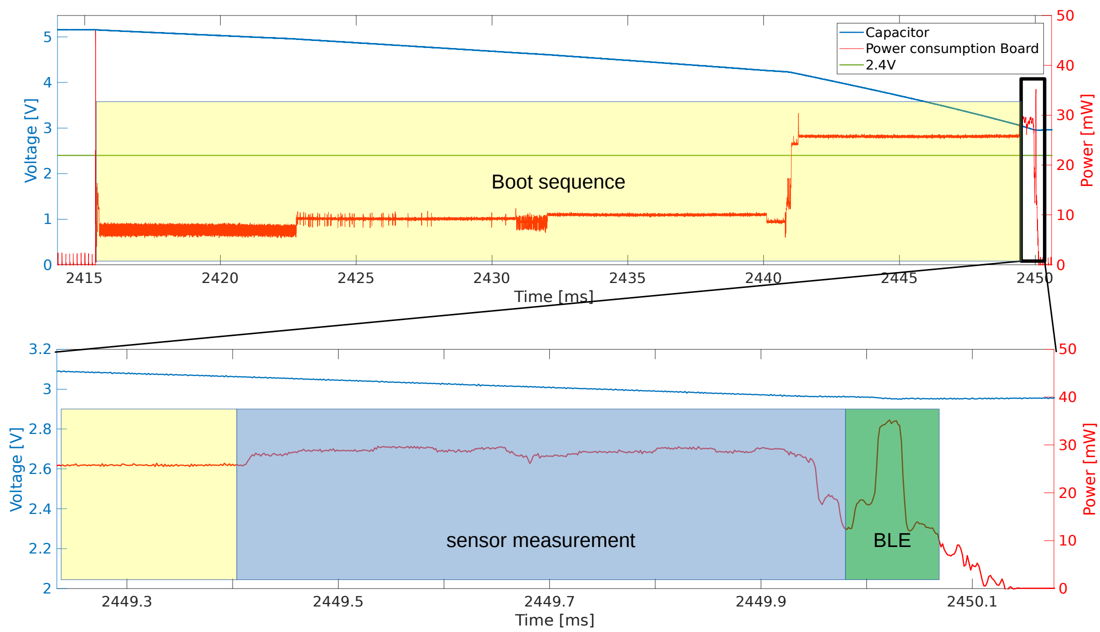


Figure 4.9: Zoomed in to the part of the task execution

Figure 4.8 shows the behavior of the transient system over a time of one minute. In blue, the voltage buffered in the capacitance of the EMU is presented, which is loaded up to 5.25 V before the sensing node is triggered and a task is executed. The voltage never falls down to the

minimal demanded voltage of 2.4V (green line), because the application in this plot is working with a capacitor of $570\,\mu V$ instead of the minimal $481\,\mu V$. After the first task execution, some additional light (flashlight) has been added and as a result the slope of the line increases.
The red line shows the power consumption of the board. It has three peaks where the sensing node is booting, reading the sensor and transmitting a packet each time. In times of loading the capacitance up, the board is consuming the $98\,\mu W$ as described above.
If we zoom in to the interesting part of the task execution (see Figure 4.9), it is easy to see the different parts described in Table 4.1. The yellow part shows the boot sequence, that consumes most of the time and energy. In numbers it is about $98\,\%$ of the whole task execution from triggering the system to shutting it down. The smaller black box shown in the upper plot contains the sensor measurement and the BLE part, which are shown in the plot below.

Chapter 5

Conclusion and Outlook

5.1 Conclusion

In this bachelor thesis, a transiently powered system is presented that harvests energy from a solar panel to sporadicly send BLE advertisements, whereas the execution rate depends on the currently available input power. It contains a sensor to measure temperature and humidity of the ambient air. The evaluation has shown that the connectionless BLE communication has a linear behavior by changing the payload of an advertising packet. There is some freedom of using the available energy in terms of payload and TX-Power. The transient system is able to perform batteryless, containing an adjusted capacitor to run without maintenance in long-term scenarios. The task execution frequency could be maximized since the boot sequence has not been optimized yet.

5.2 Future Work

5.2.1 Optimize Boot Sequence

To improve the System, the energy consumption of the booting sequence has to be reduced. In this thesis, the whole program was written on top of an operating system. A lot of energy could be saved if no operating system was used and only necessary code is written. Another approach could be to go to Standby mode instead of shutting the board down, where the board consumes a little more power but is only waking up instead of booting.

5.2.2 Scheduling for Different Tasks

In the real world application in the end only one task was executed. What if there were different tasks? With the gained information about the BLE and sensing part a smart scheduling could be achieved. The parts can be used separately. Different sizes of data can be buffered and dependent on the energy availability, different sizes of BLE Packets could be transmitted.

Bibliography

[1] A. Gomez, L. Benini, and L. Thiele, "Designing the batteryless iot," Mar 2017.

[2] P. Zhang, D. Ganesan, and B. Lu, "Quarkos: Pushing the operating limits of micro-powered sensors," in *Proceedings of the 14th USENIX conference on Hot Topics in Operating Systems*. USENIX Association, 2013, pp. 7–7.

[3] A. Gomez, L. Sigrist, T. Schalch, L. Benini, and L. Thiele, "Wearable, energy-opportunistic vision sensing for walking speed estimation," in *Sensors Applications Symposium*. Glassboro, New Jersey, USA: IEEE, Mar 2017.

[4] A. Gomez, L. Sigrist, M. Magno, L. Benini, and L. Thiele, "Dynamic energy burst scaling for transiently powered systems," in *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. EDA Consortium, 2016, pp. 349–354.

[5] S. S. Chawathe, "Beacon placement for indoor localization using bluetooth," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 980–985.

[6] H. G. Lee and N. Chang, "Powering the iot: Storage-less and converter-less energy harvesting," in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*. IEEE, 2015, pp. 124–129.

[7] J. C. Haartsen, "The bluetooth radio system," *IEEE personal communications*, vol. 7, no. 1, pp. 28–36, 2000.

[8] J. DeCuir, "Introducing bluetooth smart: Part 1: A look at both classic and new technologies." *IEEE Consumer Electronics Magazine*, vol. 3, no. 1, pp. 12–18, 2014.

[9] L. Sigrist, A. Gomez, R. Lim, S. Lippuner, M. Leubin, and L. Thiele, "Measurement and validation of energy harvesting iot devices," in *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 1159–1164.

Appendix A

Title

A.1 Implementation of the Software in the Contiki tree and explanation of the code and data

Batteryless BLE communication set up:

The operating system for all the code is contiki OS. To be able to run the different files, first the contiki tree has to be modified. For that the folder temperature inside the platform folder has to be copied to /contiki/platform. Also the files rf-ble.c and rf-ble.h have to be replaced in /contiki/cpu/cc26xx-cc13xx. They also could be copied to the temperature folder, but then in the codes the line #include "rf-core/rf-ble.h" has to be changed to #include "rf-ble.h" and in in the Makefile.launchpad file in the temperature folder the line BOARD_SOURCEFILES += rf-ble.c has to be added.

The command for flashing the cc2650 board with this platform will be:

make BOARD=temperature/cc2650 filename.upload PORT=/dev/ttyACM0 (or ACM1 for example)

In the C Code folder are the different files to flash to the Board. The MATLAB folder contains the resulting measurements made with these c files and also a Matlabcode to get plots and data matrices.

bootRF_TX_powerdownRF:
PIN 12 (GPIO) is pulled high before powering the radio up and pulled low after shutting it down, to measure the whole BLE action.

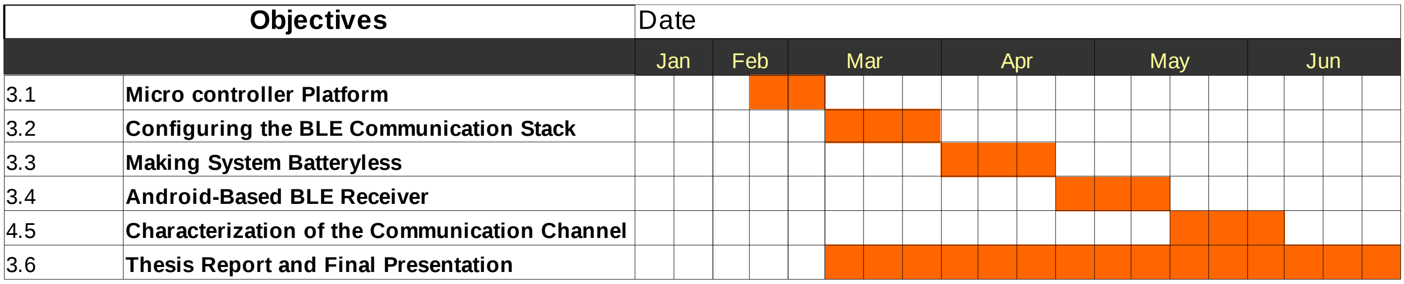
Sensor_reading_4_byte:
PIN 12 (GPIO) is pulled high before sensor reading and pulled low after, to measure the whole sensor action

Transient_System:
With this file, the energy of the whole procedure shall be measured.
The task is triggered by pulling the PIN 15 high.
The system wakes up and pulls the reset pin high. From this point to the point where PIN 27 (GPIO) is pulled low energy is consumed.
There is some additional energy by shutting the board down but this is very few compared to the rest.

In the MATLAB folder there is an additional folder called "sending BLE(only cmd to core)". This data was gained by only measuring the energy consumed for the BLE part without looking at powering up and shutting down the RF-core and without looking at configuring tasks. If you have a look at the rf-ble.c file, you can see that in line 145 and 159 there are the commands for pulling a PIN high and low. This is the section where the actual command is sent to the RF core to send the packet out.

Appendix B

Original Timetable



Date	Objectives
	Familiarization with the Microcontroller Platform
7.3	Familiarization with CC2650 and Temperature Sensor /SPI
14.3	Writing a program to read out temperature/humidity
	Configuring the BLE Communication Stack
21.3	Familiarize with different operating modes of BLE and TI BLE software stack
28.3	Integrate the software stack into temperature sensing application
4.4	Optimize software for fast wake-up
	Making the System Batteryless
11.4	Familiarization with the EMU/Rocketlogger
18.4	Connect sensor node to solar cell or etc. and characterize the tasks energy via Rocketlo
25.4	Integration of EMU control interface and possible optimization
	Android-Based BLE Receiver
2.5	interim result
9.5	interim result
16.5	Android-Based Application
	Characterization of the Communication Channel
23.5	interim result
30.5	interim result
6.6	Analyse Systems Performance
	Thesis Report and Final Presentation
13.6	
20.6	
26.6	Hand in Final Thesis Report

Appendix C

Original Problem

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Bachelor Thesis at the
Department of Information Technology and
Electrical Engineering

for

Carlo Signer

Batteryless Bluetooth Low Energy Communication

Advisors: Lukas Sigrist
Andres Gomez

Professor: Prof. Dr. Lothar Thiele

Handout Date: 20.02.2017
Due Date: 25.06.2017

1 Project Description

1.1 Introduction

In traditional energy harvesting applications, sensor nodes are able to store any excess energy in a storage device such as battery or supercapacitor for future use. If correctly dimensioned, such a system will guarantee the node's operation during periods of energy unavailability. More recently, a radically different research line has emerged which focuses on batteryless, energy-driven sensor nodes. These nodes are designed to work with transient energy sources, i.e. highly volatile sources which produce small bursts of energy. As a result, the node's operation is highly sporadic, depending entirely on the source's energy availability. Our experience shows that even in these adverse scenarios with extremely low power budgets rich data sensing and computationally intensive local processing is viable. In a next step we want to explore the feasibility of wireless data communication with commodity devices using the Bluetooth Low Energy (BLE) protocol.

2 Project Goals

The goal to this bachelor thesis is to implement a harvesting supplied wireless sensor node in order to show the feasibility of using a standardized and widely adopted wireless communication protocol in energy harvesting scenarios. The students will implement an embedded application that senses the ambient temperature and transmits the read sensor data using Bluetooth Low Energy (BLE). The application is running on an embedded platform powered only by harvested energy using the proposed *Energy Burst* execution scheme [2]. To demonstrate the communication with commercial, BLE capable devices, a basic BLE receiver application on top of a smartphone operating system will also be implemented as part of this thesis. Finally, the energy trade-offs of different communication parameters are characterized using the batteryless communication environment built in this thesis.

3 Tasks

The project will be split up into several subtasks, as described below:

3.1 Familiarization with the Microcontroller Platform

At the beginning of the project the focus lies on understanding the Texas Instruments CC2650 [4] system-on-a-chip platform used for this project. The basic principles that will be required throughout the project are the use of GPIOs, reading a temperature sensor via SPI or I2C, deep sleep and external wakeup trigger. In this step a first program to read out a temperature/humidity sensor initiated by an external trigger such as a press on a button that will later be used as sensor.

3.2 Configuring the BLE Communication Stack

As sensor values only stored on the node itself are not that useful, the student will extend the node with wireless communication using the Bluetooth Low Energy (BLE) protocol. For this, the student will familiarize himself with the different operating modes of BLE communication and the Texas Instrument BLE software stack. The BLE software stack is then integrated into the temperature sensing application to communicate the read sensor values using the stateless BLE beacon communication scheme. Once the communication is working, the software shall be optimized for fast wake-up to communication time, which will be important to turn the application into a batteryless sensor.

3.3 Making the System Batteryless

After familiarizing and understanding the Energy Management Unit (EMU) principle[2], the student will adapt the previously battery powered sensor node to support a energy burst execution scheme. This includes characterizing the individual task's energy and runtime (using the RocketLogger [3]), the integration of the EMU control interface and possible optimizations for this task execution scheme.

3.4 Android-Based BLE Receiver

To terminate the wireless communication on the receiver side and log information relevant for the evaluation, the student will implement a basic, Android based BLE beacon receiving and logging application. This application shall log all received BLE beacon messages with precise timestamps and all packet information like sender and message content. The student may directly implement this application on top of the Android API or a third-party library like AltBeacon library[1].

3.5 Characterization of the Communication Channel

In a last step, the student will thoroughly analyze the performance of batteryless BLE communication. By measuring the nodes harvesting rate and power consumption while in parallel recording the packet receptions in a stationary the energy trade-offs of packet size and transmission power shall be characterized in a one-to-one communication scheme.

3.6 Thesis Report and Final Presentation

Finally, a thesis report is written that covers all aspects of the project. The results are also presented in a final presentation during the group meeting of the Computer Engineering Lab.

4 Project Organization

4.1 Weekly Meeting

There will be a weekly meeting to discuss the project's progress based on a schedule defined at the beginning of the project. A short report concerning each week's progress (accomplished goals, difficulties, questions, next steps) should be provided at latest the day before the meeting.

4.2 Thesis Report

Two hard copies of the report need to be turned in at the end of the thesis. The copy remains property of the Computer Engineering and Networks Laboratory. All developed software and supporting documents shall be checked in to the provided repository in a clearly defined structure together with a documentation of the individual files and folders

4.3 Initial and Final Thesis Presentation

In the first month of the project, the topic of the thesis will be presented in a short presentation during the group meeting of the Computer Engineering Lab. The duration of the talk is limited to five minutes. At the end of the project, the outcome of the thesis will be presented in one 15 minutes talk. It will take place during the group meeting of the Computer Engineering Lab.

4.4 Work Environment

The work will be carried out in the framework of the SNF project "Transient Computing Systems". This means that the results of this work can be used by the involved project partners if the project goals are met.

References

[1] Altbeacon. http://altbeacon.github.io/android-beacon-library/.

[2] A. Gomez, L. Sigrist, M. Magno, L. Benini, and L. Thiele. Dynamic Energy Burst Scaling for Transiently Powered Systems. In *Conference on Design, Automation and Test in Europe*, pages 349–354, Dresden, Germany, mar 2016.

[3] L. Sigrist, A. Gomez, R. Lim, S. Lippuner, M. Leubin, and L. Thiele. Measurement and Validation of Energy Harvesting IoT Devices. In *Conference on Design, Automation and Test in Europe*, Lausanne, Switzerland, 2017.

[4] Texas Insturments. *CC2650 SimpleLink multi-standard 2.4 GHz ultra-low power wireless MCU*, 2017.

Zurich, February, 2017