

---

# Low-Power Wakeup Receiver

Semester Thesis 2019

---

**Authors**

Cédric Renda, Manuel Tischhauser

**Supervisor**

Prof. Dr. Heinz Mathis

**Assistant Supervisor**

Selina Malacarne

**Subject**

Wireless Communications

HSR Hochschule Für Technik Rapperswil

December 17, 2019



# **Abstract**

## **Introduction**

In company buildings, universities and other similar facilities, it is usual to attach occupancy schedules next to a room entrance. These schedules are most of the time printed on paper, or in more modern buildings, displayed on a screen. Using paper, it is impracticable to take short-term changes into account, since every adjustment needs to be made by printing a new schedule. Hence, screens need to be connected to the power grid, which entails extensive installation work, or powered by a battery, which needs to be replaced from time to time. All these disadvantages could be avoided by using a screen which harvests its energy from the indoor lightning, and is updated via a wireless interface.

## **Approach**

To power the system, solar cells are used to harvest energy, which is stored in a super-capacitor to ensure small power leakage. Because this kind of harvesting unit cannot provide vast amounts of energy over a long period of time, the whole system can only fully run in short bursts. To save energy, an e-paper display is used, which only needs energy to update the screen, but not for maintaining the displayed image. Furthermore a constantly listening wakeup receiver is used. When receiving a specific wakeup sequence, it will generate an interrupt to enable the power supply of the remaining system. The remaining part then receives data via Bluetooth, displays the data and in the end cuts its own power supply, to enter a new cycle of low-power listening mode.

## **Conclusion**

Energy harvesting provides enough energy, if updating the schedule does not occur too frequently. This semester thesis shows, that a wakeup receiver is suitable for this kind of problem. With the implementation of a self-holding circuit, it is even possible to switch off the whole system (except for the wakeup receiver) when unused. The wakeup receiver consumes a maximum of  $4.5 \mu\text{W}$  during listening mode while the system harvests around  $485,52 \mu\text{W}$ .



# Contents

<b>Abbreviations</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Task analysis . . . . .	3
1.2 Approach . . . . .	3
<b>2 Theory</b>	<b>5</b>
2.1 E-Paper Display . . . . .	5
2.2 Wakeup Receiver . . . . .	6
2.3 software grafik zeichen . . . . .	7
<b>3 Evaluation</b>	<b>9</b>
3.1 Wakeup Receiver . . . . .	9
3.1.1 AS3933 . . . . .	9
3.1.2 RFficient . . . . .	10
3.2 Microcontroller . . . . .	10
3.2.1 MSP430 . . . . .	10
3.2.2 STM32 . . . . .	11
<b>4 Development</b>	<b>13</b>
4.1 Hardware . . . . .	13
4.1.1 Microcontroller . . . . .	13
4.1.2 Communication . . . . .	13
4.1.3 Power Management . . . . .	15
4.1.4 Display . . . . .	20
4.2 Software . . . . .	21
4.2.1 Receiver . . . . .	21
4.2.2 Transmitter . . . . .	22
<b>5 Results</b>	<b>25</b>
5.1 Power consumption . . . . .	25
<b>6 Summary</b>	<b>27</b>
<b>Sources</b>	<b>29</b>
<b>List of figures</b>	<b>29</b>

---

<b>List of tables</b>	<b>31</b>
<b>Statement of Plagiarisms</b>	<b>33</b>
<b>A Requirements</b>	<b>37</b>
A.1 Assignment . . . . .	38
A.2 Requirement Specification . . . . .	40

# Abbreviations

**ASK** Amplitude Shift Keying

**BLE** Bluetooth Low Energy

**CPU** Central Processing Unit

**FIFO** First In First Out

**FRAM** Ferroelectric Random Access Memory

**GPIO** General Purpose Input/Output

**GUI** Graphical User Interface

**HSR** Hochschule für Technik Rapperswil

**IIS** Fraunhofer-Institut für Integrierte Schaltungen

**IoT** Internet of Things

**OOK** On-Off Keying

**SPI** Serial Peripheral Interface

**SRAM** Static Random Access Memory

**TTL** Transistor-Transistor Logic

**UART** Universal Asynchronous Receiver Transmitter

**USB** Universal Serial Bus



# Chapter 1

## Introduction

In company buildings, universities and other similar facilities, it is usual to attach occupancy schedules next to room entrances. These schedules are mostly printed on paper, or in more modern buildings, displayed on a screen. Using paper, it is impracticable to take short-term changes into account, since every adjustment needs to be made by printing a new schedule. Screens on the other hand need either to be connected to a power supply, which entails extensive installation work, or are powered by a battery, which needs to be replaced from time to time. All these disadvantages could be avoided by using a screen which can be updated wireless and is in itself autarkic. The goal of this semester thesis is to develop a functional prototype of such a schedule. To keep the energy consumption at a minimum, a low-power wakeup receiver is used.

### 1.1 Task analysis

The prototype should combine both the advantages of the paper method and the screen. Updating should happen via a wireless interface while a expensive installation should be avoided. The screen should therefore be placed on the desired location just like a paper schedule. To consume as low power as possible, the microcontroller, display and interface should be switched off when not needed.

### 1.2 Approach

To simplify the description, from now on the display with all its components is called the receiver, while the other end, a computer with additional hardware is called the transmitter. This does not mean, that the communication is uni-directional, since both ends should be able receive and transmit data, but more that the general data transfer happens from the computer to the screen.

**Transmitter** This end only consists of a computer, a wireless interface for the main communication and the transmitter module of the wakeup receiver. A short python script enables the user to define which data the receiver should print over a fixed template. Information is transferred to a wireless interface which transmits it to the receiver end after the receiver is woken up.

---

**Receiver** This part of the prototype should consume as low power as possible. An e-paper display is used, because it is bi-directional, which means it can maintain its image even when disconnected from the power supply. A wireless module receives information from the transmitter and, if needed, sends back feedback about the status. This connection should therefore be bi-directional. To process the incoming data and display it correctly on the screen, a microcontroller is needed. All these components are completely disconnected from the power supply when not used. The power is provided by a super capacitor which is charged by solar cells. When switched off, the only component which actively consumes power is a wakeup receiver. This module is constantly in listening mode and waits for a wakeup event. If such an event is received, it activates a self-holding circuit which connects the power supply to the other components. Data can now be exchanged over a different channel, established by the wireless module. After the data is processed and the display is refreshed, the microcontroller deactivates the self-holding circuit and disconnects in this step itself and all other components except the wakeup receiver from the power source.

# Chapter 2

## Theory

### 2.1 E-Paper Display

There are several different technologies which are applied in e-paper displays. But since the developed prototype uses a microencapsulated electrophoretic display, this section only describes this specific implementation.

A film, consisting of microscopic capsules, is coated onto a backplane, which basically is a matrix of electrodes. On top of that comes a common electrode, called the frontplane. This setup is shown in Figure 2.1.

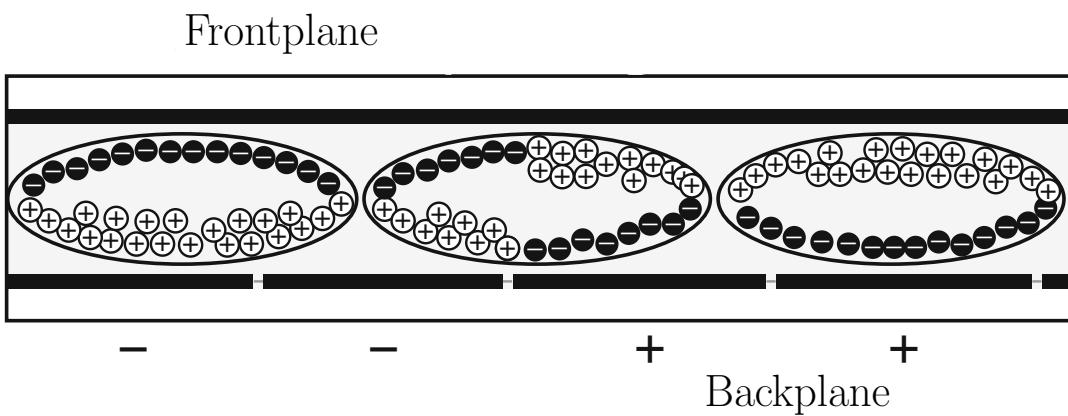


Figure 2.1: Capsules between electrodes (taken from [1] with adjustments).

Each capsule in the film itself contains a transparent fluid and particles of two opposite charges. One type of particle scatters the light while the other one absorbs it. The particles now begin to move in opposite directions through the fluid when exposed to a electrical field. The pixel (defined by the electrode size) appears white if the scattering particles are near the frontplane and black if otherwise. If no voltage is applied to the electrodes, the particles maintain their last position. This makes it possible to switch off the power supply when no image update is needed which is the reason why the power consumption is strongly reduced compared to other types of display [1].

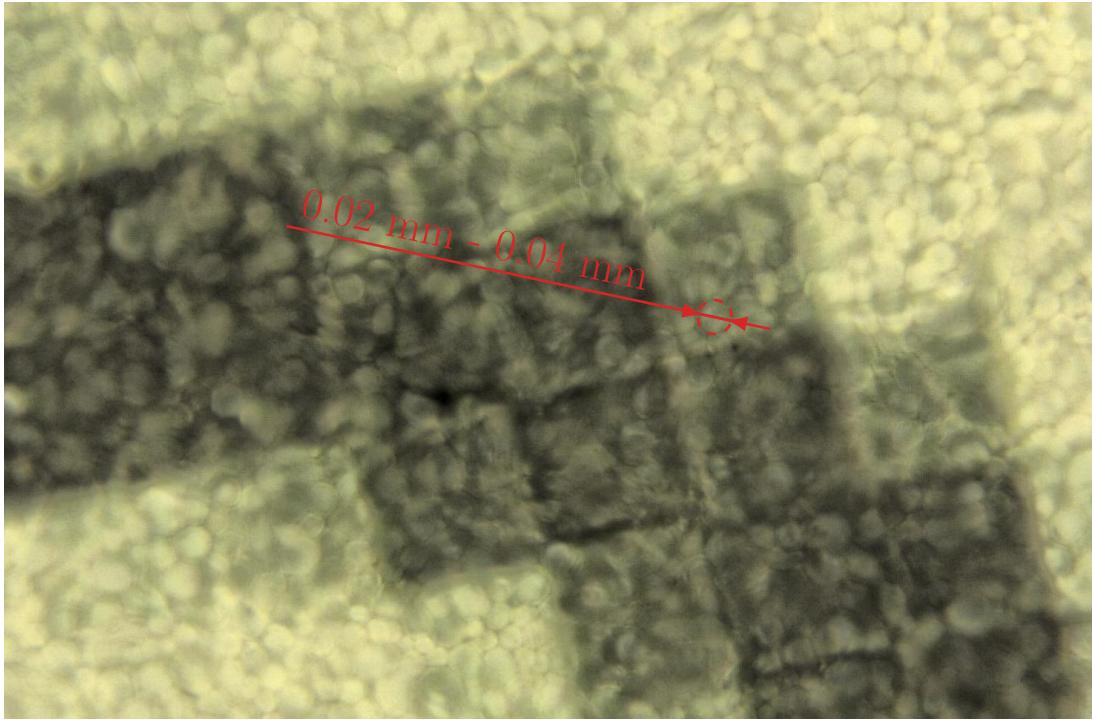


Figure 2.2: E-paper display under microscope with 250x magnification.

Figure 2.2 shows a e-paper display with 250x magnification. One electrode is coated by many capsules, which are between 0.02 mm and 0.04 mm in diameter.

## 2.2 Wakeup Receiver

To save power with a conventional receiver, it needs to be programmed in such a way, that it is kept in a sleep mode. To check if any data was sent, it needs to wake up periodically to check for notifications. To set this duty cycle is a trade off between response time and power consumption. Is the period longer, the receiver is also longer time in its sleep mode. On the other side defines this period directly the response time, since data can only be received in the running mode. A wakeup receiver now allows a device to be constantly in listening mode, while consuming low energy. Added to the system, the actual microcontroller, which coordinates the data transmission and other tasks stays shut down with only the wakeup receiver in listening mode. After receiving a defined pattern over this channel, the wakeup receiver generates an interrupt to wake up the microcontroller. It can now establish a channel over a different wireless module, or execute another task. When done, the microcontroller puts itself and all other modules except the wakeup receiver in sleep mode again. Figure 2.3 shows a comparison of both the conventional approach and the solution with the wakeup receiver.

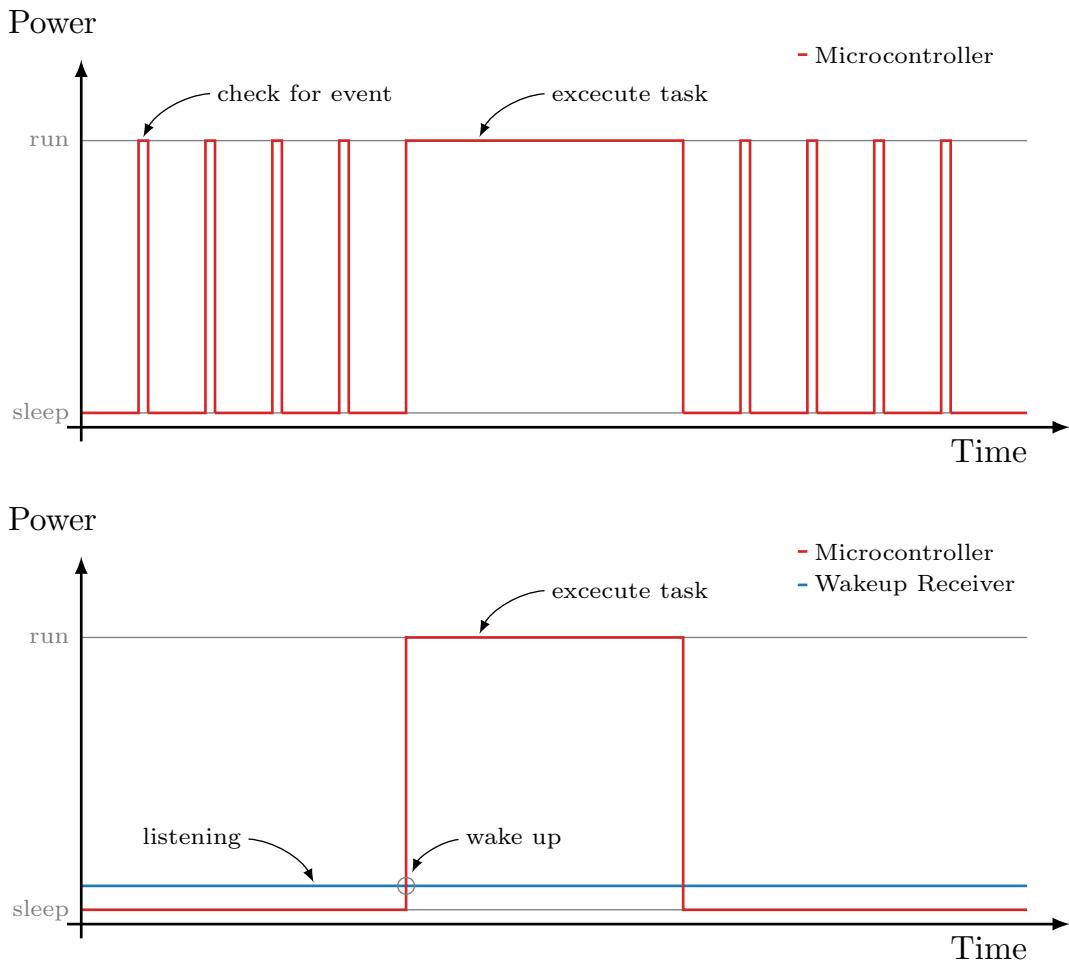


Figure 2.3: Microcontroller checks periodically for incoming data (top), additional, constantly listening wakeup receiver (bottom).

One can argue, that the wakeup receiver module consumes in general more power, than the microcontroller in sleep mode. But since the microcontroller only wakes up, when a task needs to be done, the overall energy consumption (area underneath the curves summarized) is going to be smaller if the occurring wakeup event comes infrequently or over longer periods of time. The response time on the other hand can be kept in the microsecond range.

## 2.3 software grafik zeichen



# Chapter 3

## Evaluation

### 3.1 Wakeup Receiver

In this semester thesis, two different implementations of the wakeup receiver technology were compared. The AS3933 from AMS and the RFicient from the Fraunhofer-Institut für Integrierte Schaltungen (IIS) which kindly provided an evaluation kit before the actual release of the product.

#### 3.1.1 AS3933

The AS3933 is a low frequency wakeup receiver, which uses ASK to modulate a carrier frequency between 15-150 kHz. The transmitter sends a manchester encoded, programmable wakeup pattern of length 16 or 32 bit. If this pattern is detected on the receiver end, a wakeup interrupt is generated. It is also possible to disable the the pattern decoder to run the chip in a frequency detection mode, where a wakeup interrupt is generated as soon as the specified frequency is received. More important features on the receiver end are:

- Receiver sensitivity  $80\mu\text{V}_{\text{RMS}}$
- Current consumption in 3-channel listening mode  $2.3\mu\text{A}$
- Operating supply range  $2.4\text{ V} - 3.6\text{ V}$
- Three antennas (enables 3D detection)
- Channels individually selective to reduce power consumption

The low power consumption makes it possible run the receiver in listening mode below  $8.3\mu\text{W}$ [2].

The demo kit comes with a GUI, which enables the user to set the parameters as desired and address the registers directly. The range of the receiver of about 6 m as first measurement turns out to be very limited. Even with a 3db sensitivity boost on the receiver side, it is only possible to detect wakeup events from a distance of 9 m. The environment (indoor, outdoor) seems to make no huge difference.

As a result, the limited range makes the AS3933 unusable for the prototype.

---

### 3.1.2 RFicient

The RFicient from the IIS uses OOK to modulate a 868 MHz signal. It can either run in pure wakeup mode, where the receiver generates an interrupt as soon as a code is received or a selective mode, where a 16 bit wakeup preamble needs to match the receiver. After the preamble is detected, the data rate can be changed to transfer more bits which can be sent over an SPI-bus to a connected device. This way, it is possible to transmit data bits after the actual wakeup. Data rates can be set in a range between 256 bp/s – 32 kbp/s. The most important features are:

- Receiver sensitivity -80 dBm
- Energy consumption  $3 \mu\text{A}$  at 1.5 V (data rate 1 kbit/s)
- Unidirectional data transfer possible

The power consumption therefore is in listening mode (data rate = 1 kb/s)  $4.5 \mu\text{W}$ .

Just as the AS3933, the RFicient demo kit comes with a GUI, which enables the user to set the important parameters and access the register [3]. First measurements showed that the range of the RFicient is far higher than the range of the AS3933. It is therefore used in the prototype.

## 3.2 Microcontroller

To handle incoming data and write it to the display driver, some kind of microcontroller is needed. A derivate of the MSP430 and one of the STM32 were looked at more closely.

### 3.2.1 MSP430

The MSP430FR6989 was up for selection because it is specifically developed for ultra-low-power applications. Its main features are:

- 16-bit architecture with up to 16 MHz clock
- Supply voltage range from 1.8 V to 3.6 V
- Approximately  $100 \mu\text{A}/\text{MHz}$  current consumption in active mode
- 128 KB Ferroelectric Random Access Memory (FRAM)

The display driver chip can be controlled over SPI [4].

During development, it turned out to be very useful to store a complete image. The required memory to do this assuming 16 grey levels and a  $1200 \times 825$  size display is:

$$M = 1200 \cdot 825 \cdot 4 \text{ bit} = 495'000 \text{ B} < 484 \text{ kB.}$$

the FRAM is therefore not sufficient. This is the main reason, why a different microcontroller is needed.

---

### 3.2.2 STM32

The STM32L4R5ZI by STMicroelectronics is just like the MSP430 developed for low-power applications. Most important features are:

- 32-bit Cortex-M4 CPU
- Several clock sources between 4 and 48 MHz
- 1.71 V - 3.6 V input voltage supply
- $110 \mu\text{A}/\text{MHz}$  current consumption in run mode
- 2 MB Flash and 640 KB SRAM

The display driver can again be controlled over SPI [11].

The STM32 is in general more flexible and most important has enough memory for a complete picture. That it consumes slightly more power than the MSP430 should not be too much of a problem, since it is only in run mode a few seconds, before being completely switched off again. The standby mode of the controller is not used and therefore of no interest.



# Chapter 4

## Development

### 4.1 Hardware

Figure 4.1 shows a schematics of the receiver. Every component is described in more detail in this section.

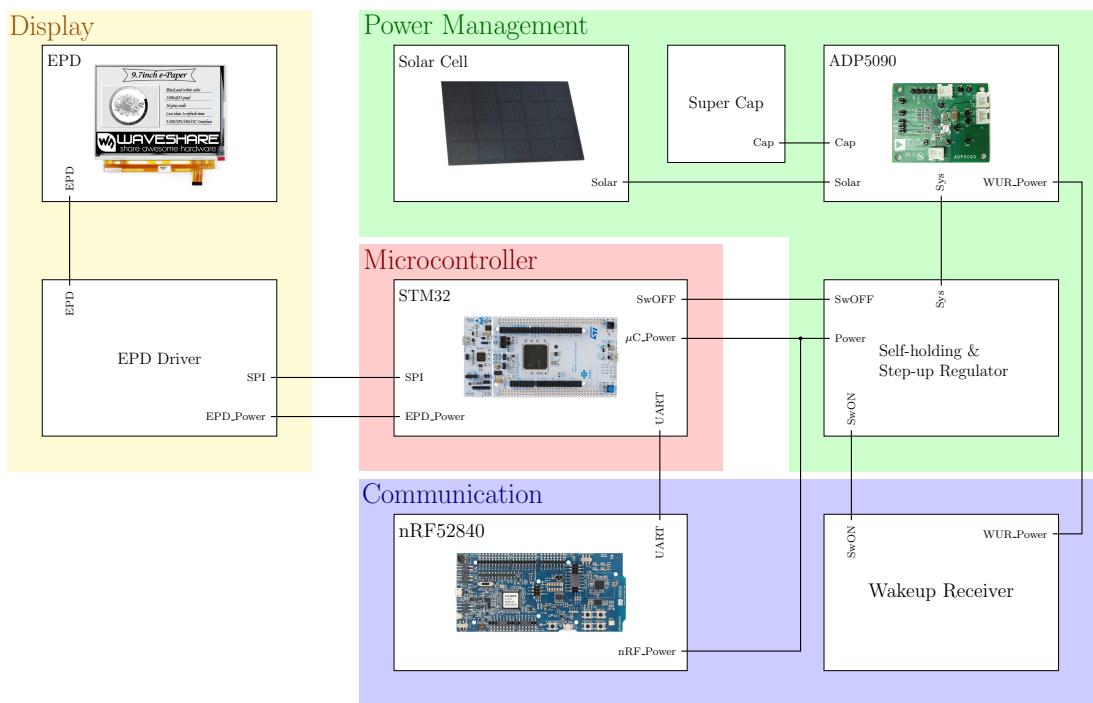


Figure 4.1: Receiver schematics.

#### 4.1.1 Microcontroller

#### 4.1.2 Communication

#### RFcient

**Transmitter** The Rfcient demo kit comes, as previously stated with a GUI. Is the transmitter module connected, simple steps make it possible to set the preamble and payload data rate. The

payload itself can partly serve as an ID and as additional transmitted data. The power of the transmission can also be set between -30 dBm and 10 dBm. The transmitter-GUI is shown in Figure 4.2. To test, if any data is received, it is helpful to activate the infinite loop (top right in

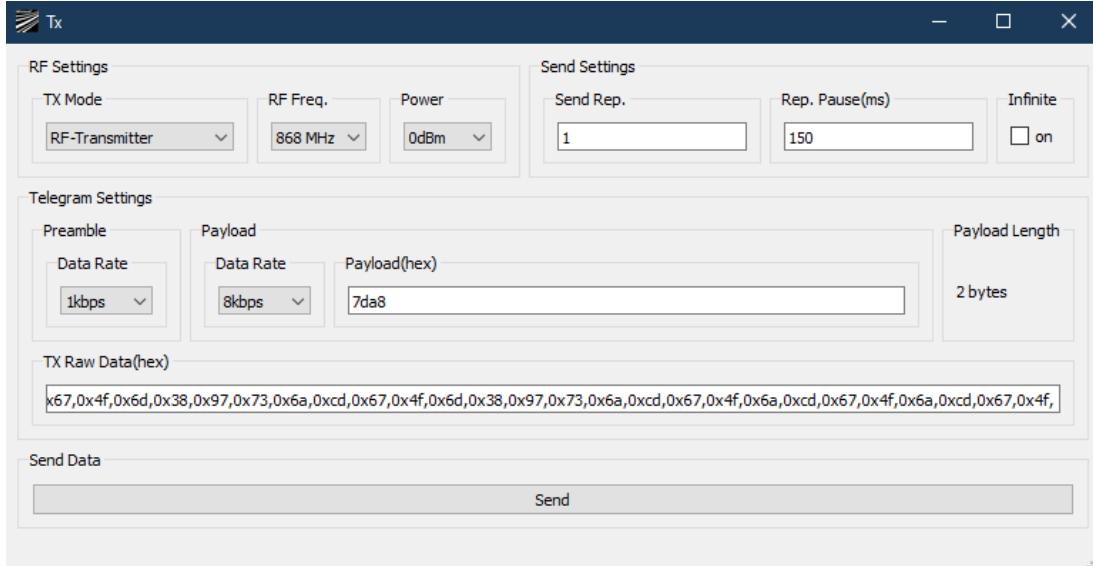


Figure 4.2: Transmitter GUI.

figure 4.2). The defined data is in this case sent repeatedly with an adjustable period.

**Receiver** To receive the data, the setting of the receiver module need to be matched to the transmitter. The provided receiver-GUI makes this step very simple. As visible in Figure 4.3, one can enable the four possible interrupt sources: CodeA/B, ID, FIFO length and FIFO overflow. In the finished product, it should be possible to wake up the receivers individually. For this reason, only the ID as interrupt source is of interest. If an interrupt is generated, the RFicient sets a GPIO-pin to high, and transfers the received data to a FIFO-buffer. This data can be passed to a microcontroller over an SPI-bus. For the purpose of this thesis, only the GPIO high level is used as a on switch of a self-holding circuit.

**Tests in a realistic environment** To test the RFicient in more detail, some indoor measurements were made. As test environment served several buildings of the Hochschule für Technik Rapperswil (HSR), since this type of facility represents a realistic condition for the developed schedule.

Purpose of the first test was to measure the achievable range. Figure 4.4 shows the hallway where this test was executed. With a transmission power of 0 dBm (1 mW) it is possible to receive interrupt over a 50 m distance (line of sight). Is the transmission power increased to 10 dBm (10 mW), interrupts can be sent over a distance up to 80 m. In this environment, the RFicient has enough range. But maybe reflections off the walls, floor and ceiling have contributed to this result.

Out of this reason, a second test was performed to check how the RFicient penetrates obstacles. As test environment served this time the HSR building 8, as depicted in Figure 4.5. An attempt was made to send wakeup signals through the ceiling. To avoid reflections, the RFicient was placed in such a way, that the zero point of the antenna pointed directly at the opposite

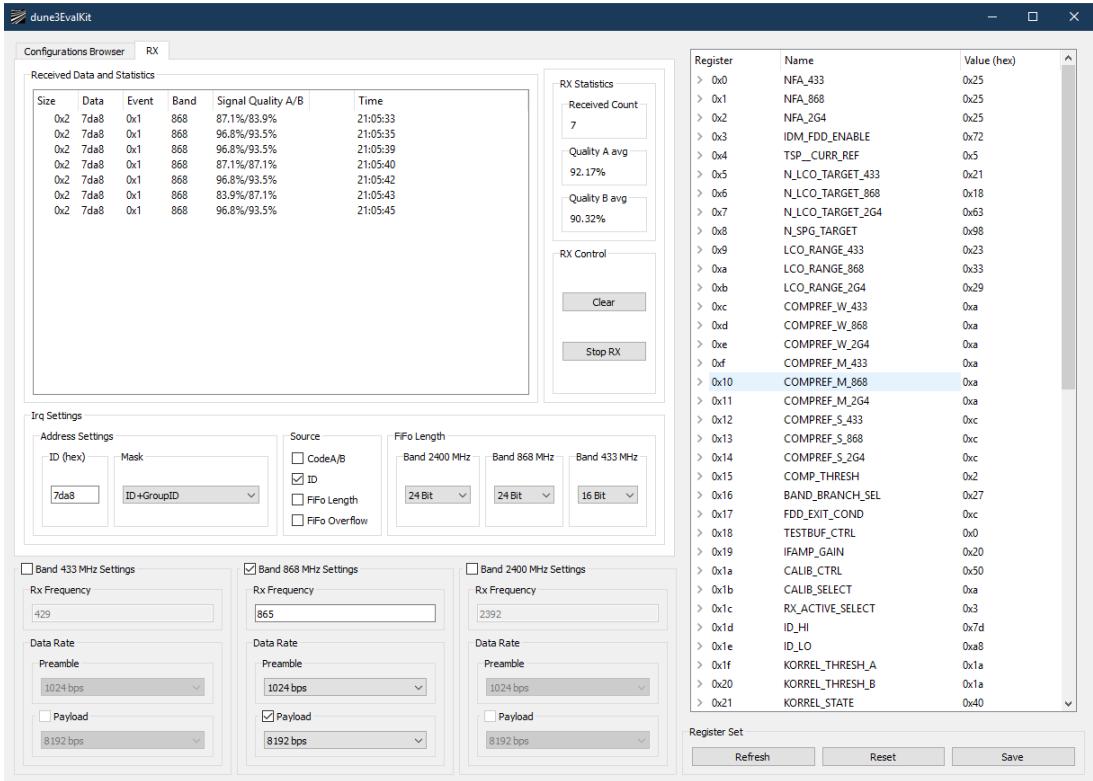


Figure 4.3: Receiver GUI.

walls. With a transmission power of 0 dBm, two ceilings can be penetrated. Is the power again increased to 10 dBm, the signal reached through three ceilings.

These measurements further confirmed, that the RFicient is suitable for the use-case of this semester thesis.

## nRF52480

A nRF52580 from nordic semiconductor serves as the wireless module. It is used because it supports multiple protocols like Thread, Zigbee and more important Bluetooth 5 which includes Bluetooth Low Energy (BLE)[5]. Since BLE allows mesh networking and is applied in many existing devices, such as smartphones, the connection to transmit data is established over this protocol. This way, the developed prototype could later be extended to a mesh network.

Two nRF52480 development kits are used, one on transmitter and one on receiver end. Data can be transferred from a computer with a USB-TTL-adapter to the UART-interface of the transmitter development kit. This data is then sent over BLE to the receiver kit and passed to the microcontroller again over UART. In the same way, data can be transferred from the receiver to the transmitter kit. The data channel is therefore bi-directional.

### 4.1.3 Power Management

The screen should be self-sustaining, thus some sort of energy-harvesting unit is needed. It was apparent to choose light as the energy source, since the screen will be used in rooms, that are artificially illuminated most of the time. The energy obtained by solar cells is converted to a



Figure 4.4: Test environment 1 (HSR building 1).

suitable voltage, using a ADP5090 chip. This way, a super-capacitor, which is used as an energy storage device, is charged.

### Solar cell

As solar cell, the AM-1522 by Panasonic is used. One panel has an area of  $55.0\text{ mm} \times 40.5\text{ mm}$  and delivers up to  $58.7\mu\text{A}$  when operating at an optimal voltage of  $2.1\text{ V}$ , provided an illuminance of  $200\text{ lux}$ . To keep a reasonable display to panel ratio, four cells were connected in parallel, which corresponds to an area of ca.  $89.1\text{ cm}^2$  (Display area =  $283\text{ cm}^2$ ). Therefore, the solar cells should provide a power of

$$P = U \cdot I = 4 \cdot 57.8\mu\text{A} \cdot 2.1\text{ V} = 485.52\mu\text{W}, \quad (4.1)$$

given a  $200\text{ lux}$  illuminance [6].

### ADP5090

The ADP5090 from Analog Devices is used to manage the energy harvesting process. This boost regulator makes it possible to charge storage elements, such as rechargeable batteries and super capacitors with the input dc-power provided by the PV-cell. Utilized features are:

- Maximum power point tracking
- Efficiency up to 90%



Figure 4.5: Test environment 2 (HSR building 8).

- Input voltage  $V_{IN}$  from 80 mV to 3.3 V
- Programmable voltage range (2.2 V to 5.2 V) for the storage element

To prevent the storage element from overdischarging, the ADP5090 enables the user to set a maximal Voltage with resistors:

$$V_{BAT\_TERM} = \frac{3}{2} V_{REF} \left( 1 + \frac{R_{TERM1}}{R_{TERM2}} \right). \quad (4.2)$$

The same procedure is applied to set a minimal voltage:

$$V_{BAT\_SD} = V_{REF} \left( 1 + \frac{R_{SD1}}{R_{SD2}} \right). \quad (4.3)$$

While discharging, the ADP5090 will switch off the output  $V_{SYS}$  if  $V_{BAT\_SD}$  is reached. This prevents the storage element from overdischarging. The output voltage  $V_{SYS}$ , where the load is attached, will therefore always stay in this programmed range ( $V_{BAT\_SD} \leq V_{SYS} \leq V_{BAT\_TERM}$ ) [7].

For this prototype, the evaluation board for the ADP5090 was used, where the internal reference voltage ( $V_{REF}$  in (4.2) and (4.3)) is 1.21 V [8].

## Super capacitor

As energy storage, a super capacitor from Taiyo Yuden (LIC1235RS3R8406) has proven to be suitable, which is a 40 F cylinder type lithium ion capacitor. The operating voltage range is between 2.2 V and 3.8 V. Discharging the capacitor lower than 2.2 V causes shorter lifetime and higher leakage current. The same unwanted behaviour occurs when charging the capacitor over 3.8 V [9].

## Combined test

To test the behaviour of the power management, supercapacitor and solar cells, a couple of measurements are executed. Figure 4.6 shows the layout of the test setup.

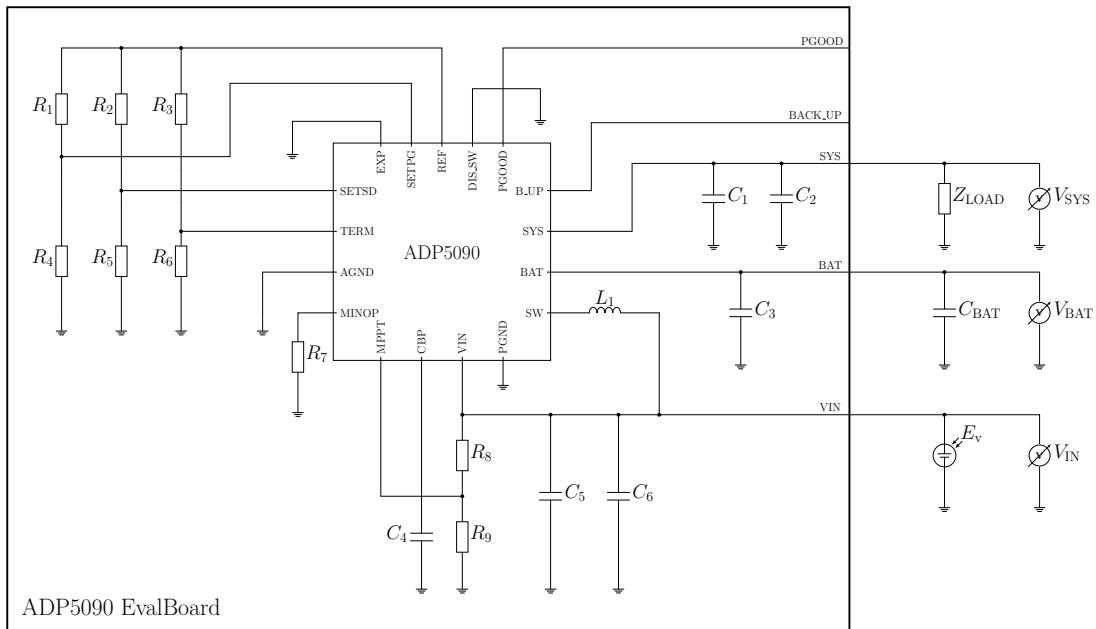


Figure 4.6: Schematics of the test setup.

To carry out these measurements, it is first necessary to adjust the minimal and maximal voltage of the ADP5090. The nrf58240 accepts supply voltages between 1.6 V up to 5.5 V [10]. The STM32 on the other side is less flexible with an input voltage range of 1.71 V to 3.6 V [11]. As stated in the section above, the super capacitors operating voltage is between 2.2 V and 3.8 V. Hence it seems reasonable, to set  $V_{BAT\_TERM} \leq 3.6$  V and  $V_{BAT\_SD} \geq 2.2$  V, to satisfy all of these three elements. In order to do this, the four resistors have to be chosen as  $R_3 = 4.3$  M $\Omega$ ,  $R_6 = 4.7$  M $\Omega$ ,  $R_2 = 4.3$  M $\Omega$  and  $R_5 = 5.1$  M $\Omega$ . Inserted in the equation (4.2) and (4.3) we get

$$V_{BAT\_TERM} = \frac{3}{2} 1.21 \text{ V} \left( 1 + \frac{4.3 \text{ M}\Omega}{4.7 \text{ M}\Omega} \right) \approx 3.48 \text{ V} \quad (4.4)$$

and

$$V_{BAT\_SD} = 1.21 \text{ V} \left( 1 + \frac{4.3 \text{ M}\Omega}{5.1 \text{ M}\Omega} \right) \approx 2.23 \text{ V}. \quad (4.5)$$

While testing, the input voltage from the solar cells ( $V_{IN}$ ), voltage of the supercap ( $V_{BAT}$ ) and the output voltage ( $V_{SYS}$ ) are tracked. Additionally, the illuminance ( $E_v$ ) near the PV-cells is recorded as indicated in figure 4.6.

The purpose of the first test is to check, if the ADP5090 converts  $V_{IN}$  to a voltage  $\leq V_{BAT\_TERM}$ . The measurements where taken over a couple hours and are plotted in Figure 4.7.

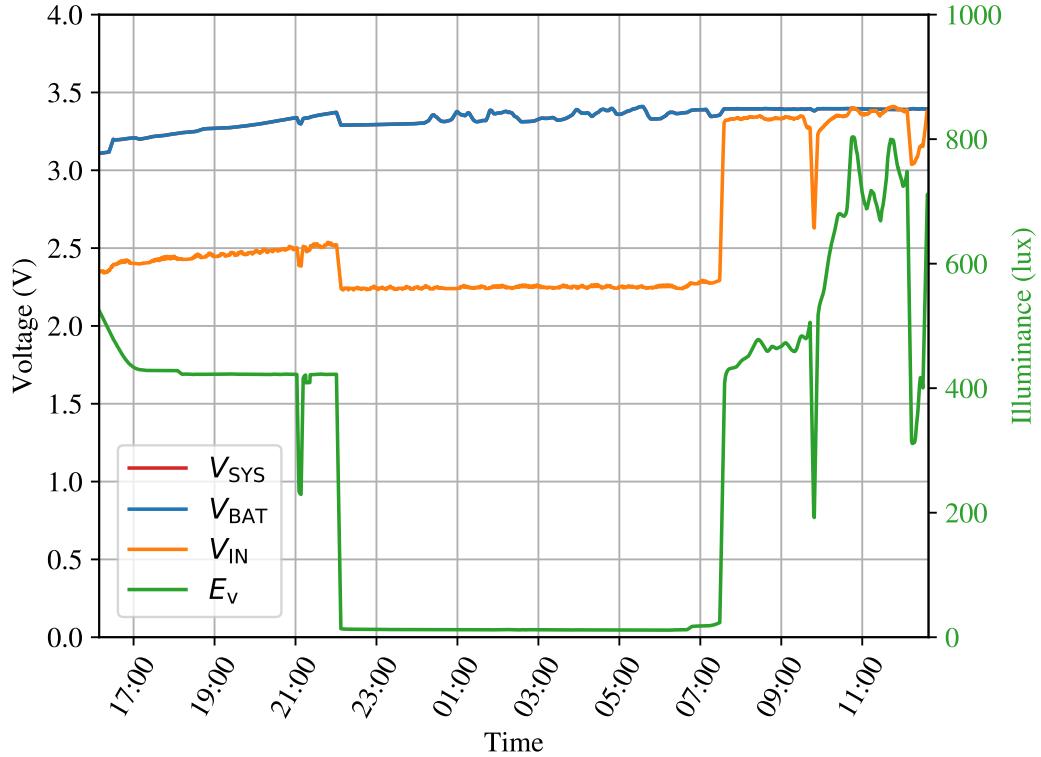


Figure 4.7: Charging behaviour.

No load was connected to the output ( $Z_{LOAD} \rightarrow \infty$ ), which is the reason  $V_{SYS}$  is overlapped by  $V_{BAT}$ . It can be seen, that between 17:00 and 23:00, the super capacitor was being charged and that the ADP5090 controls the voltage  $V_{BAT}$  like expected to the adjusted maximum voltage  $V_{BAT\_TERM}$ .

The second test should simulate the discharging when a load is connected, after the capacitor was fully charged. It was necessary to estimate the consumed power by the electronic components of the prototype. A rough measurement with a power analyser showed, that the microcontroller and the e-paper display together draw at its peak about 240 mA when connected to 5 V. The nrf52840 on the other hand, only consumes 6 mA with a 3 V source. Thus the expected consumed power at is's peak is:

$$P_e = 5 \text{ V} \cdot 0.24 \text{ A} + 3 \text{ V} \cdot 0.006 \text{ A} = 1.218 \text{ W.} \quad (4.6)$$

If  $Z_{LOAD} = 10 \Omega$  the load draws currents between 0.223 A and 0.348 A which again lead to a power consumption that should approximately match the power consumption of the finished prototype. Furthermore, the solar cells where covered to observe the discharging without interference of additionally charging behaviour. Figure 4.8 shows the result.

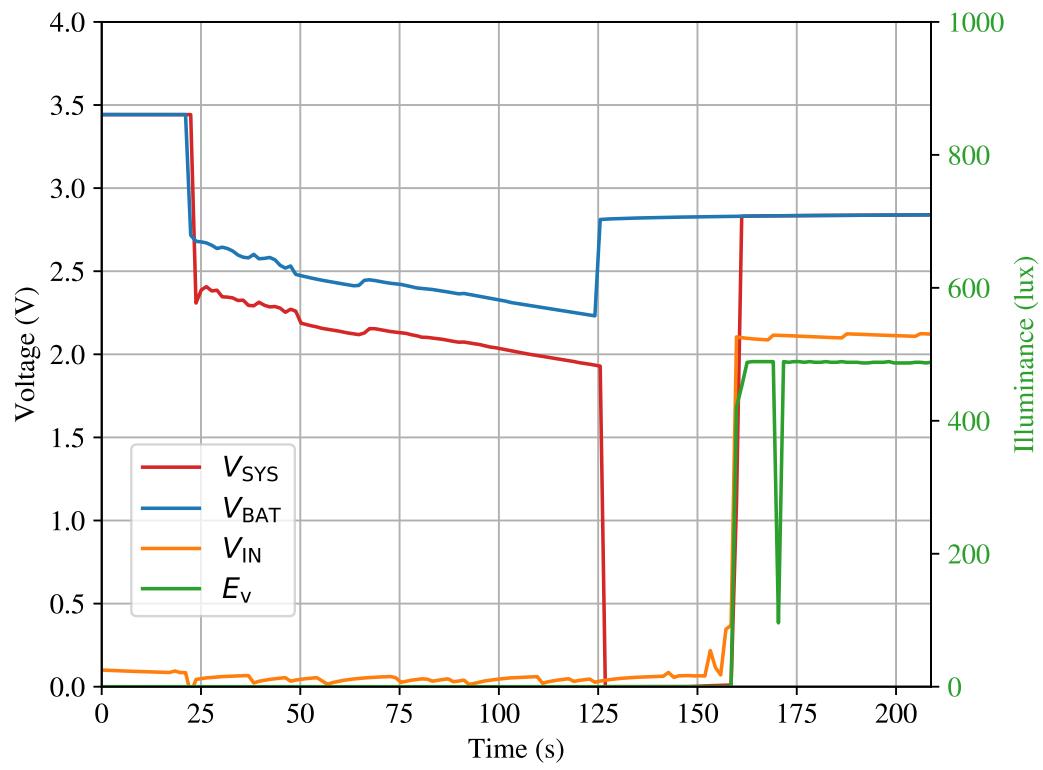


Figure 4.8: Discharging behaviour.

As soon as the load is connected (after 25 s),  $V_{SYS}$  and  $V_{BAT}$  first drop by almost 1 V and after that steadily decrease. After ca. 100 s,  $V_{BAT}$  reached the value of  $V_{BAT\_SD}$  and the ADP5090 switches the output off ( $V_{SYS}$  drops to 0) to prevent the capacitor from overdischarging. The output now stays switched off, until  $V_{IN}$  again supplies energy, and  $V_{BAT} \leq V_{BAT\_SD}$ . It can also clearly be seen, that after 160 seconds, the ADP5090 controls  $V_{IN}$  to ca. 2.1 V. Recall that this is the optimal power point of the solar cell.

#### 4.1.4 Display

## 4.2 Software

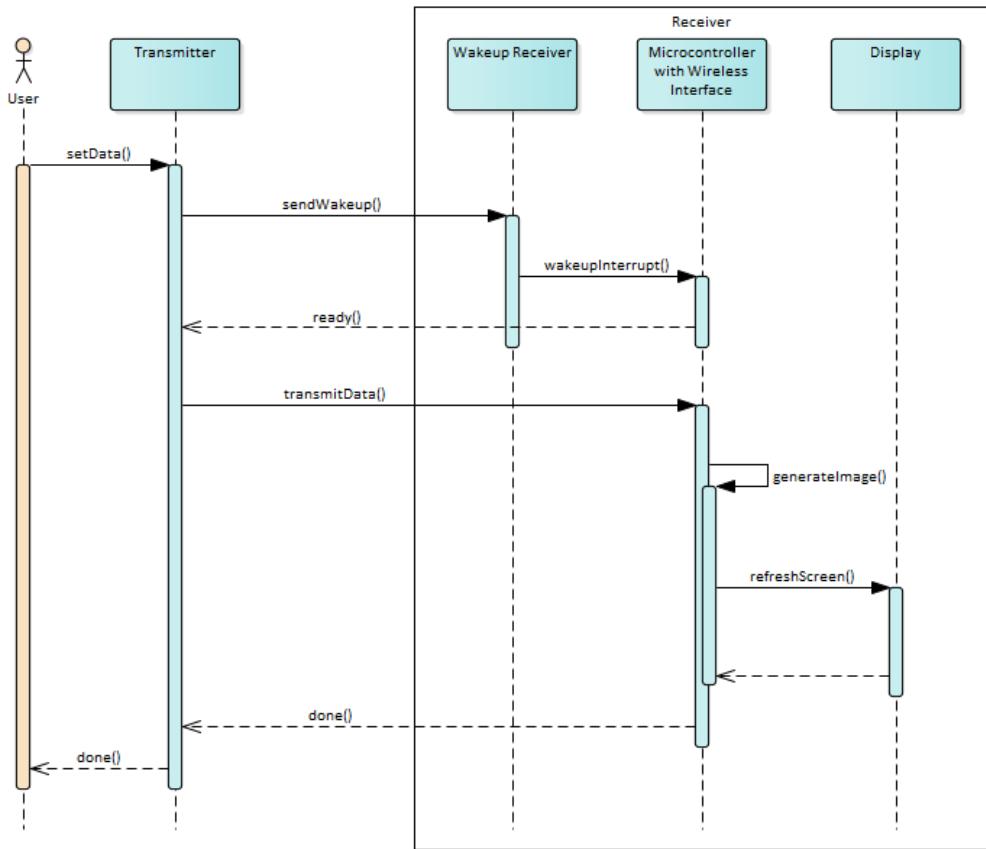
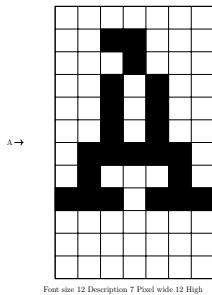


Figure 4.9: not up to date sequence

The software of the ULPWUR is separated in three different parts.



### 4.2.1 Receiver

#### nRF52840

The software on both sides of the bluetooth channel is very similar, thus it is explained in this section. A BLE example project, which was provided by nordic, was only slightly changed to fit the purpose of this prototype.

Before a connection is established, the module on the receiver side is known as peripheral, and the module on transmitter end as the central device. After the receiver is woken up with the

RFicient, the peripheral starts advertising. Advertising basically means sending data packets periodically with information for the central device. The central is scanning for this advertisement and can based on this information decide, if it wants to connect. Is the connection established, it is now possible to exchange data through this channel. This process is illustrated in Figure 4.10.

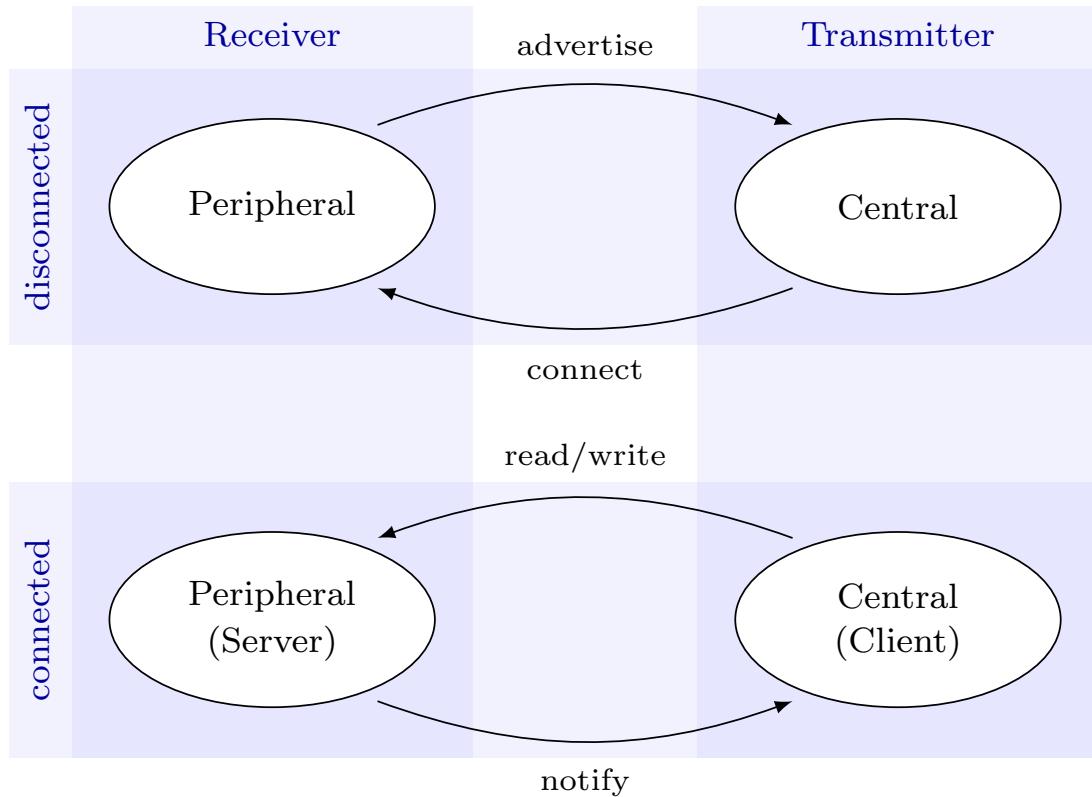


Figure 4.10: BLE-communication with the nRF52480.

The central serves after connection as the client. It can make read and write requests to access the data on the peripheral, which acts as the server. The server on the other hand can only send notifications to the client if new data is ready. It should be noted, that the roles of client and server do not in any case have to be assigned to central and peripheral, but also the other way around.

## 4.2.2 Transmitter

### Python-Script

A short python script enables the user to send the data to the display. This script is looked at here step by step.

1. Include of the used packages, time and serial.

```
import serial
import time
```

2. Set desired baudrate, select which COM-port windows assigned to the development kit and open the port.

---

```
baudrate = 115200
com_port = 'COM14'

device = serial.Serial(com_port, baudrate, writeTimeout=0)
```

3. Create a list of list with the desired data. The first list contains information about how many packages are about to follow. The following list contain the location on the template, the subject and the lecturer.

```
data = [[0, chr(3), '\0\0\0\0'], [11, 'WsComm\0', 'MAH\0'],
[3, 'DigPro\0', 'SCU\0'], [22, 'EmbSW\0', 'BON\0']]
```

4. Fill the list with '\0' so every package is 20 bytes long.

```
for i in range(len(data)):
    if len(data[i][1])<16:
        t = 14-len(data[i][1])
        data[i][1] = data[i][1]+\0*t
```

5. Store time when data transfer is started.

```
t_start = time.time()
```

6. Transfer data to the nRF52480 transmitter

```
for i in range(len(data)):
    device.write([data[i][0]])
    device.write(bytes(data[i][1], 'utf-8'))
    device.write(bytes(data[i][2]+'\n', 'utf-8'))
    device.flush()
```

7. Print elapsed time on console and close COM-port.

```
print('elapsed time: {:.3f}'.format(time.time() - t_start))

device.close()
```



# Chapter 5

## Results

### 5.1 Power consumption

To look at the energy consumption in detail, a power analyser was used as the power supply. To carry out measurements, the ADP5090 with the solar cells and super cap was disconnected and the remaining components connected to the power analyser with a fixed voltage of 3.3 V. This way, the voltage and current could be over one write-cycle could be tracked. The result is plotted in Figure 5.1. One write-cycle takes approximately 5.8 seconds. In this interval, the

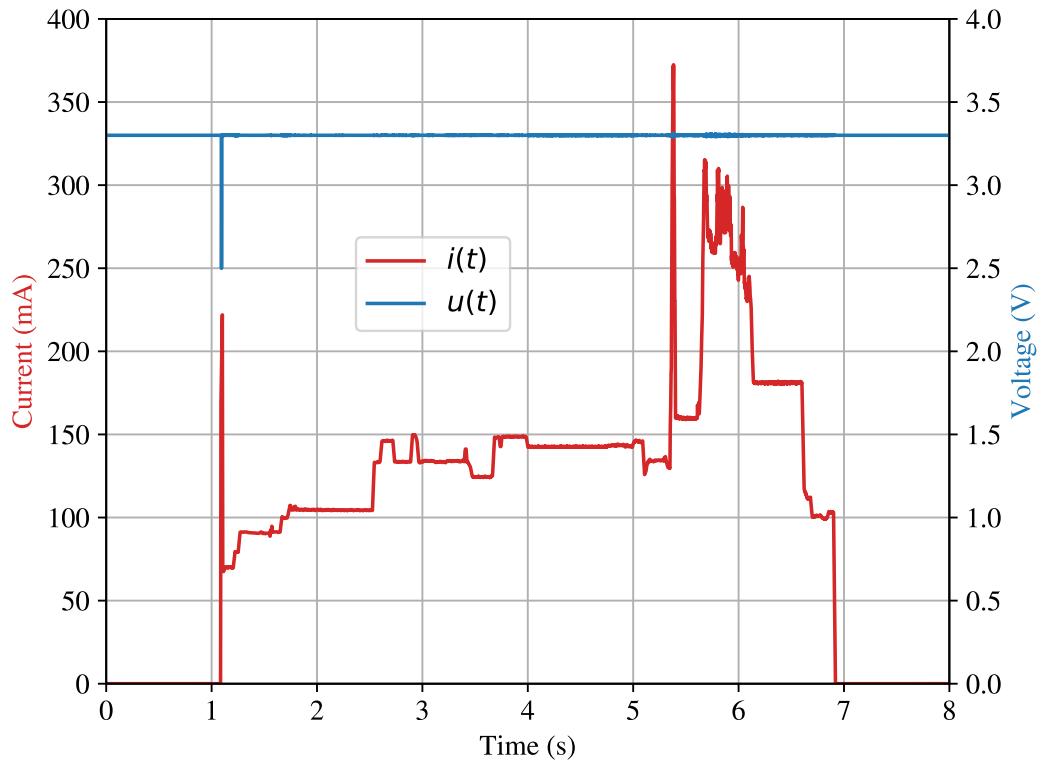


Figure 5.1: Current and Voltage during one write-cycle.

STM32, nRF52480 and display driver are active. Clearly visible is the inrush current peak.

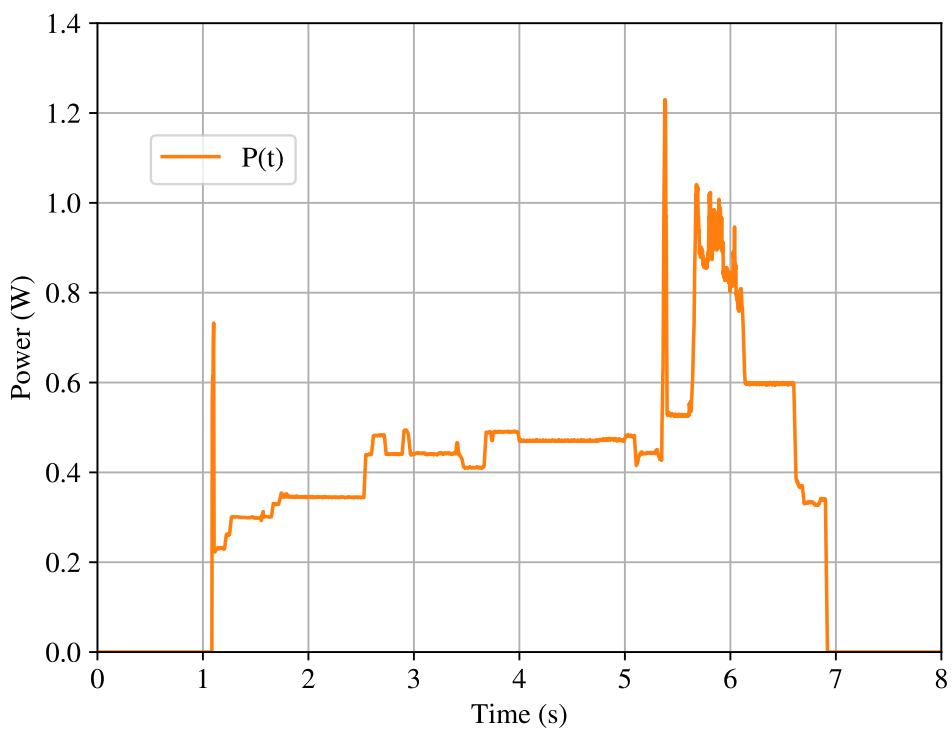


Figure 5.2: Power consumption during one write cycle.

# **Chapter 6**

## **Summary**



# Sources

- [1] K. Amundson, *Electrophoretic Displays*. Springer International Publishing Switzerland, 2016.
- [2] *AS3933 3D Low Frequency Wakeup Receiver*, Mar. 2015.
- [3] *RFicient ULTRA-LOW POWER WAKE-UP RECEIVER DATASHEET (preliminary)*, 2019.
- [4] *MSP430FR698x(1), MSP430FR598x(1) Mixed-Signal Microcontrollers*, Aug. 2018.
- [5] *nRF52480 Product Brief*, 2017.
- [6] *Amorphous Silicon Solar Cells*, Aug. 2019.
- [7] *Ultralow Power Boost Regulator with MPPT and Charge Management*, Feb. 2017.
- [8] *EVAL-ADP5090 User Guide*, 2014.
- [9] *Cylinder Type Lithium Ion Capacitors*, Aug. 2019.
- [10] *nRF52840 Objective Product Specification v0.5*, Dec. 2016.
- [11] *STM32L4R5xx STM32L4R7xx STM32L4R9xx*, Apr. 2018.



---

# List of Figures

2.1	Capsules between to electrodes (taken from [1] with adjustments).	5
2.2	E-paper display under microscope with 250x magnification.	6
2.3	Microcontroller checks periodically for incoming data (top), additional, constantly listening wakeup receiver (bottom).	7
4.1	Receiver schematics.	13
4.2	Transmitter GUI.	14
4.3	Receiver GUI.	15
4.4	Test environment 1 (HSR building 1).	16
4.5	Test environment 2 (HSR building 8).	17
4.6	Schematics of the test setup.	18
4.7	Charging behaviour.	19
4.8	Discharging behaviour.	20
4.9	not up to date sequence	21
4.10	BLE-communication with the nRF52480.	22
5.1	Current and Voltage during one write-cycle.	25
5.2	Power consumption during one write cycle.	26



---

## **List of Tables**



# **Statement of Plagiarism**

We declare that, apart from properly referenced quotations, this report is our own work and contains no plagiarism; it has not been submitted previously for any other assessed unit on this or other degree courses.

**Place**            **Date**  
Rapperswil   December 17, 2019

## **Signatures**

Cedric Renda

Manuel Tischhauser



# **Appendix A**

## **Requirements**

---

# A.1 Assignment



## Low power wakeup receiver

Semesterarbeit für Manuel Tischhauser und Cédric Renda

Herbst 2019

### 1. Einführung

Im Gebäudemanagement ist es üblich, Belegungspläne an den Eingängen der Räume anzubringen. Oftmals sind diese in Papierformat und müssen bei einer Änderung von Hand gewechselt werden. Mit dieser Methode werden kurzfristige Belegungen nicht aufgezeigt. Dies könnte man umgehen, wenn man mit Displays arbeitet die Wireless aktualisiert werden können. Dabei stellt sich allerdings das Problem, dass man entweder Kabel für die Netzeinspeisung verlegen muss oder Batterien verwendet, die regelmässig ersetzt werden müssen. Im Idealfall entfällt die Speisung komplett.

### 2. Aufgabenstellung

Zu einem Empfängermodul soll eine bidirektionale low power Kommunikationsstrecke aufgebaut werden. Der Empfänger soll durch Energy-harvesting Massnahmen möglichst passiv betrieben werden können. Dieser enthält einen Ultra Low Power Wake-up Receiver und eine Anzeige mit dem die empfangenen Daten auf entsprechende Weise dargestellt werden. Ein System kann aus mehreren Empfängern bestehen, welche unabhängig voneinander vom Sender selektiert werden können.

### 3. Ablauf

Zu Beginn der Arbeit sind ein Projektplan und ein Pflichtenheft zu erstellen, welche in den ersten Wochen dem Betreuer abgegeben werden müssen. Ein Vorschlag des Pflichtenheftes befindet sich im Arbeitsplatzordner oder als \*.docx File auf dem Public Server ([\\hsr.ch\\root\\auw\\sge\\labors\\Mk\\pub\\_for\\_students](\\hsr.ch\\root\\auw\\sge\\labors\\Mk\\pub_for_students)). Planen Sie total 240 Arbeitsstunden (8 ECTS \* 30 h/ECTS) ein. Die Arbeiten sollen innerhalb der Gruppe geeignet aufgeteilt werden; die Aufteilung ist im Bericht entsprechend festzuhalten, genauso wie ein Vergleich des geplanten und des effektiv durchgeföhrten Projektplans. Weitere Einzelheiten werden an den wöchentlichen Besprechungen festgelegt. Die Arbeiten sollen möglichst selbstständig durchgeführt werden. Die Kriterien der Beurteilung und Notengebung sind im unten erwähnten Leitfaden zu finden.

### 4. Laborjournal

Während der Arbeit ist ein persönliches (d.h. pro Person eines), gebundenes, handschriftliches und datiertes Laborjournal zu führen. Darin werden alle Tätigkeiten betreffend Dauer und Resultate eingetragen. Ebenfalls soll darin ein Protokoll geführt werden von den wöchentlichen Treffen. Das Laborjournal wird am Ende der Arbeit abgegeben und wird mitbenotet.

### 5. Bericht

Über die Arbeit ist ein Bericht zu verfassen, dessen Textteil maximal 60 Seiten umfassen und eine Dateigrösse von 5MB nicht überschreiten soll. Im Bericht sollen alle gemachten Überlegungen, Abklärungen, Berechnungen und Untersuchungen detailliert (in Text und Bild) dokumentiert werden. Der Bericht muss gut leserlich geschrieben und übersichtlich gegliedert sein. Weitere Richtlinien, wie ein Bericht aufgebaut sein kann, und weitere nützliche Informationen findet man im Leitfaden, welcher in gedruckter Version im Arbeitsplatzordner und auf dem Public Server abgelegt ist.

Des Weiteren muss im Bericht unbedingt eine unterschriebene Nicht-Plagiatsklärung enthalten sein, ein Beispiel dieser Erklärung befindet sich auf dem Public Server.

Der Bericht ist in 1 Papier-Exemplar abzugeben, mit einer beiliegenden CD-ROM, auf der alle anfallenden Daten, wie auch der Bericht selbst (im PDF-Format) gespeichert sind.

---

## **Semesterarbeit**

---

Writing in English is highly encouraged.

### **6. Termine**

Beginn der Arbeit:

Abgabe des Berichts:

Mündliche Präsentation:

### **7. Organisatorisches**

Betreuung der Arbeit:

Betreuung des Labors:

Arbeitsplatz:

Industriepartner:

Besprechungen: wöchentlich, nach Vereinbarung, an der HSR

Examinator: Prof. Dr. Heinz Mathis, [hmathis@hsr.ch](mailto:hmathis@hsr.ch)

Rapperswil, [Datum]

Viel Erfolg wünscht Ihnen

Heinz Mathis,  
Dozent Mobilkommunikation

---

## A.2 Requirement Specification



### Pflichtenheft

Projekt: Low power wakeup receiver

Version 0.1

Cédric Renda, Manuel Tischhauser

Name	Datum	Unterschrift
Prof. Dr. Heinz Mathis		
Selina Malacarne		
Cédric Renda		
Manuel Tischhauser		

---

27. September 2019

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
<b>2 Auftrag</b>	<b>3</b>
<b>3 Produktanforderungen</b>	<b>3</b>
3.1 Hardware . . . . .	4
3.2 Software . . . . .	4
3.3 Varianten/Optionen . . . . .	4
3.4 Dokumentation . . . . .	4
<b>4 Zeitplan</b>	<b>5</b>

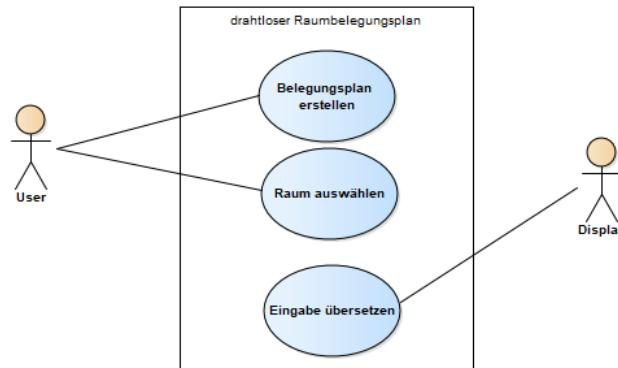
## 1 Einleitung

Im Gebäudemanagement ist es üblich, Belegungspläne an den Eingängen der Räume anzubringen. Oftmals sind diese in Papierformat und müssen bei einer Änderung von Hand gewechselt werden. Mit dieser Methode werden kurzfristige Belegungen nicht aufgezeigt. Dies könnte man umgehen, wenn man mit Displays arbeitet, die über eine drahtlose Schnittstelle aktualisiert werden können. Dabei stellt sich allerdings das Problem, dass man entweder Kabel für die Netzeinspeisung verlegen muss, oder Batterien verwendet, die regelmäßig ersetzt werden müssen. Im Idealfall entfällt die Speisung komplett.

## 2 Auftrag

Im Rahmen dieser Semesterarbeit soll eine Lösung zur oben beschriebenen Problematik ausgearbeitet werden. Der Fokus liegt auf der Erstellung eines autarken Anzeigesystems welches über eine drahtlose Schnittstelle bedient werden kann. Die folgende Punkte sollen dabei abgearbeitet werden:

- Recherche bezüglich Schnittstelle und Energy Harvesting.
- Vor- und Nachteile bestehender Technologien abwägen und geeignete Hardware wählen.
- Erstellen eines lauffähigen Prototypen.



*Abbildung 2.1: Use-Case Diagramm*

Der Prototyp richtet sich nach dem Use-Case Diagramm in Abbildung 2.1.

## 3 Produktanforderungen

Das Empfängermodul ist autark und kann auf einem Display Stundenpläne anzeigen. Diese werden über eine drahtlose, bidirektionale Schnittstelle gesendet. Der Sender wird mit einem Computer bedient.

Besteht das System aus mehreren Empfängern, so kann das Sendemodul diese unabhängig von einander selektieren.

### 3.1 Hardware

#### Sender

- Schnittelle zum Computer
- Sendemodul

#### Empfänger

- Mikrocontroller oder Vergleichbares (Prozessor, Speicher, usw.)
- E-Paper-Display
- Energy-Harvesting-Einheit
- Energiespeicher
- Empfangsmodul

### 3.2 Software

#### Sender

- Treiber für Sendemodul

#### Empfänger

- Firmware für Mikrocontroller

### 3.3 Varianten/Optionen

Ist der Prototyp funktionsfähig, soll zu einem späteren Zeitpunkt auch möglich sein, verschiedene Bildschirmgrößen zu verwenden, wobei sich auch Anzeige nicht nur auf Raumbelegungspläne beschränkt. Deshalb soll das System und insbesondere die Software so flexibel wie möglich entwickelt werden.

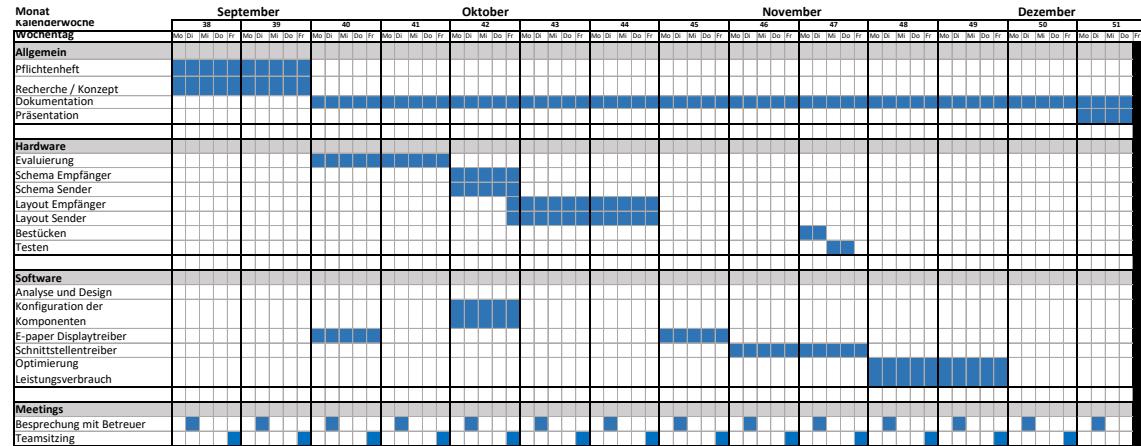
### 3.4 Dokumentation

Die Dokumentation beinhaltet sämtliche Überlegungen, Abklärungen, Berechnungen und Untersuchungen, welche im Laufe der Semesterarbeit gemacht wurden.

## 4 Zeitplan

---

© HSR Hochschule für Technik Rapperswil 2019



Pflichtenheft

Low power wakeup receiver