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 18, 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 $4.5\,\mu$W during listening mode while the system harvests around $485,52\,\mu$W.

Contents

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 of 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. Is such an event 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.

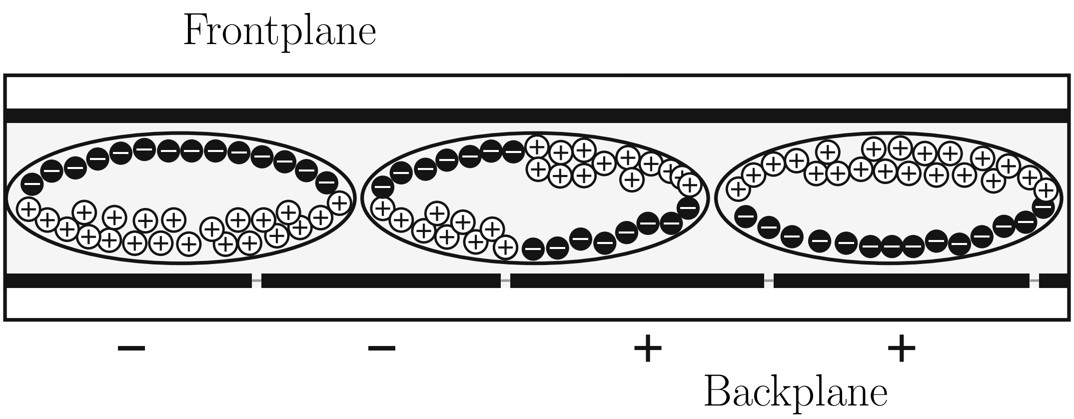


Figure 2.1: Capsules between to electrodes (taken from **amundson** with adjustments).

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 thought the fluid when exposed two 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 of 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 **amundson**.

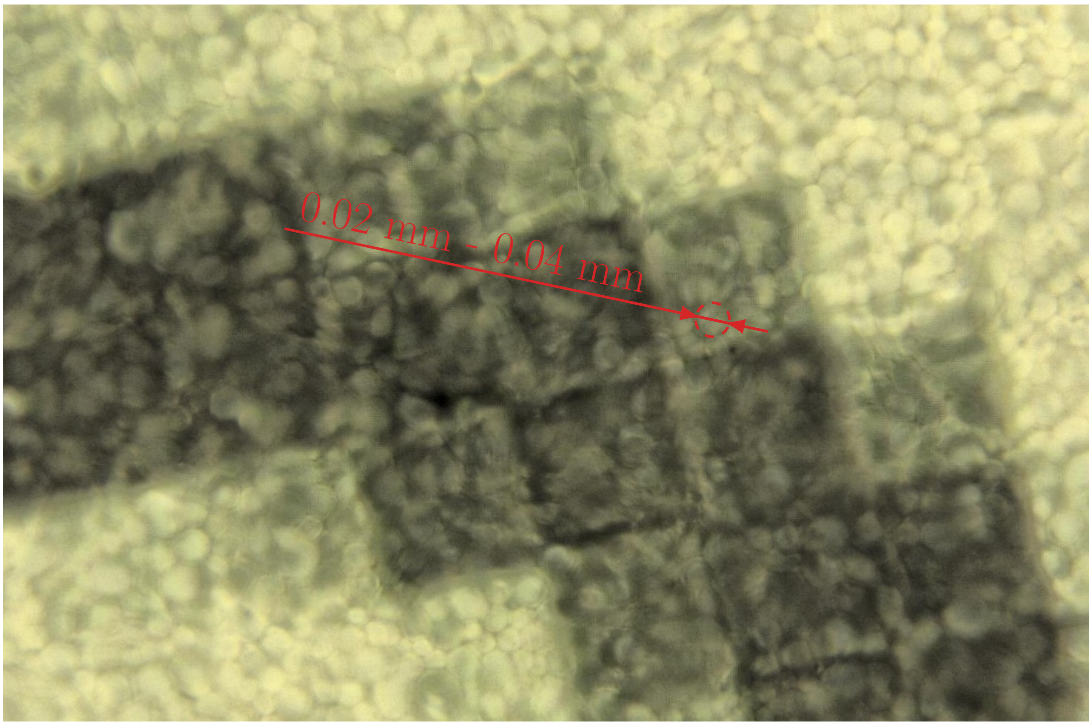


Figure 2.2: E-paper display under microscope with 250x magnification.

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 safe power with an 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 a 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.

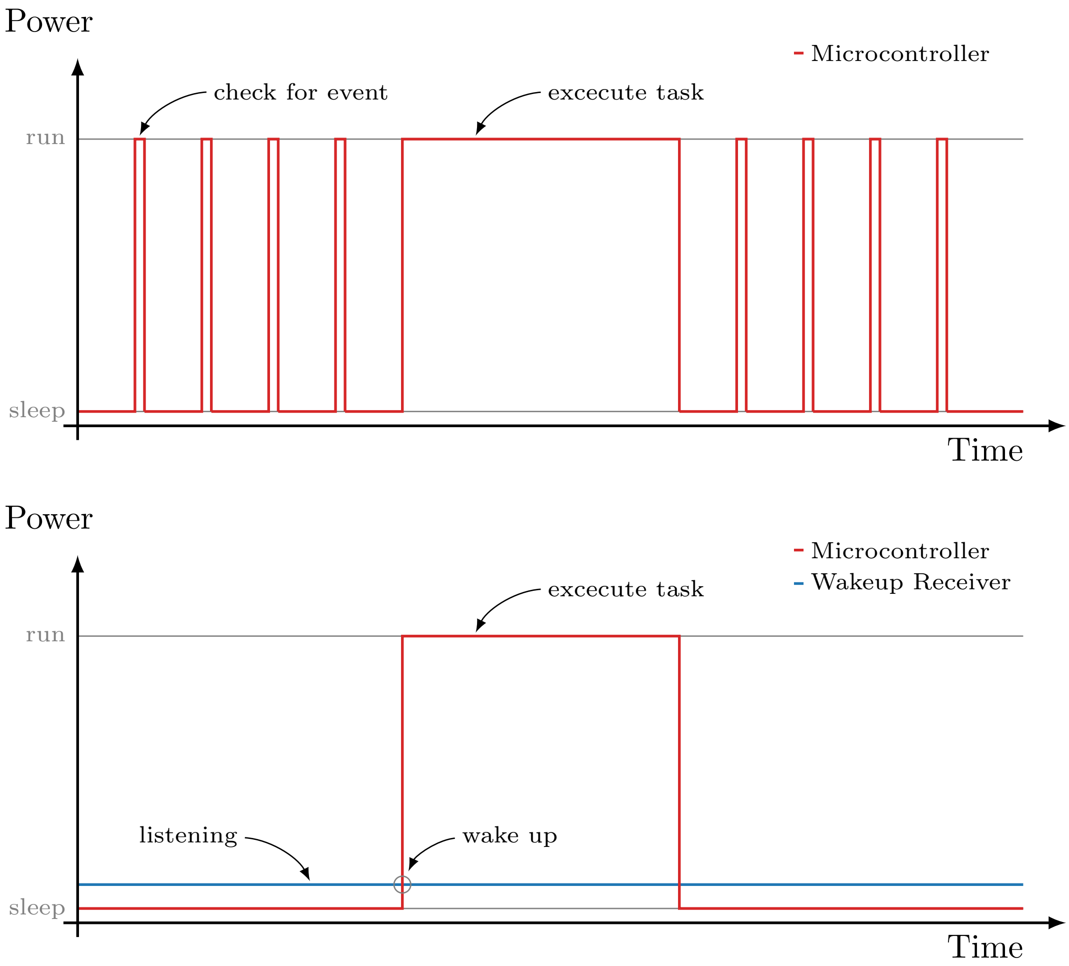


Figure 2.3: Microcontroller checks periodically for incoming data (top), additional, constantly listening wakeup receiver (bottom).

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 Drawing graphics on a graphics display

If an matrix style display is used with a microcontroller, there are a lot of different ways to draw graphical elements. This section covers only the ways used in this project.

Matrix displays

Since most displays produced have an rectangular shape the amount of pixel they hold is calculated like the surface of an rectangular. The number of pixels on both axis are multiplied and we get the total amount of pixels we have to manipulate. So it is obvious that the memory size of an picture explodes with its resolution. As an example we take a display with a pixel ratio of 1200×825 and a colour depth of 24 bit. We get $1200 \cdot 825 = 990'000$ pixel and $1200 \cdot 825 \cdot 24 = 23'760'000$ bits we have to handle. This is a huge amount of data to process with a microcontroller. **etwas über industrie schreiben wiso wir diplay treiber bausteine haben und nicht alles paralell machen**

If an display is rectangular the information of the pixels can be ordered in a matrix. The matrix is normally one or two dimensions to access the data. This matrix can be implemented as an array with one or two dimensions. The definition used for the array management and memory handling is given by the ISO C-Language standard. It specifies with the following sentences how the arrays are defined in the C-Language.

> "An array type describes a contiguously allocated nonempty set of objects with a particular member object type, called the element type. The element type shall be complete whenever the array type is specified. Array types are characterized by their element type and by the number of elements in the array."

ISO/IEC9899

The term allocated is referred as following

> "The order and contiguity of storage allocated by successive calls to the aligned_alloc, calloc, malloc, and realloc functions is unspecified. The pointer returned if the allocation succeeds is suitably aligned so that it may be assigned to a pointer to any type of object with a fundamental alignment requirement and then used to access such an object or an array of such objects in the space allocated (until the space is explicitly deallocated)."

ISO/IEC9899

Alignment is defined with the sentence:

> "**Alignment** requirement that objects of a particular type be located on storage boundaries with addresses that are particular multiples of a byte address."

ISO/IEC9899

and shows the strict handling of the memory mapping in the C-Language. Since the alignment needs to be given for arrays the amount of dimensions of the array have no influence how the information is stored in the memory. In the Figure 2.4 the one dimensional matrix style is shown. To access this kind of display the address of the 176-pixels reaches from 0 to 175 this is display can be addressed with

$$n = \lfloor \log_2(176) \rfloor = 8$$

bit.

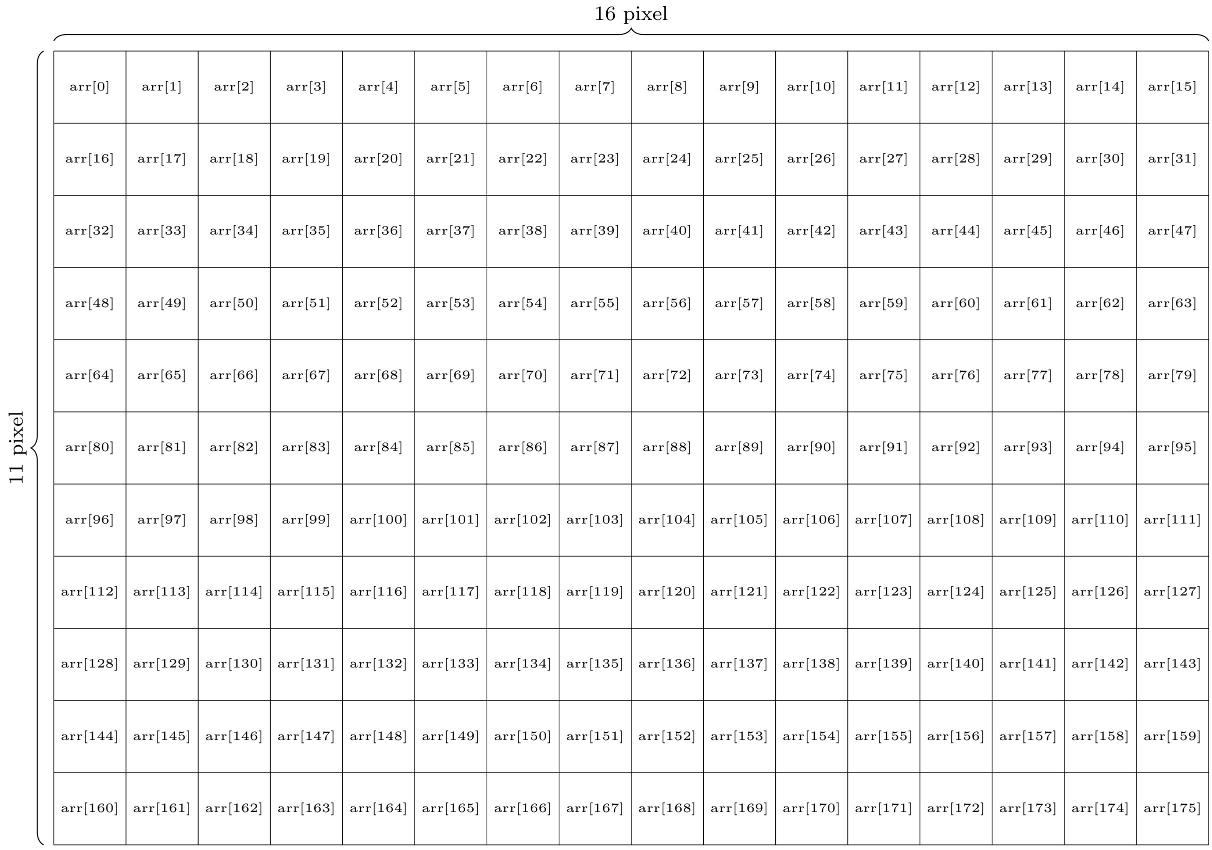


Figure 2.4: 16×11 pixel display with a flat matrix

The

16 pixel

11 pixel

arr[0][0]	arr[0][1]	arr[0][2]	arr[0][3]	arr[0][4]	arr[0][5]	arr[0][6]	arr[0][7]	arr[0][8]	arr[0][9]	arr[0][10]	arr[0][11]	arr[0][12]	arr[0][13]	arr[0][14]	arr[0][15]
arr[1][0]	arr[1][1]	arr[1][2]	arr[1][3]	arr[1][4]	arr[1][5]	arr[1][6]	arr[1][7]	arr[1][8]	arr[1][9]	arr[1][10]	arr[1][11]	arr[1][12]	arr[1][13]	arr[1][14]	arr[1][15]
arr[2][0]	arr[2][1]	arr[2][2]	arr[2][3]	arr[2][4]	arr[2][5]	arr[2][6]	arr[2][7]	arr[2][8]	arr[2][9]	arr[2][10]	arr[2][11]	arr[2][12]	arr[2][13]	arr[2][14]	arr[2][15]
arr[3][0]	arr[3][1]	arr[3][2]	arr[3][3]	arr[3][4]	arr[3][5]	arr[3][6]	arr[3][7]	arr[3][8]	arr[3][9]	arr[3][10]	arr[3][11]	arr[3][12]	arr[3][13]	arr[3][14]	arr[3][15]
arr[4][0]	arr[4][1]	arr[4][2]	arr[4][3]	arr[4][4]	arr[4][5]	arr[4][6]	arr[4][7]	arr[4][8]	arr[4][9]	arr[4][10]	arr[4][11]	arr[4][12]	arr[4][13]	arr[4][14]	arr[4][15]
arr[5][0]	arr[5][1]	arr[5][2]	arr[5][3]	arr[5][4]	arr[5][5]	arr[5][6]	arr[5][7]	arr[5][8]	arr[5][9]	arr[5][10]	arr[5][11]	arr[5][12]	arr[5][13]	arr[5][14]	arr[5][15]
arr[6][0]	arr[6][1]	arr[6][2]	arr[6][3]	arr[6][4]	arr[6][5]	arr[6][6]	arr[6][7]	arr[6][8]	arr[6][9]	arr[6][10]	arr[6][11]	arr[6][12]	arr[6][13]	arr[6][14]	arr[6][15]
arr[7][0]	arr[7][1]	arr[7][2]	arr[7][3]	arr[7][4]	arr[7][5]	arr[7][6]	arr[7][7]	arr[7][8]	arr[7][9]	arr[7][10]	arr[7][11]	arr[7][12]	arr[7][13]	arr[7][14]	arr[7][15]
arr[8][0]	arr[8][1]	arr[8][2]	arr[8][3]	arr[8][4]	arr[8][5]	arr[8][6]	arr[8][7]	arr[8][8]	arr[8][9]	arr[8][10]	arr[8][11]	arr[8][12]	arr[8][13]	arr[8][14]	arr[8][15]
arr[9][0]	arr[9][1]	arr[9][2]	arr[9][3]	arr[9][4]	arr[9][5]	arr[9][6]	arr[9][7]	arr[9][8]	arr[9][9]	arr[9][10]	arr[9][11]	arr[9][12]	arr[9][13]	arr[9][14]	arr[9][15]
arr[10][0]	arr[10][1]	arr[10][2]	arr[10][3]	arr[10][4]	arr[10][5]	arr[10][6]	arr[10][7]	arr[10][8]	arr[10][9]	arr[10][10]	arr[10][11]	arr[10][12]	arr[10][13]	arr[10][14]	arr[10][15]

Figure 2.5: 16×11 display with a two dimensional matrix

Double buffering

To eliminate the

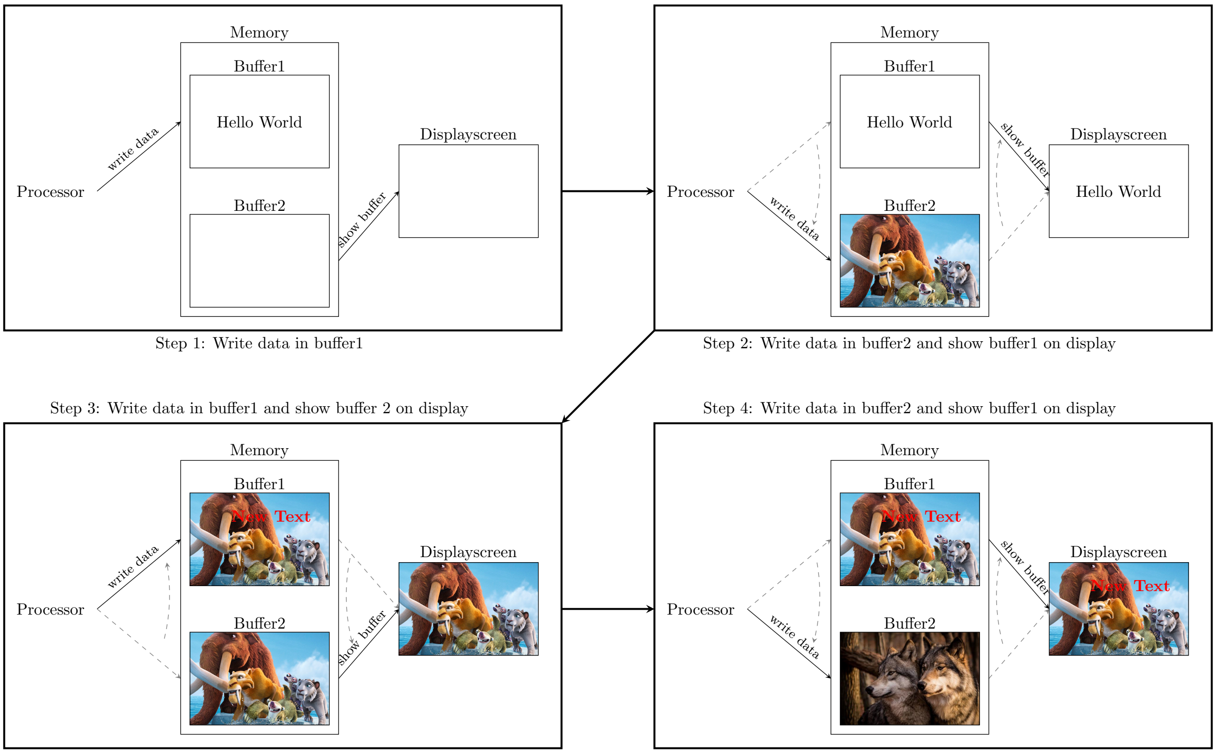


Figure 2.6: Double buffer example while writing different images to screen

Compressed fonts

The theory shown in this chapter 2.3 is only valid if the fonts used are simple fonts which are using only two different values. More complex fonts have to be compressed in a different way to avoid data loss.

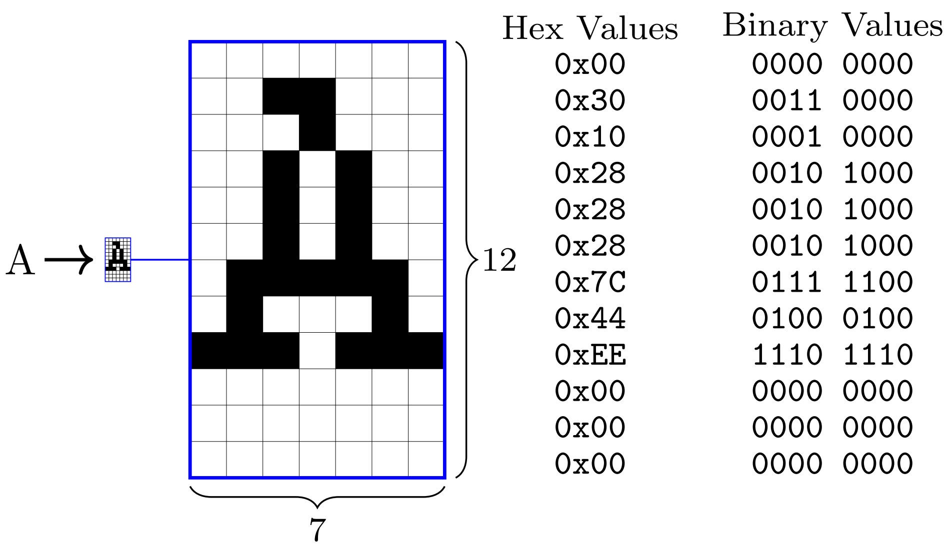
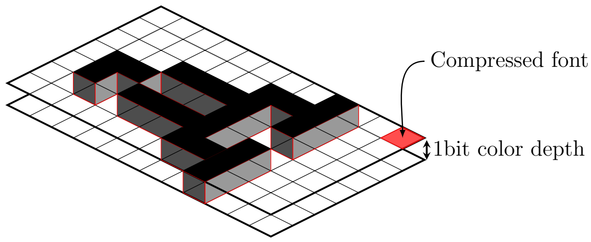


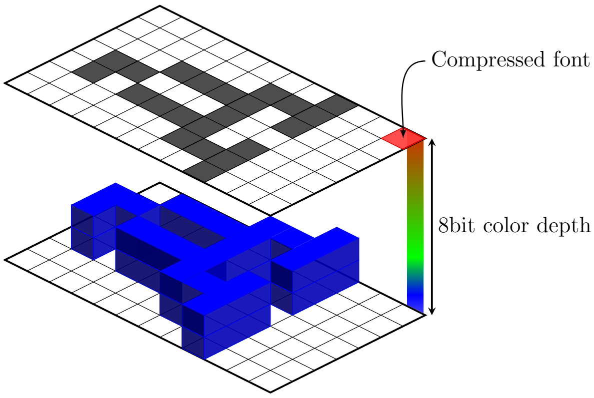
Figure 2.7: Creation of a simple seven to twelve pixel sized font

2.3.1 Drawing text to the display

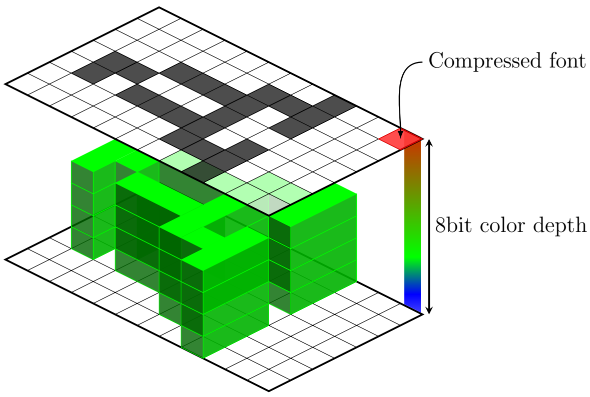
If a displays have an colour depth of eight bit per pixel the host controller needs to write for each pixel an eight bit value to the memory of the pixel.



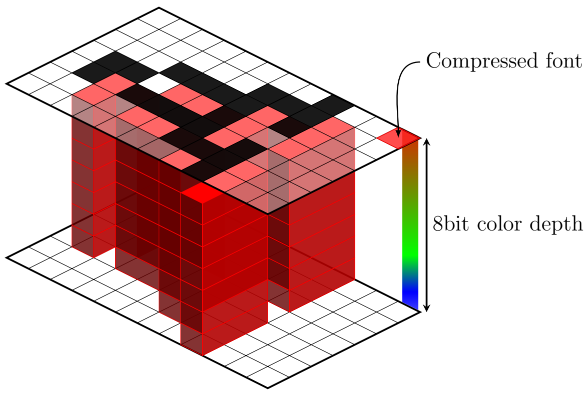
(a) Font mask displayed in one bit color depth



(b) Font mask displayed in eight bit color depth with the value for blue



(c) Font mask displayed in eight bit color depth with the value for green



(d) Font mask displayed in eight bit color depth with the value for red

Figure 2.8: 12×7 pixel font mask displayed with 1- and 8bit colour depth

Fig. 4.9, Fig. **??**, Fig. **??**, Fig. **??**, Fig. **??**

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 μV_{RMS}

- Current consumption in 3-channel listening mode 2.3 μA

- Operating supply range 2.4 V – 3.6 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 μW**as3933**.

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 μA 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 μW.

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 **rficient**. 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. It's main features are:

- 16-bit architecture with up to 16 MHz clock
- Supply voltage range form 1.8 V to 3.6 V
- Approximately 100 μA/MHz current consumption in active mode
- 128 KB Ferroelectric Random Access Memory (FRAM)

The display driver chip can be controlled over SPI **msp430**.

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 × 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 STMircoelectronics 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 μA/MHz current consumption in run mode
- 2 MB Flash and 640 KB SRAM

The display driver can again be controlled over SPI **stm32**.

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 to 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.

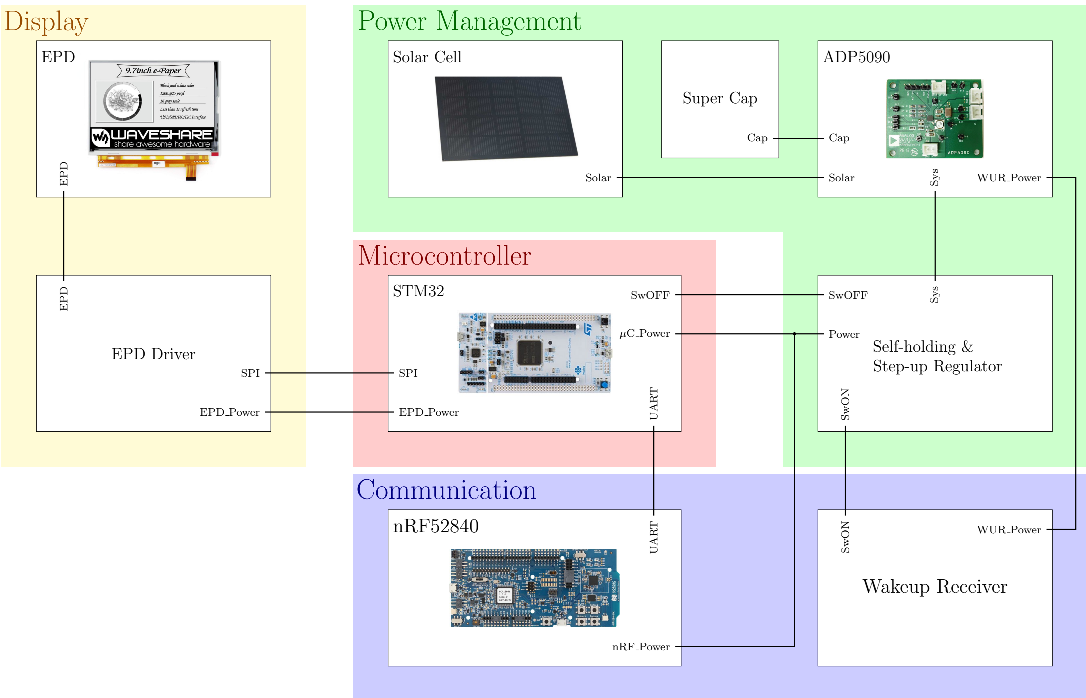


Figure 4.1: Receiver schematics.

4.1.1 Microcontroller

4.1.2 Communication

RFicient

Transmitter The Rficient 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

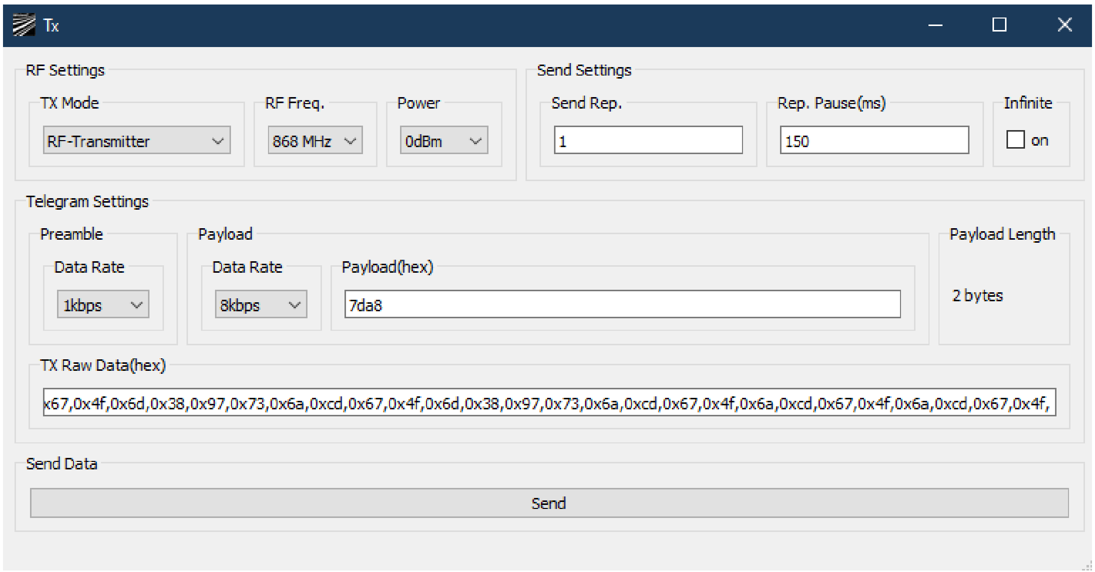


Figure 4.2: Transmitter GUI.

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 if this reason, a second test was performed to check how the RFicient penetrates obstacles. As test environment served this time the HSR builing 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

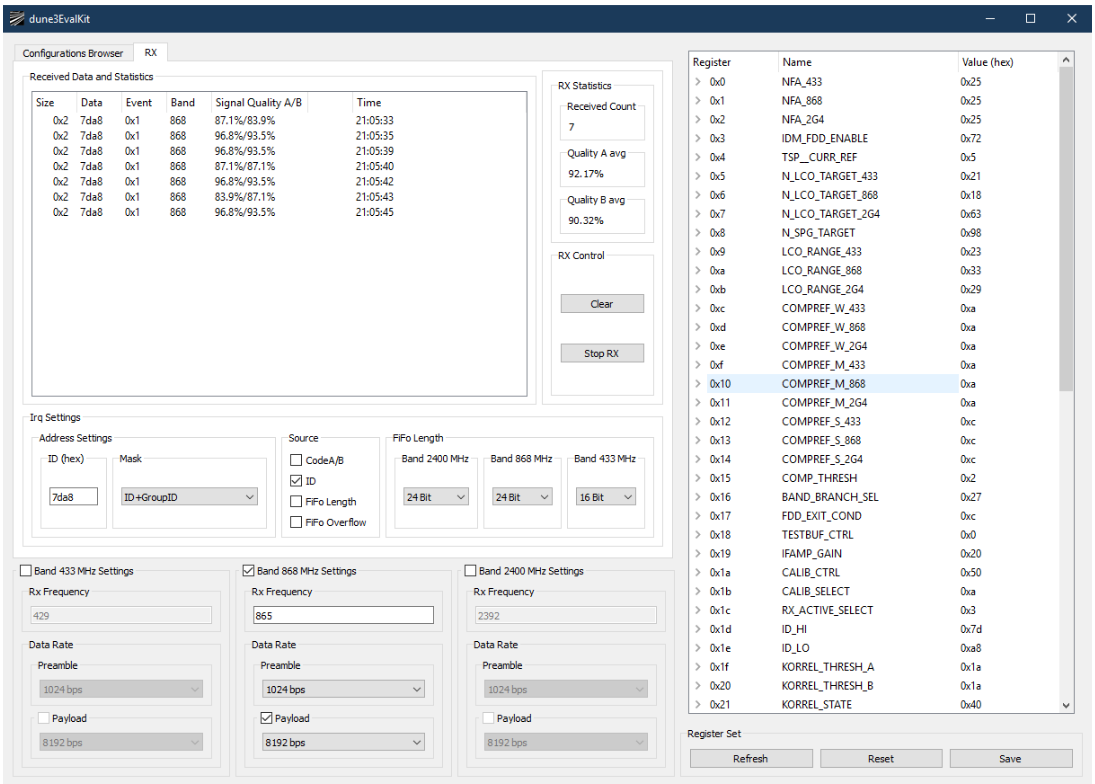


Figure 4.3: Receiver GUI.

walls. With a transmisson 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)**nrfpb**. 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 mm × 40.5 mm and delivers up to 58.7 μA when operating at an optimal voltage of 2.1 V, provided an illuminance of 200 lux. To keep a reasonable display to panel ratio, four cells where connected in parallel, which corresponds to an area of ca. 89.1 cm^2 (Display area = 283 cm^2). Therefore, the solar cells should provide a power of

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

given a 200 lux illuminance **amorton**.

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%
- 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:

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

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

The same procedure is applied to set a minimal voltage:

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

While discharging, the ADP5090 will switch off the output V_{SYS} if $V_{\mathrm{BAT\_SD}}$ 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 ($V_{\mathrm{BAT\_SD}} \leq V_{\mathrm{SYS}} \leq V_{\mathrm{BAT\_TERM}}$) **adp**.

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 **adp_eval**.

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 **yuden**.

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.

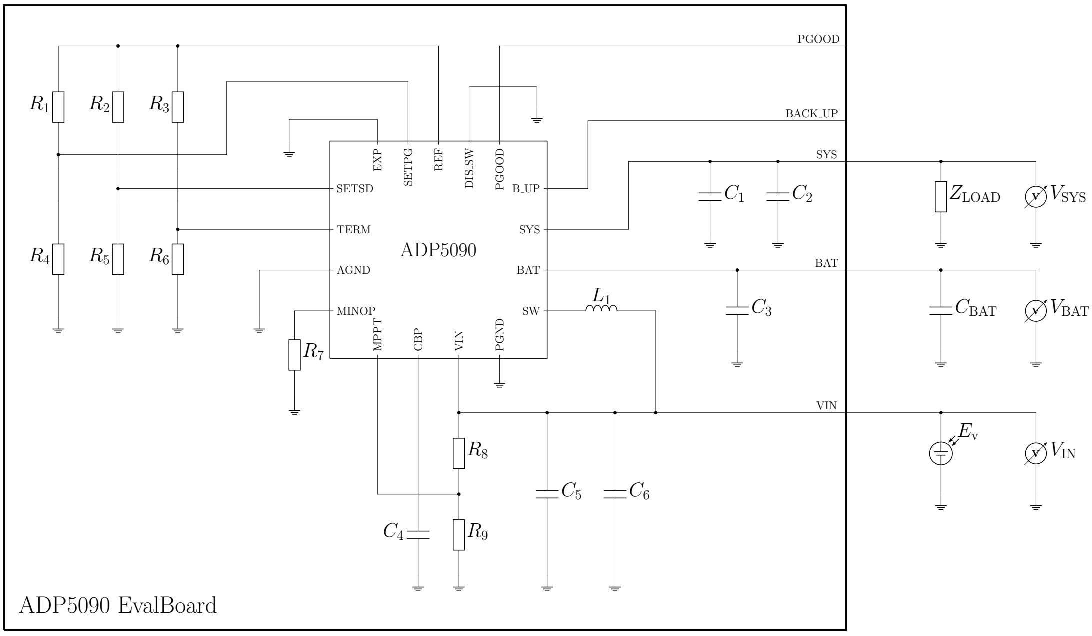


Figure 4.6: Schematics of the test setup.

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 **nrf**. The STM32 on the other side is less flexible with an input voltage range of 1.71 V to 3.6 V **stm32**. 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 $V_{\mathrm{BAT\_TERM}} \leq 3.6\,\mathrm{V}$ and $V_{\mathrm{BAT\_SD}} \geq 2.2\,\mathrm{V}$, to satisfy all of these three elements. In order to do this, the four resistors have to be chosen as $R_3 = 4.3\,\mathrm{M\Omega}$, $R_6 = 4.7\,\mathrm{M\Omega}$, $R_2 = 4.3\,\mathrm{M\Omega}$ and $R_3 = 5.1\,\mathrm{M\Omega}$. Inserted in the equation (4.2) and (4.3) we get

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

and

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

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 $\leq V_{\mathrm{BAT\_TERM}}$. The measurements where taken over a couple hours and are plotted in Figure 4.7.

No load was connected to the output ($Z_{\mathrm{LOAD}} \to \infty$), 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 $V_{\mathrm{BAT\_TERM}}$.

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

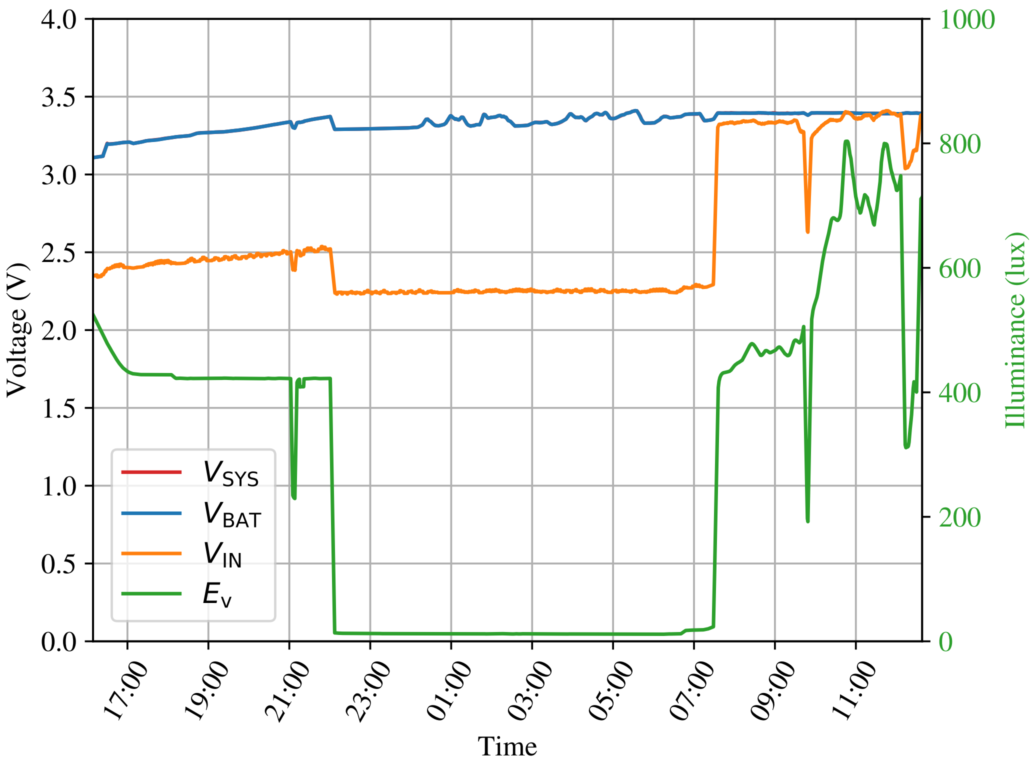


Figure 4.7: Charging behaviour.

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:

$$P_e = 5\,\mathrm{V} \cdot 0.24\,\mathrm{A} + 3\,\mathrm{V} \cdot 0.006\,\mathrm{A} = 1.218\,\mathrm{W}. \tag{4.6}$$

If $Z_{\mathrm{LOAD}} = 10\,\Omega$ 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.

As soon as the load is connected (after 25), 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 $V_{\mathrm{BAT\_SD}}$ 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 $V_{\mathrm{BAT}} \leq V_{\mathrm{BAT\_SD}}$. 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.

Power Latch

To enable and disable the power supply of everything but the ADP5090 and Wakeup receiver

Fig. 4.9, Fig. **??**, Fig. **??**, Fig. **??**, Fig. **??**

4.1.4 Display

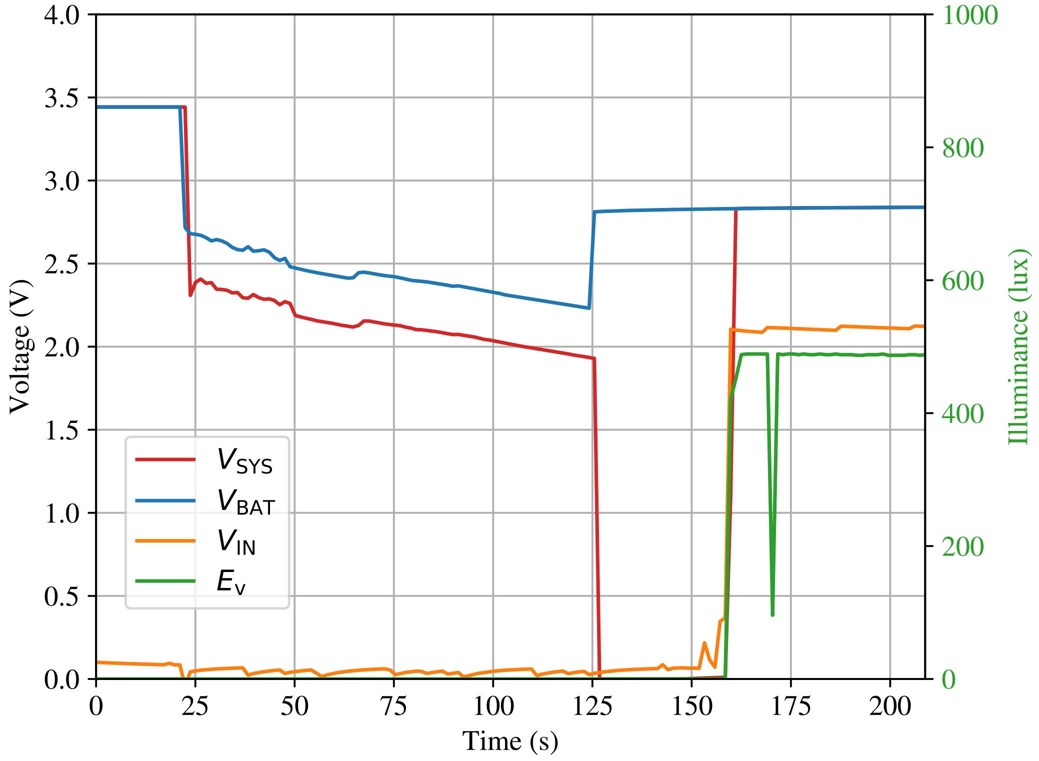


Figure 4.8: Discharging behaviour.

4.2 Software

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

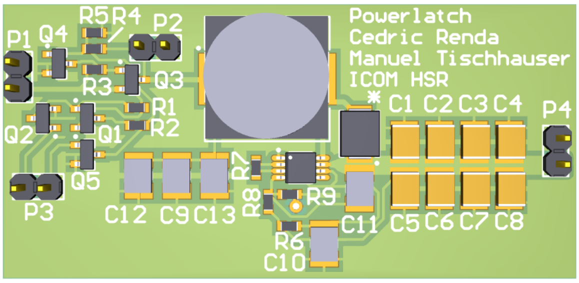
4.2.1 Receiver

nRF52840

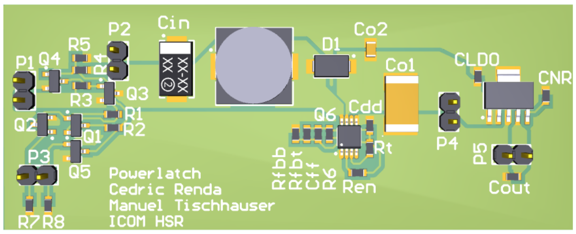
The software on both sides of the bluetooth channel is very similar, thus it is explained in this section. A BLE example project, which was provided by nordic, was only slightly changed to fit the purpose of this prototype.

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 the 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 4.12.

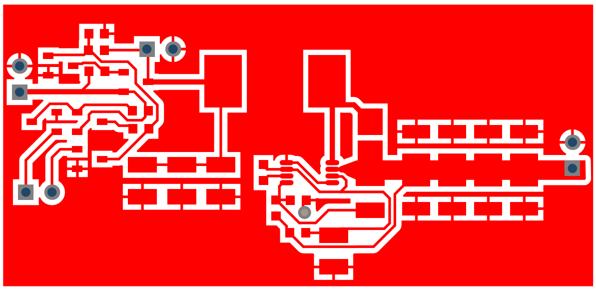
The central serves after connection as the client. It can make read and write requests to access the data on the peripheral, which acts as the server. The server on the other hand can only send notifications to the client if new data is ready. It should be noted, that the roles of client and server do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



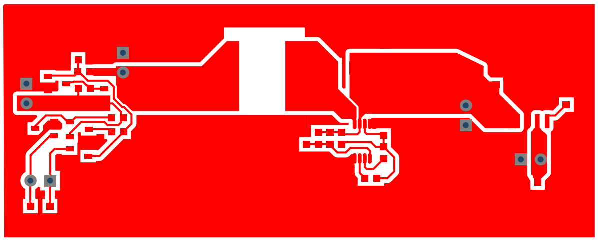
(a) 3D picture of Version one of the power latch



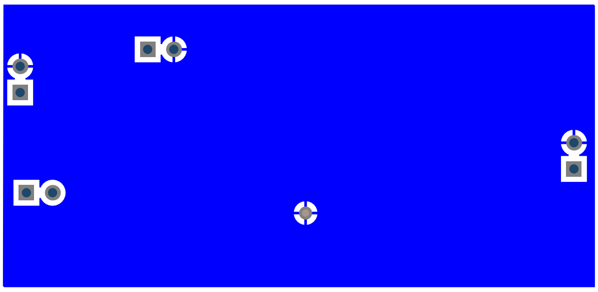
(b) 3D picture of Version two of the power latch



(c) Font mask displayed in eight bit color depth with the value for green



(d) Font mask displayed in eight bit color depth with the value for red



(e) Font mask displayed in eight bit color depth with the value for green

(f) Font mask displayed in eight bit color depth with the value for red

Figure 4.9: 12×7 pixel font mask displayed with 1- and 8bit colour depth

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)
```

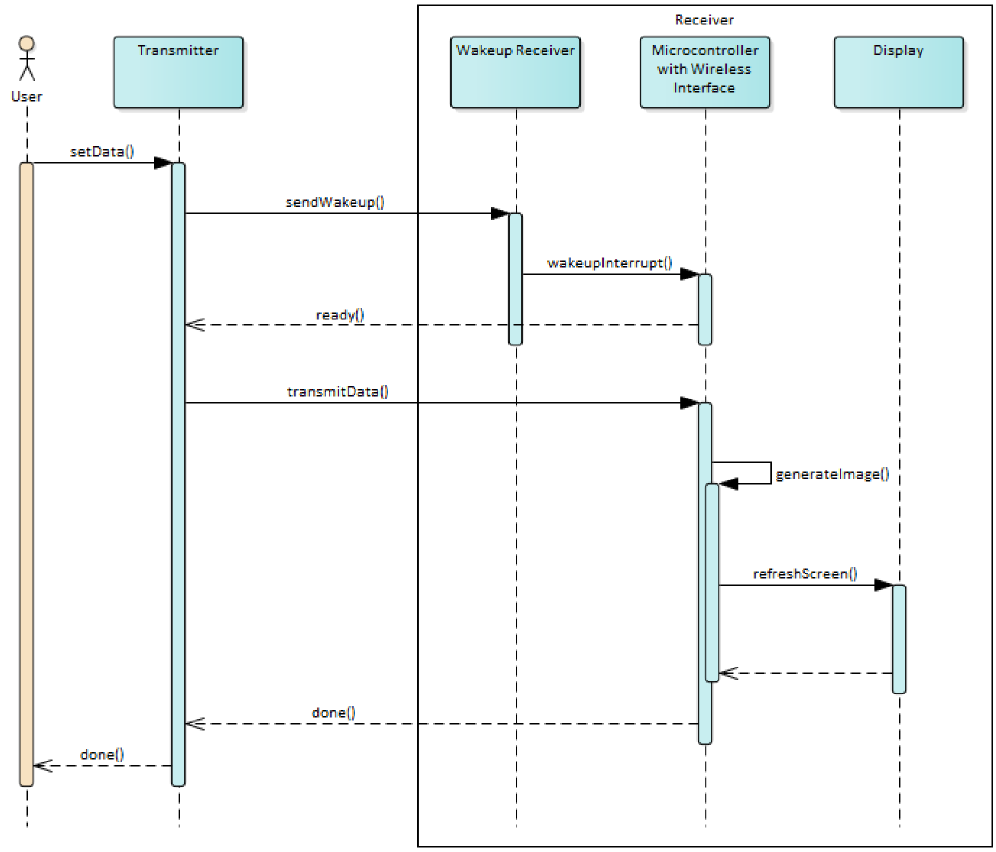


Figure 4.10: not up to date sequence

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()
```

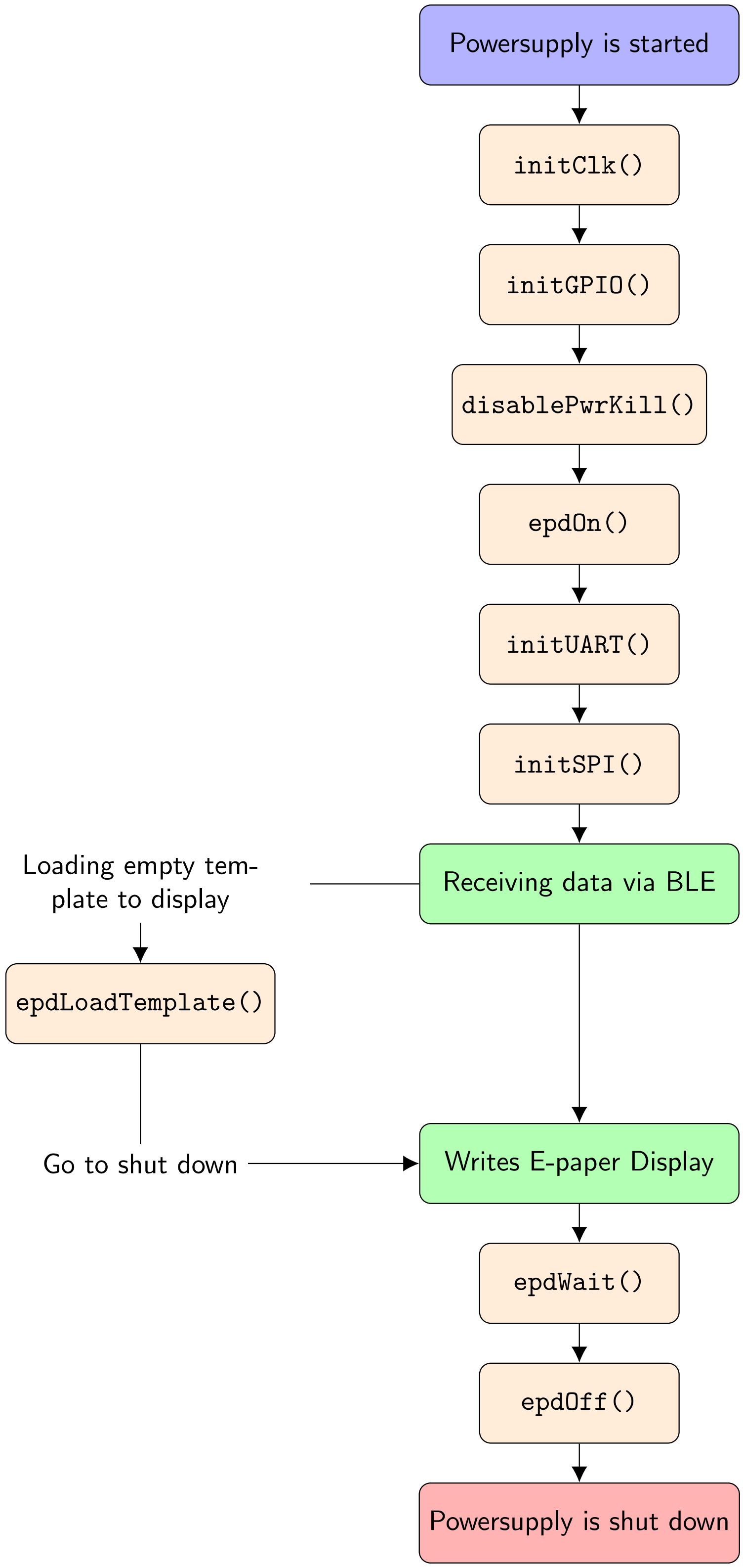


Figure 4.11: BLE-communication with the nRF52480.

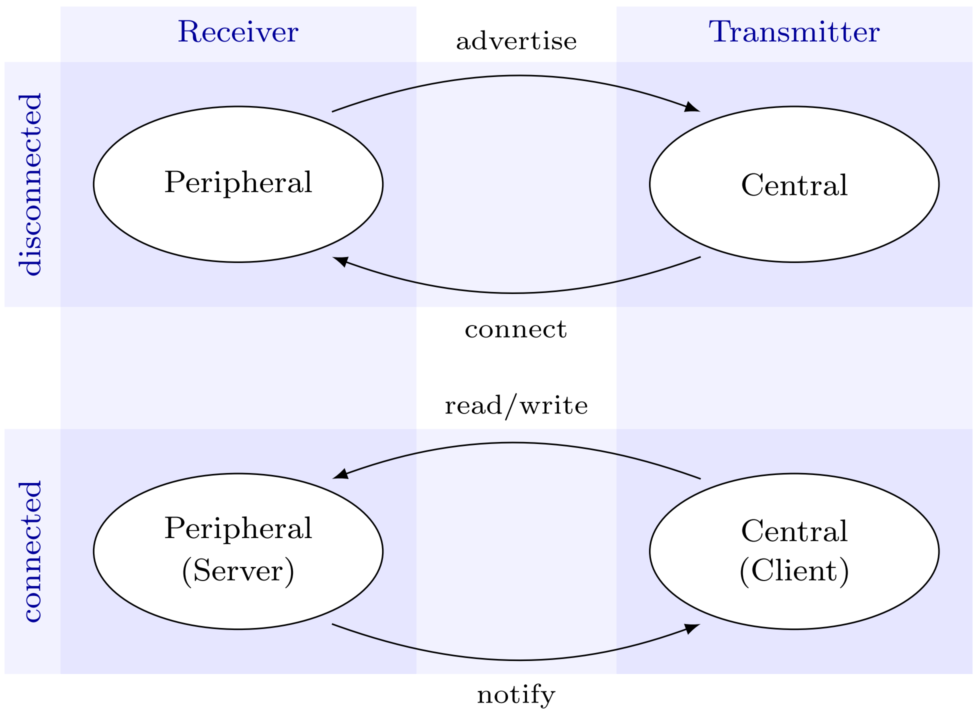


Figure 4.12: BLE-communication with the nRF52480.

Chapter 5

Results

5.1 Power supply

There is a voltage drop over the ADP5090 ($V_{BAT} - V_{SYS}$ in Figure 4.8) if a load is attached to the main output. This drop causes the chip to switch off the output way too early. Out of this reason, it was necessary to bypass the ADP5090 and connect the receiver circuit directly to the capacitor. This way, the capacitor is not protected from overdischarging, but charging of the capacitor is still controlled by the ADP5090.

Figure 5.1 shows a plot of the voltages of the power supply during one write cycle. The

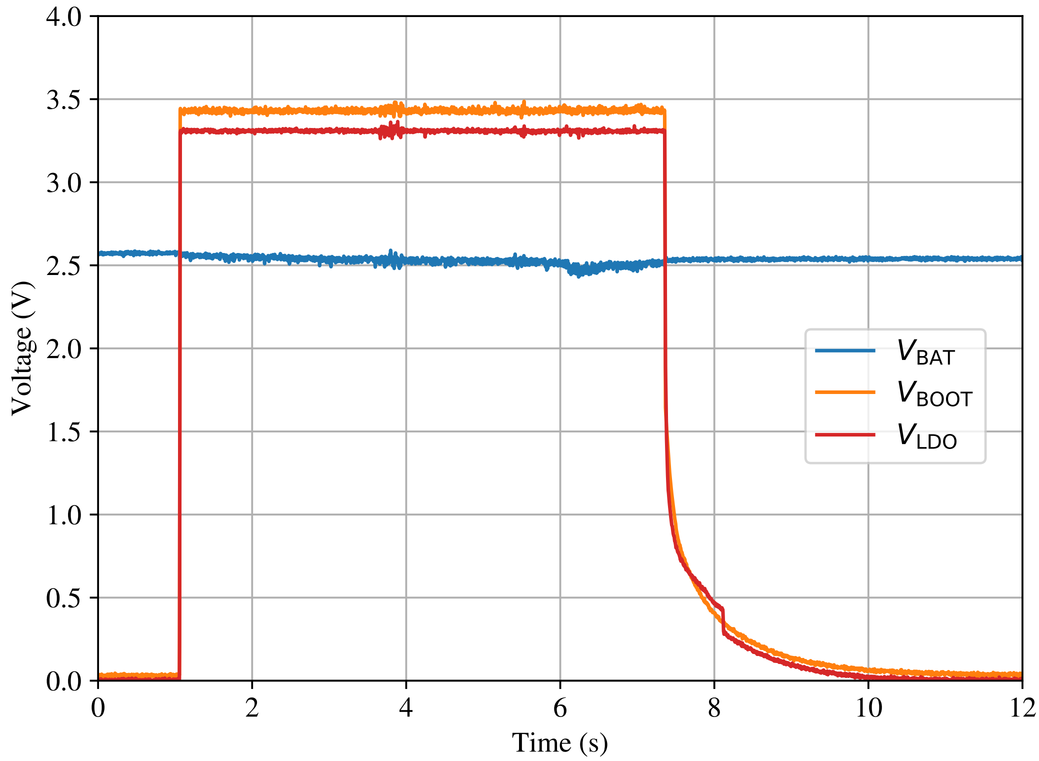


Figure 5.1: Power consumption during one write cycle.

voltage over the capacitor is labelled as V_{BAT}. Additionally, the voltages after the step-up converter (V_{BOOT}) and LDO (V_{LDO}) were tracked. The whole power supply manages to provide a stable voltage over the whole write cycle, despite high current peaks.

5.2 Power consumption

5.2.1 Power measurement

To look at the energy consumption in detail, the N6705B power analyzer from keysight 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 over one write-cycle could be tracked. The result is plotted in Figure 5.2. One write-cycle takes about 5.8 seconds. In this

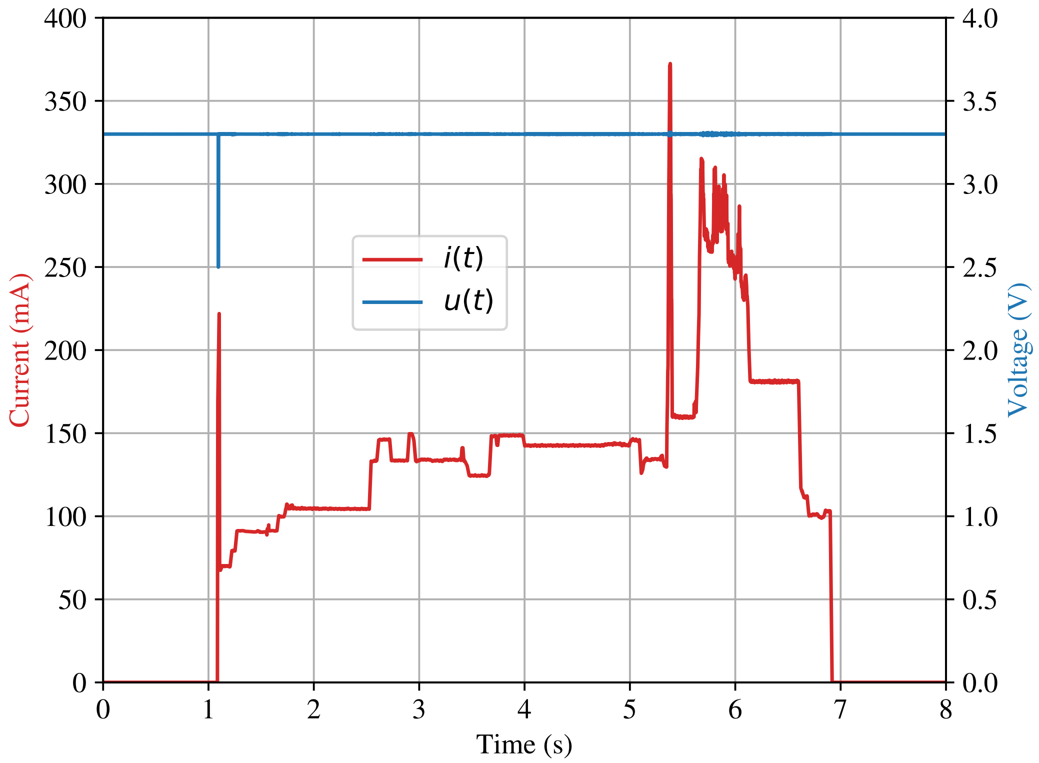


Figure 5.2: Current and Voltage during one write-cycle.

interval, the STM32, nRF52480 and display driver are active. Clearly visible is the current peak which occurs when the system is woken up. Short after the 5 seconds mark, initializing and data receiving is finished and the display is refreshed. The current peaks there are due to the charge pumps on the display driver.

With voltage and current given, the power over time can be calculated. This curve is plotted in Figure 5.3. The power follows obviously the same course like the current since the voltage is constant. The energy needed for one write-cycle is given with the integral of the power over

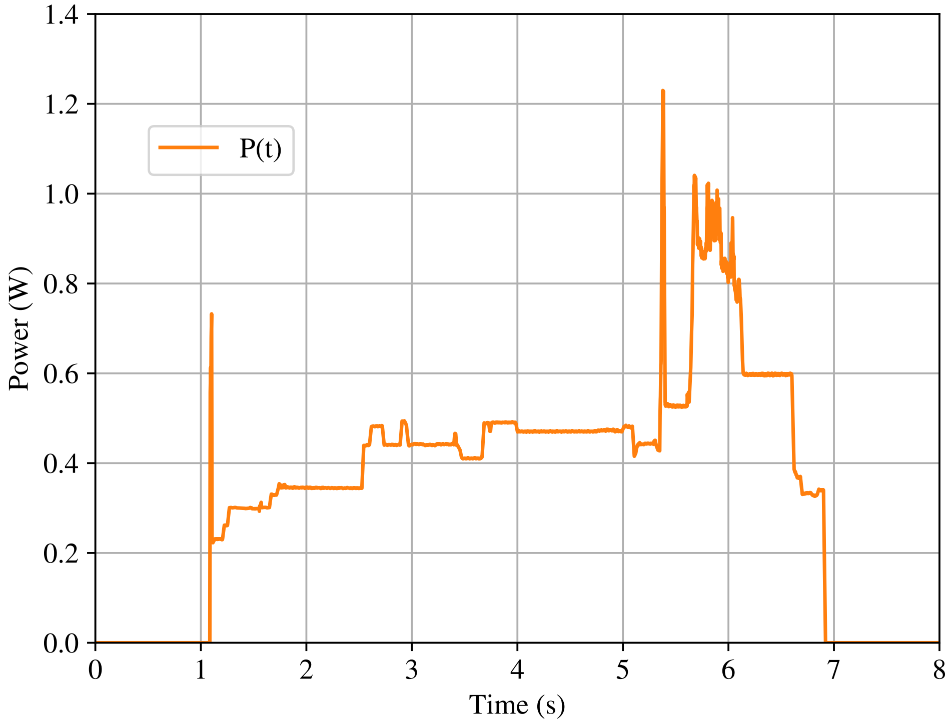


Figure 5.3: Power consumption during one write cycle.

time:

$$E = \int_0^{8\,\mathrm{s}} P(t)\mathrm{d}t \approx 2.7504\,\mathrm{J}. \tag{5.1}$$

Like stated in (4.1) the solar cells provide 485.52 μW at 200 lux. Considering the Rficient which consumes 4.5μW, it is possible to estimate the time needed, to harvest the power for one write cycle:

$$t_\mathrm{E} = \frac{2.7504\,\mathrm{J}}{485.52\,\mu\mathrm{W} - 4.5\mu\mathrm{W}} = 5717.85\,\mathrm{s} \approx 1\,\mathrm{h}\;35\,\mathrm{min}. \tag{5.2}$$

Not taken into account is the leakage of the super capacitor and losses in the ADP5090.

5.2.2 Energy storage

The stored energy in a capacitor is

$$E = \frac{1}{2}CU^2.$$

since the capacitor operates between 2.2 V and 3.6 V, the total stored energy is

$$E_\mathrm{ges} = \frac{1}{2}C(U_\mathrm{max}^2 - U_\mathrm{min}^2) = \frac{1}{2}\cdot 40\,\mathrm{F}\cdot((3.6\,\mathrm{V})^2 - (2.2\,\mathrm{V})^2) = 162.4\,\mathrm{J}. \tag{5.3}$$

If again the loss in the ADP5090 and the leakage of the super cap is neglected, it is possible to estimate the time

$$t_{\text{func}} = \frac{162.4\,\text{J} - 2.7504\,\text{J}}{4.5\,\mu\text{W}} = 3.54877 \cdot 10^{-7}\,\text{s} \approx 411\,\text{d} \tag{5.4}$$

in which the prototype is fully functional when no energy is harvested. The 2.7504 J are subtracted, because on should be able to refresh the display after this time. This result should be taken with caution, because it neglects losses and does not consider the tolerance of the capacitor (±20% corresponds to ±8 F with a 40 F capacitor **yuden**). The order of magnitude tough should be correct.

5.2.3 Amount of display refreshes

With the results from the Equations (5.1) and (5.3), the amount of display refreshes can be calculated:

$$n = \left\lfloor \frac{162.4\,\text{J}}{2.7504\,\text{J}} \right\rfloor = 59. \tag{5.5}$$

Not that in this calculation, additional charging is neglected and the super capacitor is fully charged.

To verify this value, the super cap was charged to 3.5 V and the whole refresh sequence was repeated until the voltage reached 2.2 V. Under this conditions, the display could be refreshed 46 times.

Chapter 6

Summary

6.1 Conclusion

In the course of this thesis, a working prototype of a occupancy schedule was developed. As components served several development kits and simple prints. Using a low power wakeup receiver turned out to be the right approach, since the power consumption during standby mode could that way be reduced to a minimum of $4.5\,\mu$W. This power loss is easily compensated by the solar cells which deliver $485.52\,\mu$W given an illuminance of 200 lux.

With a self-holding circuit, it was possible to turn off parts of the receiver completely and switch them back on with the a wakeup interrupt.

6.2 Future work

6.2.1 Power management

It is necessary to protected the capacitor from overdischarging.

Sources

List of Figures

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 18, 2019

Signatures

Cedric Renda

Manuel Tischhauser

Appendix A

Requirements

A.1 Assignment

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

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 Arbeits-platzordner oder als *.docx File auf dem Public Server (\\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 durchgeführten Projektplans. Weitere Einzelheiten werden an den wöchentlichen Besprechungen festgelegt. Die Arbeiten sollen möglichst selbständig 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-Plagiatserklärung 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.

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

Rapperswil, [Datum]

Viel Erfolg wünscht Ihnen

Heinz Mathis,
Dozent Mobilkommunikation

A.2 Requirement Specification

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

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

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.

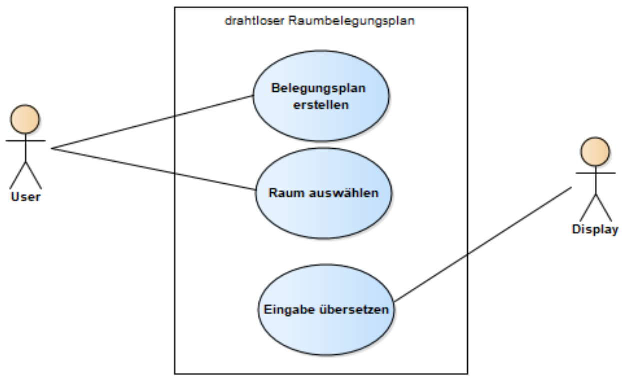


Abbildung 2.1: *Use-Case Diagramm*

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 voneinander 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

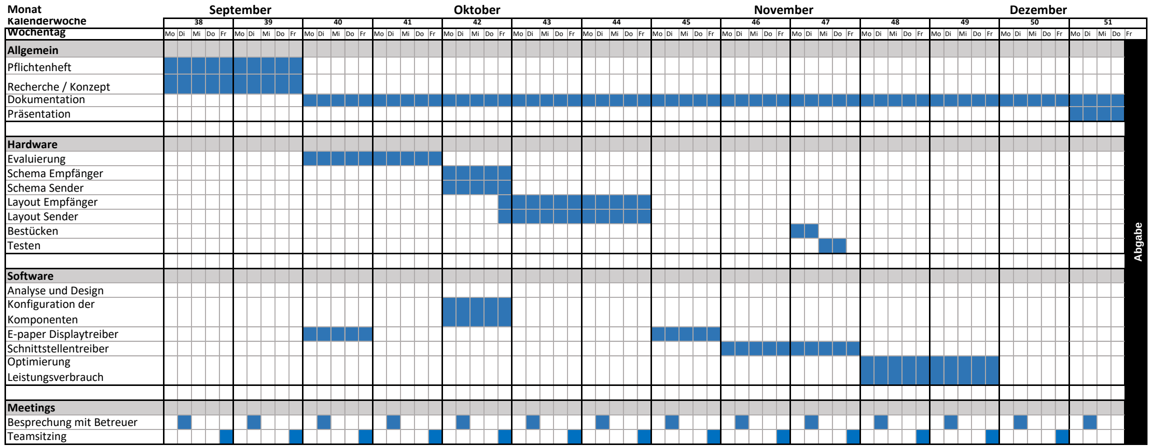
3.3 Varianten/Optionen

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


Monat
Kalenderwoche
Wochentag
September
Oktober
November
Dezember
38
39
40
41
42
43
44
45
46
47
48
49
50
51
Mo Di Mi Do Fr
Allgemein
Pflichtenheft
Recherche / Konzept
Dokumentation
Präsentation
Hardware
Evaluierung
Schema Empfänger
Schema Sender
Layout Empfänger
Layout Sender
Bestücken
Testen
Software
Analyse und Design
Konfiguration der Komponenten
E-paper Displaytreiber
Schnittstellentreiber
Optimierung Leistungsverbrauch
Meetings
Besprechung mit Betreuer
Teamsitzing
Abgabe