# Sensor patch

## BAP Group T
## Subgroup:
## Microcontroller and Communication

R. van Dijk
G. Gianfranceschi

Technische Universiteit Delft

TU Delft

# Sensor patch

## BAP Group T

## Subgroup:

## Microcontroller and Communication

by


R. van Dijk
G. Gianfranceschi




| | | |
|---|---|---|
| Student numbers: | Robbin van Dijk | 4568761 |
| | Giuseppe Gianfranceschi | 4572718 |
| Project duration: | April 20, 2020 – July 3, 2020 | |
| Thesis committee: | Dr. ir. G. de Graaf, | TU Delft, supervisor |




An electronic version of this thesis is available at http://repository.tudelft.nl/.

# Abstract

This research investigates the feasibility of a sensor patch for newborns. A sensor patch would be a device that is applied to the skin of the newborn. It can monitor the patient's temperature, heart rate, respiration rate, oxygen saturation and asses its general appearance of the skin so that it can check the newborn for jaundice. It should then communicate wirelessly to a base station which should have a compatible communication protocol. What happens after receiving of sensor data at the base station is beyond the scope of this project. One of the major challenges in the design process is energy efficiency, since the device should be small enough to fit on a baby and the battery should thus also be small. Additionally, the sensor patch should be able to operate for 3 days continuously.

The overall design comprises sensors, a battery with battery control system, a microcontroller and a communication module. The overall design is split up over three reports. This report focuses on implementing the microcontroller and communication protocol.

It is found that using a design with separate microcontroller and communication device is more efficient than using a so-called System on Chip where all of this is mounted on a single device. Then a comparison between different microcontrollers, communication protocols and communications devices is made. It is found that a design with an MSP430FR5994, Bluetooth Low Energy protocol and RSL10 communication chip fits the requirements best.

The research concludes that with the chosen hardware it is possible to meet the set requirements for the subsystem. Further research still needs to be done towards optimising software for the chosen devices.

# Preface

This report was written in spring/summer of 2020. At this time, Covid-19 has caused limitations on the research. Due to the disease many facilities at the university were not available. During the report it will sometimes be mentioned that due to circumstances certain actions were not possible. This refers to the covid-19 situation.

This project is part of the bachelor graduation project of 6 students at Delft University of Technology. The group of students is split up in smaller groups of 2 students which each solve a part of the design. The research was proposed by Dr. ir. G. de Graaf who works at the microelectronics department of the Delft University of Technology. After the proposal was accepted by the students, they came into contact with Dr. J. Dudink who is a specialist in neonatology and neoroscience. With his help, a better understanding about what the design should look like was found.


## Acknowledgements

During this project there have been several people who contributed to our design voluntarily. We are very grateful for that and would like to thank them.
A special thanks to our supervisor Dr.ir. G. de Graaf for all his time and advice in the guiding of the project and writing of the report.
Also, a special thanks to Dr. P. Pawelczak for his advice on microcontrollers and embedded systems in general.
Finally, a special thanks to Dr. J. Dudink for his advice on healthcare.


*Robbin van Dijk & Giuseppe Gianfranceschi*
*June 2020*

# Contents


1 Introduction 1
1.1 Problem definition . . . 1
1.2 Scoping and bounding . . . 1
1.3 State of the art analysis . . . 1
1.4 Thesis synopsis . . . 2

2 Program of requirements 3
2.1 Background . . . 3
2.2 Requirements of entire system . . . 3
2.3 Decomposition of the system . . . 5
2.4 Requirements for communication and microcontroller group . . . 6

3 Functional decomposition and physical assignment 8
3.1 Functional decomposition . . . 8

4 Physical implementation microcontroller 12
4.1 Microcontroller requirements . . . 12
4.2 Comparison . . . 13
4.3 Discussion . . . 16

5 Implementation wireless communication 17
5.1 Safety . . . 17
5.2 Choice of communication protocol . . . 17

6 Physical implementation BLE module 20
6.1 BLE roles . . . 20
6.2 Making a connection . . . 21
6.3 Operations per packet . . . 21
6.4 Packet size . . . 21
6.5 Simulating BLE energy usage . . . 22
6.6 Results . . . 26

7 Software 27
7.1 Hardware used for software testing . . . 27
7.2 Choice of wired communication protocols . . . 27
7.3 MSP430FR5994 programming . . . 28
7.4 nRF52810 programming . . . 29
7.5 Results . . . 30
7.6 Future work / improvements . . . 31

8 Results 32
8.1 Microcontroller and communication design . . . 32
8.2 Overall design . . . 32
8.3 Physical characteristics prototype . . . 33
8.4 Results of final design . . . 33
8.5 Conceptual use case . . . 34
8.6 Sterilisation methods . . . 34
8.7 Conclusion . . . 35
8.8 Future work . . . 35

9 Recommendations 36
9.1 Future work for overall design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
A Communication 37
B Simulation results 40
B.1 Bluetooth Low Energy simulations . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
C Program code 46
C.1 MATLAB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
C.2 MSP430FR5994 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
C.3 nRF52810 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
D Figures 48
Bibliography 50

# 1
# Introduction

## 1.1. Problem definition

Maternity care has always been a big concern in hospitals. It has developed a lot over the course of the years. It was long thought that, for instance, babies were unable to feel pain [1]. Recently, there has also been more attention for the psychological effects on newborns on neonatal intensive care [2]. It was discovered, there is a relationship between the bonding of a newborn with its mother in the first couple of days and long-term emotional well-being of the newborn.

Due to the rising awareness of this fact, hospitals are trying to improve the amount of contact a parent can have with their child [3]. Using a sensor patch with Near-Field Communication (NFC) it is possible to remove all wires from the body of the the newborn. This way it is easier for the parents to pick up the newborn, thereby making it easier to bond with the newborn. This is a clever solution using NFC, but it is mainly meant for hospitals.

In order to improve the bonding between parents and newborns even further, the baby should be able to go home as soon as possible. However, newborns that are in the risk category should still be monitored. Therefore, a wireless sensor patch is needed that can fulfil all the tasks that would have been done by nurses at the maternal care in the hospital. This allows for better bonding with the parents, is cheaper than keeping newborns in the hospital and could trace diseases early in newborns to prevent disease.

## 1.2. Scoping and bounding

The objective in this report is to provide a theoretical design of the sensor patch. The sensor patch will be designed in such a way that it can measure, process, and deliver the data to a base station. The base station, in short, will be a device that receives the data wireless from the sensor patch. What happens after the data reaches the base station is beyond the scope of this research. For real implementation of this design, an IT infrastructure would be needed to handle the data.

Due to current circumstances a couple of experiments could not be performed. These experiments would be needed in order to:

- Calibrate the sensors of the sensor patch
- Provide the actual total power consumption of the designed system
- Accurately analyse and compare the power consumption of different microcontrollers and communication modules

## 1.3. State of the art analysis

Wireless sensors are gaining territory in the a wide range of applications. This is due to their increasingly smaller sizes, lower power consumption and increase in energy harvesting efficiencies. A wide range of wireless sensor nodes are developed such as office sensors[4], machine health monitoring

[5] and medical sensors [6]. In addition, a lot of research has been done in biomedical wireless sensors field already. Wireless sensors measuring heart rate, temperature, oxygen saturation and many more are already common in patches or implants [7]. Also systems with real-time monitoring are being developed for, for instance, cough detection, breathing rate [8] and electrocardiogram (ECG) measurements [3].

The main concern in such products is battery life. The battery life is determined by the battery capacity and energy consumption. In energy consumption, the wireless communication often has the largest share. However, battery life increased a lot recently. This is due to both increases in miniature battery capacity [9] and low-power reliable communication protocols [10].

Most designs are, however, not yet adapted to health care situations that give rise to very specific requirements. Most designs are made to measure vital functions and are made for adults, while much progress can also be made in more specific applications. An example of this is about research in wireless sensor node in neonatal intensive care in hospitals [3]. However, hardly any research has been done toward a design that can help monitor newborns at home.

## 1.4. Thesis synopsis

This thesis contains nine chapters. First, the program of requirements will be presented: first the requirements for the overall system, then the requirements of the microcontroller and communication system. Then, in the functional decomposition, the behaviour of the ideal system will be described and the system will be split up in subsystems. In Chapter 4, 5 and 6 a comparison is made between different options for the microcontroller, wireless communication protocol and wireless communication device respectively. Each chapter concludes with a solution that best meets the set requirements. In Chapter 7, the software design for the given devices is presented. Additionally, the choice for software programmable characteristics is elaborated on. In Chapter 8, the final design and results will be discussed. Finally, Chapter 9 will discuss recommendations for future research.

# 2 Program of requirements

## 2.1. Background

Every year, a lot of babies find themselves in maternity care shortly after birth. This is a very costly environment for the baby to be in. Next to this, the environment causes a lot of stress for the babies. In addition, there are some babies every year that are not monitored enough in home care. The consequence of wrong monitoring at home results in permanent handicaps due to jaundice for 10 babies each year in just the Netherlands. Therefore, the goal of this project is to replace the maternity care with a sensor patch.

In order to outperform maternity care in hospitals, the sensor patch should at least measure all the parameters the nurses measure. Next to that, it should also do this at least as frequently as the nurses do. In the Netherlands, a baby in maternity care often stays there for around 2-3 days. Nurses in maternity care check on babies every 4 hours and check the following parameters:

1. Temperature
2. Heart rate
3. Respiration rate
4. Oxygen saturation
5. General appearance of the skin, including an assessment of jaundice.

This last point, general view at skin, means the nurse checks whether the baby looks healthy overall. This is done to get an impression of whether the baby might have underlying conditions that are hard to monitor with current technology. Jaundice is one of the most important examples of these conditions.

In addition, measurements performed in hospital maternity care involve multiple wired electrodes to be attached to the babies. This decreases comfort for babies and can increase their stress levels. Secondly, for the parents of the newborns this is not a pleasant sight.

## 2.2. Requirements of entire system

Based on the previously sketched background and consultations with J. Dudink [11] the requirements of the entire system can be described in greater detail. J. Dudink is a specialist in neonatology and neoroscience. From 2004 to 2016 he has worked as a neonatologist and researcher of neonatal neuroimaging at Erasmus Medical Center. J. Dudink is currently working at the Department of Neonatology at University Medical Center Utrecht.

During the consultations with Dr. Dudink, it was decided that the sensor patch should be located on the baby's chest. At this location, the sensor patch is not prominently visible, which is pleasant for the parents.. Moreover, there is more room for the sensor patch here than in other places. In addition, the doctor stated that the size of the sensor patch may not exceed 70mm x 30mm x 10mm (L x W x H). In order to outperform the maternity care, as it is now the sensor patch will need to perform measurements at least once every two hours to improve on the once every four hours in hospitals, in order to provide data significantly more frequent. In addition, at least the same measurements as maternity care must be carried out to guarantee at least the same level of care. This means that at least temperature, heart rate, oxygen saturation and bilirubin need to be measured. Furthermore the sensor patch needs to last for 3 days without the need to perform a recharge of the battery as this is how long babies are monitored in hospitals. In modern life, environmental awareness plays an important role. The sensor patch can eventually be used on many newborn babies, creating a lot of electronic waste if it is not reusable. It would be irresponsible not to take this into account during the design process. Moreover, one of the goals of the world health organisation is to create less waste and environmental impact [12]. Therefore it was decided that the sensor patch has to be reusable and preferably usable for at least 1 year with 2 uses every week. This does mean that it must be possible to clean the sensor patch to medical standards and preferably to make it sterile.

The previous defined requirements are divided between mandatory requirements and trade-off requirements. Mandatory being the requirements which are absolutely essential for the system to be marked a success. Trade-off requirements are the requirements which are pursued as much as possible but, should they not be met, the system will still be usable to a satisfactory extent.

### 2.2.1. Mandatory Requirements

1. Dimensions shall not exceed 70mm x 30mm x 10mm (L x W x H) .
2. Must measure temperature, heart rate, oxygen saturation, bilirubin
3. The previously mentioned measurements need to be performed at least once every 2 hours.
4. Must have battery life of at least 3 days.
5. The sensor patch must be wireless.
6. It should be able to detect neonatal jaundice when the baby is at risk.
7. Be able to communicate these measurements to a hospital real-time.
8. Must be reusable.
9. Must be sterile as it will be medically used.

### 2.2.2. Trade-off Requirements

As the sensor patch will be worn continuously by babies for possibly three days, it is desirable to make it comfortable for the baby. This can however impact the mandatory requirements negatively like size and battery life as a smaller sensor patch would probably be more comfortable, but would limit the space for the battery and components. Making the sensor patch waterproof would make it possible for the baby to take a bath without damaging the sensor patch. It would make the design more complex and possibly negatively influence sensor accuracy. On the other hand making the device waterproof will increase the lifetime of the sensor patch. The duration of reliable operation of the sensor is influenced by the price and availability of components and the resistance to damage. The sensor must be reliable for at least a year as this will make the sensor reusable to a high degree, however making it last longer can end up in a more expensive product. Therefor these are trade-off requirements.

1. Should be waterproof
2. The patch must provide reliable operation for one year, used on two patients per week.
3. Should not bring discomfort to the baby

## 2.3. Decomposition of the system

The project has been divided into three subsystems. The systems are: energy, sensors and communication and microcontroller. The system has been divided into these subsystems since they are expected to have equal weights. Besides this, each subsystem has a clear unique challenge that needs to be solved. The energy group needs to simulate the energy consumption and pick an appropriate battery. The sensor group needs to design sensors that meet a set of specific requirements. Finally, the microcontroller and communication group needs to design an efficient smart system that controls the entire system.
By dividing the system in these 3 clear divided subsystems, it is possible to develop each subsystem in parallel since there are little dependencies between systems. This allows for more streamlined and faster development.

#### Energy system

The energy group focuses on everything related to energy and power within the system. Energy storage, power and energy management and visualising data with the aid of a MATLAB model are the topics and tasks of this subgroup.

#### Sensor system

The sensor group will design an sensor system to measure temperature, heart rate and respiration rate, oxygen saturation and bilirubin. The bilirubin sensor has to be designed since this kind of sensor is not yet available to buy off the shelf. The other measurements can be performed using already available sensors. These sensors will need to be selected based on several parameters (e.g. accuracy, energy consumption)

#### Microcontroller and communication system

This group will look at the implementation of a microcontroller which will act as a central processing unit for driving all actions within the system. Additionally, this group will look at different possibilities for choices of communication protocols by which the system will communicate wirelessly with the user. When a communication protocol has been chosen, this group will look at the most efficient implementation and integration with the microcontroller.
In the following section, the specific requirements for the subsystem of this thesis will be elaborated on further.

## 2.4. Requirements for communication and microcontroller group

In order to drive the entire system, a central processing unit is required which will handle when operations in the system take place. Additionally, in order for the system to be able to communicate its measured data with the base station, the system needs a way to transmit this data to the user. For these purposes a microcontroller and a communication module are required. This distinction is made in order to draw up requirements for the two sub-parts separately. However, it could be possible that in the final design both the microcontroller and the communication will be handled by a single component. The microcontroller will act as a central control unit. It will drive the actions of the entire system, including the communication module. It will tell the sensors when to take a measurement and will be able to receive their readouts.
Here, the requirements have been divided into mandatory requirements and trade-off requirements, just like in the previous section.

### 2.4.1. Mandatory requirements

#### Control

The micro-controller shall be able to drive all operations on the system. When they will take place and how often. It will drive the sensors as well as the communication module.

#### Power

In order to facilitate the system wide requirement of a battery life of at least 3 days. A battery life of less than 3 days would be unusable in monitoring newborns since it requires replacing the patch too often. The micro-controller and communication module should be bound in their power usage such that this is realisable.

#### Readout frequency

The system should produce a readout to the end user once every hour.

#### Readout sensor

The microcontroller shall be able to interpret the sensor readouts. The sensor readouts can be both digital or analogue. Depending on that, the microcontroller should be able to either readout the analogue sensor signal or be configurable with the sensor's protocol.

#### Multiple sensor readouts

The microcontroller shall be able to handle the data from the different sensors. This means the microcontroller has an input for all sensors and can identify which input value corresponds to which sensor.

#### Throughput

In the early stages correspondences with the subgroup responsible for the design of the sensors of the system took place about the amount of data which will be received from the sensors. A maximum was given by the sensor design team of 120.000 bits. The amount of data from the sensors per measurement will never exceed this amount. Therefore the microcontroller and communication module shall be able to process up to 120.000 bits according to the rest of the requirements.

### 2.4.2. Trade-off requirements

#### Dimensions

In order for the system to be comfortable for the wearer, in our case a new born, it should be small in size. Therefore it is desire able to minimise the dimensions of the micro-controller and communication module. This will allow the system to have a small form factor and be more comfortable to be worn.

#### User feedback

The user should be appraised of the status of the system. They should receive notifications for when the battery is about to be empty or when the contacts for the sensors are not applied properly.

Configurability
The micro-controller shall facilitate that the user will be able to modify settings such that the operation of the system can be altered. Measurement frequency and which types of measurements are executed will be configurable.

Two way communication
In order for the system to receive commands from the user as well as relay its measurement data, a two way communication stream is needed. Therefore the communication module will be able to transmit and receive information. Additionally the micro-controller will be able to communicate with the communication module in two ways as well.
Now the requirements for the subsystem 'Microcontroller and communication' have been clearly defined. This gives clear guidelines for the design process. Additionally it will allow the final result to be tested against the requirements to see if they are met. If this is the case the final design will be deemed a success. If not, it will require additional work and research.

In the following chapter the functional decomposition of the subsystem will be elaborated on. Giving a clear overview of all in and outputs of the system.

# 3 Functional decomposition and physical assignment

The program of requirements set the requirements for the micro controller and communication module. Now it is important to find a solution that fits these requirements. Therefore it is important to first look at the functional decomposition: which requirements should the microcontroller satisfy and which requirements should the communication module satisfy. Since it is not within the scope of this project to design a microcontroller or communication module from scratch, off-the-shelf modules will be considered only. A look will be taken at which off-the-shelf components best fit these requirements, making sure they meet the mandatory requirements and satisfy the trade-off requirements to a degree as large as possible. Once these components have been chosen, the software will be presented that needs to run on the devices. It is equally important that the software is in line with the program of requirements since it will have a large influence on how well the design will meet these requirements.

## 3.1. Functional decomposition

The first decision to be made in the functional decomposition is whether the micro controller and the communication module should be separate off-the-shelf components or a single off-the-shelf component. The most important requirements here are energy and dimensions: all other requirements are not influenced by whether the micro controller and communication module will be one off-the-shelf component or multiple.

There are many System on Chips (SoC) nowadays that implement both a communication module with a microcontroller. The functional decomposition will now assume that they are separate, but later in the research a comparison will be made between those two configurations. One reason why the functional decomposition assumes separate components is because some papers suggest such a design is in general more power efficient [13]. Also, a separate microcontroller and communication module are a good place to start: when the ideal combination between those components have been found, it can always later be integrated onto a single SoC.

Figure 3.1 shows two parts in the design: the sensor patch and the base station. Both will now be discussed.

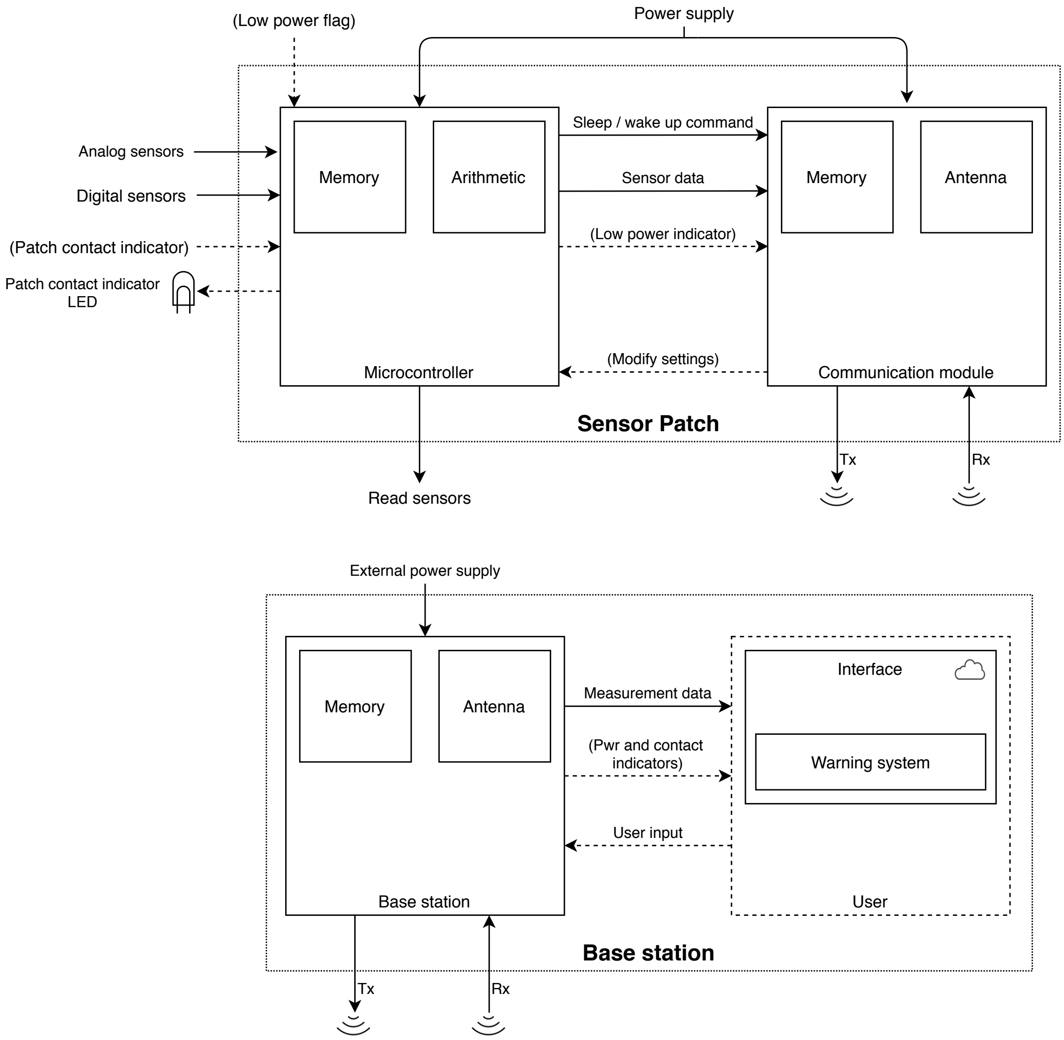



Figure 3.1: Functional decomposition of system with in- and outputs

### 3.1.1. Sensor patch

The sensor patch is divided in the microcontroller and communication module as described before. Most inputs are digital, but 2 sensors have analogue data. In this section the functions of the inputs, outputs, microcontroller and communication module will be described. The dashed lines are inputs and outputs that are not needed for mandatory requirements and thus optional. The normal lines are inputs and outputs that are necessary to meet the mandatory requirements.

#### Analogue sensors

This is the analogue data the microcontroller get from the sensors. These analogue signals should be sampled and stored on the microcontroller before passing them to the communication module. This input is required to satisfy the "Data accessibility" requirement.

#### Digital sensors

This is the digital data the microcontroller gets from the sensors. This can be communicated in several ways like I2C, SPI or UART for example. This input is required to satisfy the "Data accessibility" requirement.

#### Low power flag

This is an input coming from the power management module. It should inform the micro controller so that it can take take proper action, like notify the user. This input is required to satisfy the "User feedback" requirement.

### Power supply

This is the power supply coming from the battery. It provides the energy for the micro controller and the communication module.

### Read sensors

This is an output that notifies the sensors that the micro controller would like to receive their data. An implementation where each sensor can be requested separately is preferable since this gives more freedom in design. This output is required to satisfy the "Multiple sensor readouts", "Data accessibility" and "Control" requirements.

### Tx and Rx

These input and output communicate with the base station. These input and output are required to satisfy the "Data accessibility", "Processing ability", "User feedback", "Configurability" and "Two way communication" requirements.

### Sleep / wake up command

This is an output of the micro controller and input of the communication module. The communication module is one of the most power consuming part of the sensor patch when active. Therefore this signal should carefully control when the communication module is on. This internal output/input is required to satisfy the "Power" requirement.

### Sensor data

This is an output of the micro controller and input of the communication module. This is data that has been stored by the micro controller and is now ready to be communicated to the end user via the communication protocol. This internal output/input is required to satisfy the "Data accessibility" requirement.

### Low power indicator

This is an output of the micro controller and input of the communication module. This is a flag that indicated that the battery is almost empty. It is a flag the should be communicated to the end user via the communication module. This internal output/input is required to satisfy the "User feedback" requirement.

### Patch contact indicator

This is both an input and an output to the microcontroller. The input gives a signal to the microcontroller to indicate if the patch is applied correctly. It this is not the case, the microcontroller will turn on a LED to indicate the user that the patch is not applied properly. This in- and output is required to satisfy the "User feedback" requirement.

### Modify settings

This is an output of the communication module and input of the micro controller. It is data that is communicated by the end user via the base station. It is supposed to alter settings like measurement intervals. This internal output/input is required to satisfy the "configurability" requirement.

### Microcontroller

The microcontroller should decide when the sensors should be read using the read sensor output. Also, it should receive the sensor data and handle the data in such a way that it can be transmitted using the communication module. It should also be able to generate the following outputs at the right time: sleep mode flag, sensor data flag, low power indicator flag and patch contact indicator flag. All this should be done with minimal power consumption.

#### Communication module

The communication module should be able to receive all sensor data coming from the microcontroller. It should transmit the data to the base station. All this should be done with minimal power consumption.

### 3.1.2. Base station

The base station is referred to as the station that receives the data from the sensor patch. Multiple base stations may be possible: it just needs to receive the data from the patch and communicate it to the internet so that the end user can retrieve the data from the cloud.

#### External power supply

This is the power supply coming from an external battery or the net. It provides the energy for the base station.

#### Tx and Rx

These input and output communicate with the sensor patch. These input and output are required to satisfy the "Data accessibility", "Processing ability", "User feedback", "Configurability" and "Two way communication" requirements.

#### Measurement data

This is an output of the base station and input of the cloud. It communicates the measured data from the sensor patch to the cloud where the end user can access them. This output is required to satisfy the "Processing ability" and "Data accessibility" requirements.

#### System indicators

This is an output of the base station that communicates the notifications of the sensor patch, like low battery flags, to the cloud where the end user can see them. This output is required to satisfy the "User feedback" requirement.

#### User input

This is an output of the cloud and input of the base station. This output/input allows changes in settings, like changing the measurement intervals. This output/input is required to satisfy the "Configurability" requirement.

#### Base station

The base station is the device that should receive the data from the sensor patch. This device can be any device that has both a communication protocol that is compatible with the sensor patch and an internet connection. The internet connection will be necessary for transmitting the data to a cloud where the doctors can reach it. This base station and infrastructure are beyond the scope of this project as explained in Section 1.2.

Now that is is clearly defined what the system should behave like and what its in- and outputs will be, the following chapters will determine which specific off-the-shelf components best suit the needs of this project. The comparisons will rely on the requirements and functional decomposition as given up to this point.

# 4
# Physical implementation microcontroller

## 4.1. Microcontroller requirements

In order to pick the best suited micro controller it is important to know what the important parameters are for the micro controller. The microcontrollers will be compared based on these criteria:

1. Energy use of microcontroller
2. Microcontroller is able to store the program
3. Microcontroller is able to store the data received from the sensors (120kbits)
4. Size of Microcontroller
5. Microcontroller has the peripherals: I2C, UART, SPI, GPIO, ADC
6. Peak current
7. Clock configurability
8. Voltage configurability

Since energy is a complex topic in microcontrollers, and it is an important characteristic in the design, it will be further elaborated on first. Then the other, "general" criteria will be elaborated on shortly.

### 4.1.1. Energy

The power consumed by the micro controller is determined as:

$$P = V \cdot I \tag{4.1}$$

Where V is the supply voltage and I is the current. The current is not always the same however. Depending on the state that the microcontroller is in, the current can be different. Therefore, the average current is an important property. This average current is determined by[14]:

$$I_{Av} = I_{Standby} \cdot R_{Standby} + I_{Active} \cdot R_{Active} + I_{Av,Peripheral} + I_{Data} \cdot R_{Data} \tag{4.2}$$

In which R is the ratio of time for which the concerned current is present compared to the period. The standby current is the current in the standby state, the active current is the current in active state. The peripheral current is the current drawn by peripheral devices and can be calculated as:

$$I_{av,peripheral} = I_{I2C} \cdot R_{I2C} + I_{SPI} \cdot R_{SPI} + I_{UART} \cdot R_{UART} + I_{ADC} \cdot R_{ADC} \tag{4.3}$$

Finally, the data current is the current drawn by the memory. Since these currents and ratios depend on many factors on many devices, a more in closer look will be taken at this in the comparison between devices

### 4.1.2. General

#### Storage

The microcontroller should be able to store the finite state machine program, which follows from the control requirement in Chapter 2. At this point it is not clear how big the program will be, but after storing the sensor data, there should be significant space left for the program. In the prototype, the device with a memory as large as possible should be chosen so that in later models of the sensor patch this can be reduced. As stated in 2, the sensor data will be no more than 120kbits.

Dimensions
The size of the microcontroller should be as small as possible.

Peripherals
The microcontroller should have the following peripherals: I2C, UART, SPI, GPIO and ADC. I2C will be used for the communication between the sensors and the microcontroller. SPI or UART will be used for communication with the BLE device. ADC will be used for the analogue sensor data (jaundice measurements). Finally, the GPIO will be used for any of the other inputs/outputs.

Peak current
Peak currents are undesirable for the battery. Therefore the peak current should stay as low as possible.

Clock configurability
The clock frequency should be high enough to handle the data throughput. Since the data throughput is very small in this device, a clock speed of higher than 1 MHz will not be necessary. Higher clock speeds will result in higher energy consumption and are therefore undesirable. A property that would be very useful is high flexibility for clocks. The ADC for instance needs a very low clock frequency of around 50Hz, while the CPU needs a frequency that is higher than that.

Voltage configurability
The voltage of the system should have some flexibility so it can be adjusted to a voltage level that is also useful for the rest of the device.

## 4.2. Comparison

A comparison of 3 microcontrollers/System on Chips will now follow. They will be tested on previously mentioned criteria. The energy consumption is based on Equation 4.2. It has been calculated in MATLAB using the respective data sheets, since actual measurements were not possible due to current circumstances. The code for this can be found on the GitHub described in Appendix C.1. These three microcontrollers have been chosen based on the fact that they have free easy to use integrated development environments (IDE) and their manufacturers have a reputation of being high end and low power.
Some assumptions will be made to make the comparison as fair as possible. All microcontrollers will be compared under the condition that:

1. The CPU will run at 1 MHz
2. The period will be 1 hour
3. The active mode will last 15 seconds (measuring, transmitting and receiving will happen here)
4. The supply voltage will be 3V
5. The ADC will measure during 10 seconds

### 4.2.1. MSP430FR5994

Energy
The MSP430FR5994 [15] [16] has a low power mode where everything is turned off, except for a Real Time Clock(RTC). This RTC allows the microcontroller to wake up after a certain time interval. The power consumed in this standby state is 350nA. In active mode, the microcontroller uses 118$\mu$A/MHz. The ADC converter uses 85$\mu$A at 1.25MHz. However, the ADC converter should operate at 50Hz. Since there is no data available about this frequency, it is uncertain how relevant this data is. A UART connection consumes 6.3$\mu$A/MHz, an SPI or I2C each consumes 4$\mu$A/MHz. With a 1MHz master clock for these devices, their on-time can be calculated taking the overhead of each protocol into account. With this the ratio can be calculated. When transmitting 50kbits over each of these protocols, using Formula 4.3, the average peripheral current becomes 236nA. The MATLAB code for this can be found on the GitHub in Appendix C.1.
The MSP430FR5994 has no additional data logging current. This current is already included in the

active mode current of 120$\mu$A/MHz. This is due to the FRAM technology, which does not require additional erase and has low power writes. This is a big advantage in this microcontroller. This results in a total average current of:

$$I_{Av} = 350 \cdot 10^{-9} \cdot \frac{3600-15}{3600} + 118 \cdot 10^{-6} \cdot 1 \cdot 10^{6} \cdot \frac{15}{3600} + 236 \cdot 10^{-9} = 790nA \tag{4.4}$$

#### General

The MSP430 series is the Texas Instrument's low power microcontroller. It has a 16 bit RISC core. It has 256KB of FRAM memory and 8KB of SRAM and it is a 16 bit microcontroller. The size is down to 6x6mm using the NFBGA (ZVW) package. It uses 4 Enhanced Universal Serial Communication Interfaces (eUSCI) for UART and SPI and another 4 eUSCIs for I2C and SPI. It also has 20 ADCs.
Peak currents are often generated by active modes and/or memory erases and writes. MSP430 uses FRAM which takes away these currents (these currents are included in current analysis when FRAM is on). Therefore the only peak in current is caused by the active mode and the peripherals. This has an advantage on energy consumption but is also a huge advantage for the battery since high peak currents are undesirable. The total peak current is then calculated as the current of the active mode, I2C, SPI, UART and ADC. At 1 MHz this results in a peak current of 217.3$\mu$A.
The clock frequency goes up to 16MHz. The clock system offers 4 output clocks which are all software dividable, and can also pick several clock sources. Several subsystems also have the option to divide these clocks again at their input. The ADC converter can sample at a rate that comes very close to the desired 50Hz (approximately 52Hz). The voltage is also very flexible and can range from 1.8V to 3.6V.

### 4.2.2. STM32L422CB

#### Energy

The STM32L422CB has a low power mode where, similar to the MSP430, everything is turned off and wake up happens from a RTC. The power consumed in this state is 320nA. In active mode, the microcontroller uses about 84$\mu$A/MHz. The ADC converter does not allow for low-frequency sampling, so an external ADC converter which can do this is needed. The ADS1115 can be used for 50Hz measurements at 150$\mu$A and a voltage support of 2.0 to 5.5V. A UART connection consumes 600nA/MHz. SPI consumes around 1.6$\mu$A/MHz and I2C consumes around 0.95$\mu$A/MHz. SPI works at 1MHz and I2C at 100kHz. When transmitting 50kbits over each of these protocols, using Formula 4.3, the average peripheral current becomes 417nA.
The STM32L422CB has FLASH memory. This means that before the memory can be written, it first has to be erased per segment. This causes additional currents. All data from the sensors has to be erased and rewritten every cycle. The current drawn when data is erased or written is 3.4mA. Programming of a 2kB page takes 20.91 ms and erasing a page takes 22.13 ms. The number of bits coming from the sensors is approximately 120kbits. 2kbyte is 16kbit, which means that each cycle 8 pages have to be erased and programmed, which comes down to a total time of 344.3ms. This results in an average current of:

$$I_{Av} = 320 \cdot 10^{-9} \cdot \frac{3600-15}{3600} + 84 \cdot 10^{-6} \cdot 10^{6} \cdot \frac{15}{3600} + 417 \cdot 10^{-9} + 3.4 \cdot 10^{-3} \cdot \frac{0.3443}{3600} = 1.41\mu A \tag{4.5}$$

#### General

The STM32L422CB is a low power microcontroller of STMicroelectronics. Its core is based on a 32-bit ARM Cortex M4 It has 128kB of FLASH memory and 40kB of SRAM. The size is down to 5x5mm using the UFBGA64 package. It has 3 I2C ports, 3 UART ports, 1 Low-Power UART port (the one previously considered) and 2 SPI ports. It also has 52 GPIOs. However, one external ADC component is needed since the ADC in the microcontroller can't measure at a low enough sample rate.
The STM32L422CB has a rather high peak current mainly due to its flash memory. The total peak current is calculated as the current of the active mode, I2C, SPI, UART, ADC and data logging current, since these might occur simultaneously. Since the Flash programming current is 3.4mA and the system operates at 1MHz this results in a peak current of 3.637mA.

The microcontroller has a clock system that goes up to 80MHz. The clock system does not seem very flexible: just for implementing the ADC an additional component is needed since the clock system can't deliver a low frequency. In general, the clock system seemed to be aimed at higher frequencies, where there was more flexibility. The voltage is very flexible and ranges from 1.71V to 3.6.

### 4.2.3. nRF52810

#### Energy

The nRF52810 is a SoC with BLE. It was chosen for its low transmitting power. In this analysis a look is taken at whether it is also suitable as main microcontroller. It was chosen because of its low transmitting power for BLE. However, now it is judged at its ability to perform the role of a microcontroller in the system. The BLE module will not be considered in the energy analysis, since this energy comparison is based on merely the microcontroller. The main reason for considering this microcontroller is to compare the energy consumption of a separate microcontroller with a SoC built-in microcontroller.
The nRF52810 has a low power mode where everything is turned off, except for the RTC. In this mode, the SoC consumes $1\mu$A. In active mode, without the BLE transmitting, the SoC consumes 2mA. The data sheets don't specify the peripheral currents. The data sheet also does not specify currents drawn by erasing and programming the Flash memory. However, since it is the same memory type as in the STM32L422CB it is safe to assume that this will be in the same order of magnitude.
It is hard to make a complete analysis about this device since so many parameters are missing. Yet, from just the active current and passive current, it can be seen that this is not likely a good design. The active current is around 9 times higher than the MSP430FR5994 and the passive current is around 3 times higher. This alone results in an average current of:

$$I_{Av} = 1 \cdot 10^{-6} \cdot \frac{3600 - 15}{3600} + 2 \cdot 10^{-3} \cdot \frac{15}{3600} = 9.33\mu A \tag{4.6}$$

Which does not yet include data logging current and peripheral current.

#### General

The device uses 192KB Flash memory and has 24KB SRAM. It is the smallest in size with 2.48 x 2.46 mm. It has the least peripherals but still meets the mandatory requirements: it has 1 of each required peripheral. The SoC uses a cortex m4 processor at 64 MHz, where the clock system is not configurable. The peak current for this device is the highest. Since the parameters of its data logging current were missing, for this the current of the STM was taken since it also uses Flash memory. Then, the peripherals were left out. The peak current is then calculated as the sum of the active mode current and data logging current. Since the clock system is not configurable, the active mode current is very high. This also results in a high peak current of 5.4mA without peripherals.

### 4.2.4. Results

The devices will now be compared based on how well they meet the requirements. Some criteria will be filled in with their actual value, like size and average current. Other criteria that are less tangible, like clock configurability, will be graded with a +, o, or -. A + will be assigned if the device acts in a significantly better way than the criteria asks. An o will be assigned if the device simply qualifies the specific requirement. A - will be assigned if the device fails to meet the requirement.
Table 4.1 summarises the findings of the previous chapter. It can be seen that the MSP430FR5994 has the lowest power consumption, which is a crucial part of the design. This is mainly because the nRF52810 has a core that always runs on 64MHz and the STM32L422CB has high data logging current. Also, the MSP430FR5994 has the lowest peak current. This is mainly due to the fact that its data logging current is much lower than the other two due to FRAM. The MSP430FR5994 has less SRAM than the other two candidates. This should not be an issue however: if the program becomes too large it can always be run from its FRAM. This may cause some delays, however the system does not require many fast operations. The nRF52810 is the smallest device of all. All devices have all

| Microcontrollers | MSP430FR5994 | STM32L422CB | nRF52810 |
|---|---|---|---|
| Average current [nA] | 790 | 1410 | 9330* |
| Storage (main) | 256KB FRAM | 128 KB FLASH | 192 KB FLASH |
| Storage (SRAM) | 8KB | 40KB | 24KB |
| Size [mm x mm] | 6 x 6 | 5 x 5 | 2.48 x 2.46 |
| UART | 4 | 4 | 1 |
| SPI | 8 | 2 | 1 |
| I2C | 4 | 3 | 1 |
| GPIO | 68 | 38 | 32 |
| Peak current [mA] | 0.2173 | 3.637 | 5.4** |
| Clock configurability | + | o | - |
| Voltage configurability | + | + | + |

Table 4.1: * no peripherals and data current. ** no peripherals and for data current, STM FLASH current was used

required peripherals, even though the nRF52810 does not have any additional peripherals for future expansions. Voltage configurabilty is good on all devices and should be no problem.
A thing that also strikes is that the nRF52810 in general does not perform well as a microcontroller. This is mainly due to the fact that the core always runs at 64MHz which results in a high peak current. Also, it runs on Flash which also does not perform well. Therefore this device does not seem like a good option as microcontroller.
The STM32L422CB seems to perform better in general. It has a better clock configurability and the peak current is lower. However, the average current is still twice as high as on the MSP430FR5994. Therefore the STM32L433CB also does not seem like the good option.
This leaves the MSP430FR5994 as the best design choice. It implements the requirements best: it is the lowest current, has all peripherals, can store the data of the sensors, has a lot of memory left for the program, is not smallest in size but still workable, lowest peak current and good voltage and clock configurability. This microcontroller will be chosen in the final design of the sensor patch.

## 4.3. Discussion

This section will discuss the results of the previous section. The comparison done to pick the microcontroller is limited. This is due to the circumstances of the research. Some of the main flaws in the research will now be discussed

### 4.3.1. Based on datasheets

The comparison is mainly based on datasheets. This is not the best realistic way to compare the performance of microcontrollers. Microcontrollers can deviate from the datasheets and not all scenarios are treated in the datasheets resulting in lacking results. The ideal situation would be performing measurements on many different microcontrollers on development kits. This was not possible, so the complete analysis was based on datasheets.

### 4.3.2. Number of compared microcontrollers

The analysis only compares 3 microcontrollers. This is a low amount of microcontrollers for a comparison. This is mainly due to the small time interval of the project. Microcontrollers are complex devices and understanding how each one works, merely based on datasheets, is a time consuming activity.

# 5
# Implementation wireless communication

This chapter will take a look at which wireless communication protocol best fits the set requirements. First, the safety of wireless communication in general will be discussed. Then a comparison between different communication protocols will be made.

## 5.1. Safety

Radiation has different forms. Some are more harmful than others. Bluetooth is a form of radiation that operates at 2.4GHz. That places Bluetooth in the radio frequency band and therefore it is a so called non-ionising radiation. In contrast to for instance X-ray, non-ionising radiation cannot break chemical bonds or ionise atoms in the human body. The World Health Organisation has guidelines for the use of this radiation.

The WHO [17] has done (or refers to) a lot of research regarding the influence of mobile telecommunications on health. This telecommunication operates at the same frequency as Bluetooth and is therefore also representative for this project. A thing that might deviate is the power that is used to transmit or receive the signals. This might be different for Bluetooth modules. For mobile telecommunications, which is assumed in this source, typical transmit values range from 0.1W to 2W. The WHO says that there are some other factors that also play a role in the amount of received radiation. One of them is the distance between the transmitter and the skin. This is a big factor since received power has a reverse quadratic relationship to distance. Another factor is the amount of transmitter time. There are two types of health effects that need to be considered: short term and long term. For the short term, the WHO states, there is no scientific evidence that there are any negative effects. The long term is a bit harder to tell. Most of the research in this category has been aimed at a relation between exposure to radio frequency EM waves and brain tumours. Since these kinds of tumours are only discovered after a long period after they have been caused and mobile telecommunications have only become widely available in the 1990s, it is only possible to take into consideration the tumours that have evolved in the shorter-term. However, research with animals consistently show no extra risk in brain tumours to animals exposed to long-term radio frequency EM waves. There are some indications that there might be a slight increase in risk of brain tumours. This however is by no means sure yet and only in the highest 10% of mobile users. Further research to this is still ongoing.

The conclusion of this research is that radiation in the radio frequencies, definitely for younger children, might be harmful. However, this is by no means sure yet and is only the case in extreme exposure to this radiation. Nevertheless it is a good idea to turn off any communication device if possible.

## 5.2. Choice of communication protocol

There exist many different protocols for enabling wireless communication between devices. To make a well informed choice of protocol, only the most used protocols in the industry will be considered. This is because enough literature on these protocols is required to make a meaningful decision of which protocol is best suited for the purposes of this project. The most popular communication protocols are:

- Bluetooth

- Bluetooth Low Energy (BLE)
- ZigBee
- Wi-Fi
- GSM

NFC has been omitted from this list since it has a maximum operating distance of about 20 cm [18]. For the purposes of this project this maximum distance does not give enough freedom to the patient to move around and thus NFC will not be considered.

In theory, a smartphone could be used as a base station for the sensor patch. Because most smartphones feature multiple wireless connection technologies, an application could be written to enable them to act as a base station for the sensor patch. For this reason it is also desirable to only consider wireless communication protocols which can be received by the most common types of smartphones.

### 5.2.1. Comparison based on literature study

GSM is mostly used for long range applications and has a high power usage [19]. Since for this project very long range is not needed, the range does not weigh up against the power consumption and thus GSM will be discarded as a viable option.
When comparing Bluetooth, ZigBee and Wi-Fi, a study [20] found that the power consumption for Wi-Fi was significantly higher than compared to Bluetooth or ZigBee. The Wi-Fi protocol as studied in this paper uses around 7 times more power than Bluetooth and ZigBee. Therefore, in order to support the requirement of low power usage, Wi-Fi will also be discarded as a viable option. It must be noted that this paper is somewhat dated as it was written in 2007. Its conclusions may therefore not be entirely accurate in the present time.

The remaining comparison between Bluetooth, BLE and ZigBee needs to be made. From literature study it was found that compared to ZigBee, BLE has a lower power consumption. This is because BLE has a higher coding efficiency for transmitting payloads [21]. Thus less overhead is present when transmitting data and thus fewer bits need to be sent in total. Secondly it was found that in a cyclic sleep scenario, which is also the case for this project, BLE uses less energy because of its faster connection time [22]. Therefore the module needs to be out of its sleep mode less amount of time than a ZigBee variant. Consequently causing the overall power consumption to decrease [23].

### 5.2.2. Comparison based on off-the-shelf components

Because not many sources could be obtained from a more recent period and because of the rapid changes in the IoT industry, an additional survey was done on top of literature study. A list of the most energy efficient off the shelf communication modules was gathered for each communication protocol. By examining their data sheets, an estimate could be made into which type of module would be most power efficient.

As can be seen in Table 5.1, a number of modules have been compared with each other. This was done by searching for modules which were branded as the most energy efficient of their respective types i.e. their used communication protocol. The data sheets of these modules were examined and the values of 5 attributes were used for this comparison. Namely: dimensions (height and width), operating voltage, transmission current and sleep current. From these 5 attributes, three attributes can be derived: area, transmit power and sleep power. The results of these calculations can be found in Table 5.1, the values which were extracted from the datasheets which were used for these calculations can be found in table A.1. Since the maximum achievable transmission speed for the different protocols is known (see Table 5.2). An estimation can be made for the overall average power consumption for each module. Since the communication module will never need to transmit more than 120.000 bits, as stated in the mandatory requirements, a maximum transmission time and consequently a sleep time can be calculated. Using the ratio between time spent transmitting

Table 5.1: Comparison of communication modules

| | Area [mm$^2$] | TX Power [mW] | Sleep power [$\mu$W] | Average power [$\mu$W] |
|---|---|---|---|---|
| BLE | | | | |
| Tayo Yuden EYSHSN | 27,79 | 22,50 | 3,60 | 3,60 |
| SmartBond TINY DA14531 | 3,40 | 10,50 | 2,25 | 2,25 |
| MtM M905 | 42,25 | 23,43 | 3,96 | 3,96 |
| ISP091201 | 96,00 | 38,10 | 6,00 | 6,00 |
| Silabs BGM13S22F512GA | 42,25 | 29,37 | 4,62 | 4,62 |
| Silabs EFR32BG22 | 16,00 | 12,30 | 4,20 | 4,20 |
| u-blox ANNA-B112 | 42,25 | 15,90 | 6,60 | 6,60 |
| Wi2Wi WC7220B0 | 96,00 | 26,40 | 2,97 | 2,97 |
| Zigbee | | | | |
| Digi XBee 3 Zigbee 3.0 | 247,00 | 114,00 | 5,70 | 5,71 |
| E75-2G4M10S | 416,00 | 99,00 | 3,30 | 3,31 |
| Telit ZE51 | 390,00 | 115,50 | 3,30 | 3,31 |
| EFR32MG22 (SoC) | 16,00 | 12,30 | 4,20 | 4,20 |
| Wi-Fi | | | | |
| GS2200M | 240,98 | 485,10 | 26,40 | 26,40 |
| Silabs AMW006 | 308,56 | 37,62 | 12,54 | 12,54 |
| TI CC3200 | 81,00 | 547,80 | 13,20 | 13,20 |

and time spent in sleep mode, a rough estimation for the average power consumption of each module can be made. When comparing the results of this analysis, the following is concluded:

Table 5.2: Maximum transmission speeds for wireless protocols

| | Maximum data rate | |
|---|---|---|
| BLE [24] | 1,434 | Mbps |
| ZigBee [25] | 250 | Kbps |
| Wi-Fi [26] | 2,4 | Gbps |

- Wi-Fi modules have much higher power usage both in sleep mode and in transmission mode than compared to BLE or ZigBee modules. This is in support of the previous finding according to [20]. Thus this proves that the fairly dated conclusion of this paper still holds presently. Additionally the dimensions of the Wi-Fi are much larger than those of the BLE modules.
- ZigBee modules have a comparable low power usage to BLE when in sleep mode. Contrarily, their power usage is considerably higher when they are transmitting data. The EFR32MG22 is an exception in this case, having a comparable transmit power to BLE modules. This module is, however, a SoC (system on a chip). The purpose of this analysis is to determine the best type of communication module and not a whole system. Therefore this result will not be taken into account.

It it also apparent that, like the Wi-Fi modules, ZigBee modules are significantly larger in size.
- BLE modules performs best on all attributes in general. They have the lowest power consumption in transmitting and receiving mode. Additionally they are available in many tiny sizes designed for use in small applications.

### 5.2.3. Results

From the literature study which was conducted, it is clear that BLE is the most suitable wireless communication protocol for the purposes of this project. This conclusion is further supported by the brief examination of current off-the-shelf wireless communication modules. Combined with the results of Section 5.1 that Bluetooth is not harmful if not exposed excessively, leads to the conclusion that Bluetooth Low Energy will be the wireless protocol of choice going forward. The following chapter will elaborate more on how the protocol works and how its behaviour can be simulated in terms of power usage.

# 6

# Physical implementation BLE module

In order to make a meaningful design choice for the Bluetooth Low Energy module in terms of power usage, there must be an overview of all the operations a BLE module performs when it is operational. A comprehensive overview of the BLE protocol will now be given:

## 6.1. BLE roles

A Bluetooth Low Energy module can be in either one of four states when it is turned on. This state is defined by the Generic Access Protocol (GAP), and the possible states are as follows [27]:

- Broadcaster (TX only)
  - Sends advertising events and broadcast data.
- Observer (RX only)
  - Listens for advertising events and broadcast data.
- Peripheral (RX and TX)
  - Always slave, is connectable and advertising. Designed for a simple device using a single connection with a device in the Central role.
- Central (RX and TX)
  - Always master, never advertises. Designed for a device that is in charge of initiating and managing multiple connections.

For the BLE module of the sensor patch there are 3 role options, since the device must be able to transmit data. Namely: broadcaster, peripheral and central. Since the central role is reserved for a device which is in charge of the connection and will thus be able to change connection parameters and be able to send commands, it would be logical to assign this role to the base station. Then, for the sensor patch two options remain: broadcaster or peripheral.

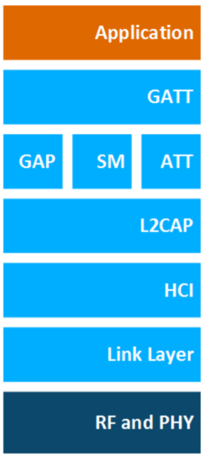



Figure 6.1: BLE architecture

When a BLE device is in a broadcasting role it does not need to establish a connection with a master in order to send data. It can freely send data packets in advertising channels and observers or centrals who are listening on these channels can receive these packets. An upside to this is that there does not need to be established a connection which will save some power in transmitting packets. Secondly sent packets do not need to be acknowledged by a receiver, also reducing power usage. The downside to this is that since there does not exist a L2CAP channel, the packets are not encrypted and therefore anyone can intercept and read them. Since we are designing a medical device which will measure sensitive data about someone's well being this is not acceptable.

Hence the BLE module will assume the peripheral role. This will, once a connection with a central has been established, create a L2CAP channel over which can be communicated securely without the risk of interception of data. Additionally this will allow the sensor patch to receive instructions from the user. Possibly allowing the system configuration to be changed remotely.

## 6.2. Making a connection

A peripheral will continuously send advertisement packets on the primary advertising channels. This will be done with a advertisement interval which is defined by the link layer. A central device will be listening on these advertisement channels for advertisement packets. When it receives one it will read the advertisement packet containing all relevant information about the peripheral device. It can then decide to make a connection with the device by sending a connection request containing important information like the hop sequence the devices will take. This will allow the two devices to synchronise and know when to listen on which channel. When the central receives a confirmation from the peripheral, a connection has been established and the devices are synchronised.

When two devices are connected, they have to periodically exchange a packet to ensure the connection is still alive. The start of each period in which the devices talk to each other is called a connection event. The period between connection events is called the connection interval. This interval therefore defines how long a connection event is.

## 6.3. Operations per packet

Once a connection is established, the attribute protocol (ATT) defines the communication between the devices. It defines two roles: Server and Client. The server stores data as attributes, the client collects information from one or more servers. In the case of this project, the sensor patch will act as a server, since it will be the one to store sensor data to be transmitted. The base station will act as a client. To define how the server communicates with the client there are two options according to the attribute protocol[27]. The server can communicate with the client unsolicited either by sending an notification or an indication. A notification does not need a confirmation back from the client that the packet has been received. Contrarily, an indication does require confirmation from the client back to the server.

According to the BLE protocol, there must be a mandatory $150\mu s$ delay between each packet sent, this is called the Inter Frame Space. To summarise, when a packet is sent from the server to the client via an indication (acknowledged), first the packet itself is sent. Then there is a delay of $150\mu s$. Then the server listens for an acknowledgement packet. Finally an additional delay of $150\mu s$ follows. After these four stages, the connection event is finished and a new connection event can take place. This is illustrated in Figure 6.2.

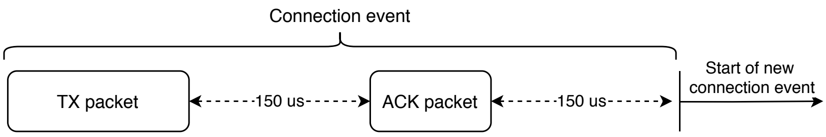



Figure 6.2: BLE connection event

## 6.4. Packet size

There are two possible maximum sizes for a packet which can be sent via the BLE protocol. This maximum size is defined by the Link Layer (Fig. 6.1). Link Layer packets or LL packets have a standard size of 41 bytes. As of BLE 4.2, LE Data Packet Length Extension (DLE) was added [24]. This extends the maximum LL packet size to 265 bytes. This decreases the amount of overhead when sending data. The make up of packets with or without DLE can be found in Figure D.1 and Figure 6.3 [28].

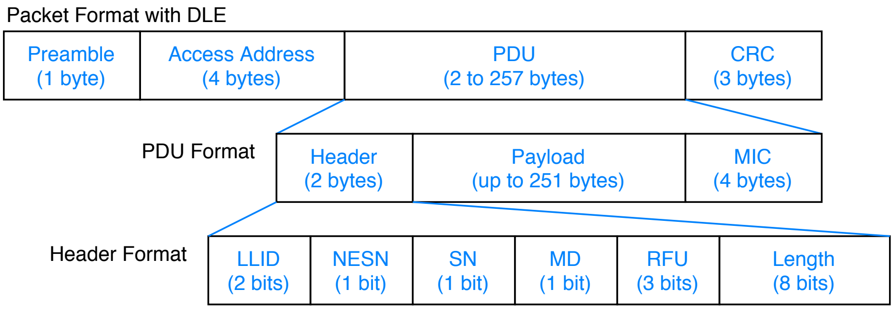



Figure 6.3: LL packet make-up with DLE

The main structure of the LL packet is described as follows [29]:

- The Preamble: A sequence of bits used by the receiver to set its automatic gain control and determine the frequency corresponding to the radio data rate itself. Its length can be 1 or 2 bytes, depending on the PHY rate (1Mbit/s or 2Mbits/s respectively)
- The Access Address: includes four bytes that identify the communication on a physical link, and it is used to exclude packets directed to different receivers.
- The Protocol Data Unit (PDU): This is the section of the packet which contains the payload data which needs to be transmitted. It consists of 2 header bytes, the payload and 4 MIC bytes which are used in LL encrypted connections.
- The CRC: a subsection of three bytes used to check the presence of errors in the PDU.

## 6.5. Simulating BLE energy usage

There are many options of of the shelf BLE modules available on the market. In order to make a well considered choice in module, first, the most important attributes of a module need to be determined for this project. I.e. which main parameters should be considered when selecting a module?

To make this determination, simulations are developed in order to analyse which aspects of a BLE module have the most influence on its power usage. In [30], a thorough power profile is made of a BLE module based on power measurements performed during transmission of packets using a BLE module. Additionally, an online power profiler has been released by Nordic Semiconductors [31] which incorporates measurements done by Nordic on 3 of their BLE modules in order to simulate their power usage under different use cases. Both these two sources where used to develop a simulation of the power usage of a BLE module, which can be based on information derived solely from data sheets of modules.

From [31] and [30] it is derived that, during transmission, the module is in 3 states. These states are: transmitting, receiving or switching radio mode (this is also referred to as turnaround). As observed in [30], the module has additional stages per connection event: pre-processing, standby and ramp up. However, these stages only occur once every connection event and contribute, on average, very little to the overall power consumption. In the profile made in [30] these stages contribute only 1,763% of the total energy usage when transmitting 3 packages. For this project, a higher payload will be used of approximately 15 kB. With a maximum theoretical payload of 251 bytes per packet (when using DLE, section 6.4), $15360/251 \approx 61$ packets are needed for transmission. Therefore making the contribution of these stages even less. Additionally, it is not possible to determine the current consumption and duration of these stages when only taking into consideration data sheets. Physical measurements are required to determine their impact. For these two reasons, the simulation will only take into account the consumption during the transmitting, receiving and switch

stages.
The states which a BLE module runs through per transmitted packet are shown in Figure 6.4.

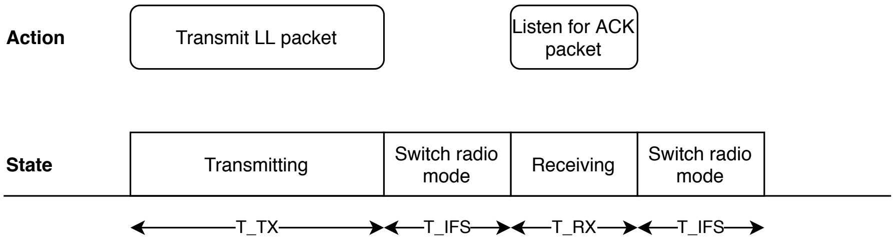



Figure 6.4: Module states when transmitting 1 packet

The times spent in the transmitting and receiving state are dependent on the physical layer (PHY) specifications. BLE has either a transmission speed of 1Mbps or 2Mbps (since Bluetooth 5.0) [24].

$$T_{TX}, T_{RX} = \frac{\#\ of\ bits\ to\ be\ sent}{PHY\ Speed} \tag{6.1}$$

The amount of bits to be transmitted can be determined as follows:

$$N_{full\ packets} = \text{floor}\left(\frac{Payload}{Max\ payload/packet}\right) \tag{6.2}$$

$$N_{bits\ in\ last\ packet} = (\ [Payload] \bmod [Max\ payload/packet]\ ) + Overhead\ bits \tag{6.3}$$

$$N_{TX\ bits} = N_{full\ packets} \cdot LL\ packet\ size + N_{bits\ in\ last\ packet} \tag{6.4}$$

Acknowledgement packets consist of the minimal packet size, thus 0 payload in the PDU.
Therefore:

$$N_{ACK\ bits} = Overhead\ bits \tag{6.5}$$

Consequently for the tranmission and receiving times:

$$T_{TX\ (full\ packet)} = \frac{LL\ packet\ size}{PHY\ Speed}, \quad T_{TX\ (last\ packet)} = \frac{N_{bits\ in\ last\ packet}}{PHY\ Speed}, \quad T_{RX} = \frac{N_{ACK\ bits}}{PHY\ Speed} \tag{6.6}$$

Using the known times spent in each stage, and by extracting the current usage in each stage from the data sheet of a BLE module. The average current consumption per transmitted packet can be calculated.

From most data sheets, the transmitting and receiving currents can be found for BLE modules as well as their idle current. However the turnaround or 'switch' current as used in both [30] and the Nordic Online Power Profiler [31], cannot be found in data sheets. This value can only be measured and is therefore, because of the restrictions of this project, not possible to obtain. For the development of the simulation, the three modules from the Nordic Online Power Profiler will be used (nRF52840, nRF52832 and nRF52810) and consequently the measurements of Nordic can be used in the simulation. Later on, the effect of this switch current on the overall power consumption will be analysed.

#### Implementation of the simulation

For demonstration purposes, a simulation will be described of the nRF52840 from Nordic Semiconductors. The simulation can then be extended by adding different modules with their associated attributes.
The nRF52840 has the following attributes, according to [31]:

Table 6.1: nRF52840 attributes

| Description | Symbol | Value | |
|---|---|---|---|
| Transmission current (0 dBm) | $I_{TX}$ | 6.7 | mA |
| Receiving current | $I_{RX}$ | 6.9 | mA |
| Switch current (TX->RX) | $I_{SW,TR}$ | 3.5 | mA |
| Switch current (RX->TX) | $I_{SW,RT}$ | 3.7 | mA |
| Sleep current | $I_{Sleep}$ | 2.7 | $\mu$A |
| Physical layer speed | PHY Speed | 2 | Mbps |
| Data length extension possible | DLE | True | |

Using the specifications of the nRF52840 and formulas 6.6 a model for the current consumption of the nRF52840 can be made. In case DLE is enabled: LL packets consist of max. 265 bytes and max. payload is 251 bytes. This gives an overhead byte count of $265 - 251 = 14$ bytes. The current for transmitting 3 full LL packets can then be modelled as follows:

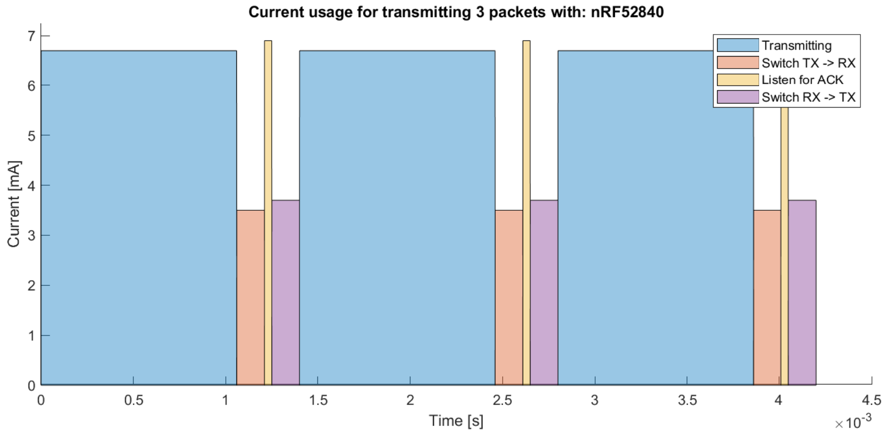



Figure 6.5: nRF52840 current usage for transmitting 3 full LL packets

The average current per full LL packet can be calculated as follows:

$$T_{packet} = T_{TX\ (full\ packet)} + T_{RX} + 2 \cdot T_{IFS}$$

$$I_{full\ packet} = \frac{I_{TX} \cdot T_{TX\ (full\ packet)} + I_{RX} \cdot T_{RX} + (I_{SW,TR} + I_{SW,RT}) \cdot T_{IFS}}{T_{packet}} \tag{6.7}$$

And for the last packet:

$$T_{last\ packet} = T_{TX\ (last\ packet)} + T_{RX} + 2 \cdot T_{IFS}$$

$$I_{last\ packet} = \frac{I_{TX} \cdot T_{TX\ (last\ packet)} + I_{RX} \cdot T_{RX} + (I_{SW,TR} + I_{SW,RT}) \cdot T_{IFS}}{T_{last\ packet}} \tag{6.8}$$

Then

$$T_{active} = N_{full\ packets} \cdot T_{packet} + T_{last\ packet}$$

$$I_{active} = \frac{N_{full\ packets} \cdot T_{packet} \cdot I_{full\ packet} + T_{last\ packet} \cdot I_{last\ packet}}{T_{active}} \tag{6.9}$$

And

$$I_{avg} = \frac{I_{active} \cdot T_{active} + (T_{connection\ interval} - T_{active}) \cdot I_{sleep}}{T_{connection\ interval}} \tag{6.10}$$

By varying the amount of data to be sent, i.e. varying the payload size, the ratio between time spent in active mode and sleep mode is varied. By varying the payload size a simulation can be run, examining the effect of payload size on the total average current of the BLE module. This simulation is run for a connection interval time of 1 hour, as per the mandatory requirements of the project. A simulation is performed for a maximum payload of 120.000 bits, since this is the maximum expected amount of information according to the mandatory requirements, and using the specifications of 4 BLE modules from Nordic Semiconductors: nRF52840, nRF52832, nRF52810 and nRF8001. The results can be found in Figure 6.6.

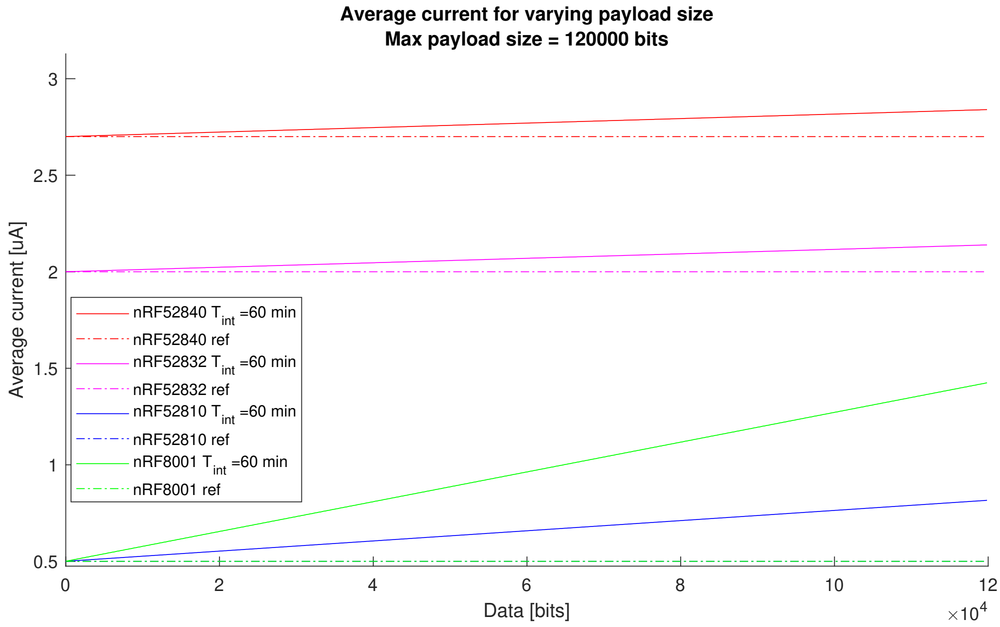



Figure 6.6: Varying payload for different BLE modules

Per module there are two lines plotted: simulated average current (solid line) and the sleep current (dash-dot line for reference purposes, indicated by $\mathrm{nRFXXXXX\ ref}$). The current for the nRF8001 rises much faster for an increasing payload than the other three modules. This is because its transmission and receiving currents are more than twice that of the other modules and since it is a much older model (2015), it maximum PHY speed is 1 Mbps. Consequently increasing both $T_{TX}$ and $T_{RX}$ and thus the time spent in active mode.

The nRF52840, nRF52832 and nRF52810 are more recent models and, as can be seen in the simulation, are more efficient. It is also apparent that the sleep current has the most effect on the overall power consumption for this cyclic sleep scenario. Because of the long sleep times and relatively very short active times (around 0.27 seconds for the nRF52810), the sleep current becomes the most dominant contributor to the overall power consumption.

### Validity model

The previously discussed simulation was derived from both [31] and [30]. In both these models there exists a switch current during radio turnaround. Because this switch current can only be measured and is not indicated on standard data sheets of BLE modules, the effect of this switch current on the overall power consumption must be investigated in order to ascertain its contribution to the average current.

A simulation is run for the three modules which are also used in the Nordic Online Power Profiler [31]. These modules where chosen because their switch currents are known and the validity of the model can easily be checked. For each module, the switch current is varied from 1/10th its actual value to 10x its actual value since these are assumed to be reasonable bounds for any realistic switch current. I.e. for the nRF52840, its switch current is 3,5 mA. Thus the simulation will run from 0,35 mA up to 35 mA for the nRF52840. The simulation is run for a connection interval of 1 hr and with different payload sizes. The results are shown in Figure B.1.

From the simulations in Figure B.1 it can be seen that the switch current has no significant effect on the overall current usage. In the case of the nRF52840 for a payload of 120000 bits, the average current varies by

$$\frac{2.843 - 2.84}{(2.843 + 2.84)/2} = \frac{0.003}{2.8415} = 0.1\%$$

It can therefore be concluded that the switch current contributes very little to the average current and can therefore be set to an arbitrary constant number (within the simulated range) for BLE modules for which the switch current can not be found in their respective data sheets, for simulating their power profile.

#### Analysis of connection interval

To analyse the effect of the connection interval on the average current simulations have been run. This is done to determine if the mandatory requirement of an interval of 60 min is the optimal choice for efficient operation. The results of this simulation can be found in Figure B.2.

As can be seen in the simulations of Figure B.2 after the 1 hr mark, the derivative of the average current vs connection interval time is less than $-0.3 \cdot 10^{-3}\ \mu A/s^2$. Meaning that any further increase in connection interval would not significantly contribute to a lesser power consumption.

### 6.5.1. Choice of BLE module

Using the previously established simulation, additional BLE modules can be compared with each other. In order to make a meaningful comparison, the attributes of different low power BLE modules are collected from data sheets. The modules which will be compared are all branded as ultra low power BLE solutions. The same attributes as described in table 6.1 are extracted from the data sheets of these additional modules. The list of modules for which this is done as well as the corresponding attributes can be found in Table A.2. To make a meaningful comparison between the modules, the reference voltage at which the different currents in the data sheets are given is noted as well. The simulation as described above for the average current versus the payload size is performed for each module in Table A.2. By multiplying the result of Equation 4.2 by the reference voltage $V$, the average power consumption versus payload size can be plotted for each module. The results of this simulation are shown in Figure B.3. In order to avoid confusion and increase readability, the reference lines for each modules sleep current are omitted from the figure. From this simulation. The four most power efficient modules are plotted again in Figure B.4. From this figure, it can be seen that for each payload size on the x-axis, the RSL10 from *ON Semiconductor* [32] is has the lowest average power. This simulation is run for a connection interval of 60 minutes. The switch currents for each module are kept the same at 3.5 mA. As was shown previously, the switch current does not have any significant effect on the average power consumption of the BLE modules and therefore is kept the same for each module.

## 6.6. Results

From this last simulation, it can be shown that the RSL10 from *ON Semiconductor* [32] is the most suitable choice of wireless communication module for this project. It has an extremely low sleep current of 25nA when operating at 3V. Furthermore, it posses UART, SPI and I2C capabilities and includes all Bluetooth Low Energy 5 features like 2Mbps PHY speed and Data Length Extension. It is therefore concluded that this is the ideal choice of wireless communication module and the RSL10 should be implemented in the final design.

The following chapter will cover the development of the software which is needed to operate the system. However, due to time constraints this software is developed for the nRF52810 and not for the RSL10. This is further explained in Section 7.1.

# 7 Software

In the past chapters, the off the shelf components have been chosen so that they best meet the requirements. At this point the hardware design thus best meets the requirements to a large degree. But the way these components are programmed and interconnected also determine the characteristics of the system to a large degree. Therefore the next chapter is devoted to programming and interconnecting the hardware so that it meets the requirements as well as possible. Due to the complexity of both a microcontroller and a BLE module, it is impossible to simulate their behaviour and therefore impossible to test software without the actual devices. Consequently, in order to develop and effectively test software written for these devices, physical hardware devices must be acquired. To this end two development boards have been purchased, one for the microcontroller and one for the BLE module.

## 7.1. Hardware used for software testing

Before moving on to the software it is important to note which hardware was used for testing the software. For the microcontroller, the MSP-EXP430FR5994 development kit was chosen. This is a development kit that uses the MSP430FR5994. From the comparison in Chapter 4, the MSP430FR5994 was found to be the most suitable choice of microcontroller.

During the early stages of simulating BLE power consumption, the nRF52810 was found to be very energy efficient. Due to time constraints related to order times, a choice was made to order the nRF52DK (the development board containing the nRF52810). This choice was made before a complete comparison of modules could be made. Later it was found, as concluded in Chapter 6, that the RSL10 [32] is the ideal choice of module. It should be noted that the nRF52810 used in this section is for software verification purposes only. The final design would include the RSL10.

## 7.2. Choice of wired communication protocols

The sensors, nRF52810 and MSP430FR5994 will be connected with each other via wired communication busses. The communication between each system can take place according to a few different wired communication methods. The choice of communication method between the systems will now be treated.

### 7.2.1. Communication MSP430FR5994 and nRF52810

For the communication method between the MSP430FR5994 and nRF52810, three generally used communication protocols are available. According to the data sheets of both devices (MSP430FR5994 [15], nRF52810 [33]) both modules feature UART, SPI and I2C capabilities. As further explained in Section 7.4, example code is used for the software of the nRF52810. Example code is available to quickly implement the serial communication to the nRF52810 and connectivity over BLE. There is however only one such example available and it makes use of UART for the serial communication method. Because of the small time frame of this project, the choice for UART is made for the purposes of testing the software implementation. Since no time is available to write a complete BLE solution using another serial communication method.

### 7.2.2. Communication MSP430FR5994 and sensors

The digital sensors which where chosen by the subgroup responsible for developing the sensors rely on I2C for digital communication. Therefore communication between the MSP430FR5994 and the digital sensors should take place via I2C. The analogue sensors shall be read by the MSP430FR5994 via its built-in ADCs.

## 7.3. MSP430FR5994 programming

In this section, various programming implementations will be elaborated on. The MSP430FR5994 offers a wide array of operating options and these can be specified by setting registers or calling compiler specific commands. Code Composer Studio [34] is used as an IDE for the MSP430FR5994.

### 7.3.1. Low Power Modes

The MSP430FR5994 offers several Low Power Modes (LPM), ranging from 0 to 5 with two additional LPMs, 3.5 and 4.5. As the LPM modes increase, they consume less power. However, also less peripherals/clocks become available. In order to meet the requirements, it is important to pick a LPM that is as low power as possible but is still able to wake up once an hour. This results in choosing LPM 3.5 which consumes 350nA. This LPM is the lowest power mode that still can use the Real Time Clock, which is a peripheral that allows the microcontroller to wake up on set intervals. The MSP430FR5994 remains in this state the longest.

### 7.3.2. Clocks

The clocks have a large impact on the energy consumption of the MSP430FR5994. Often in the data sheet energy is referred to as $\mu$A/MHz. Therefore it is important to keep clock frequencies as low as possible while still being able to process all necessary data. The Master Clock is the clock that sets the frequency for the core of the MSP430FR5994. The moment the most data passes through the core is when the MSP430FR5994 transmits its data to the nRF52810. At that moment, 120.000 bits need to be transferred. The ideal frequency for this could not be derived due to constraints in time. However, for the master clock frequency 1MHz was chosen since this will allow the data to pass through the core within an acceptable amount of time while still keeping power consumption low.

### 7.3.3. UART

According to the data sheet of the MSP430FR5994 [15, p.108], to use UART, PIN6.0 can be selected as transmitting pin by setting the select registers of port 6. Setting $\mathrm{P6SEL1.0} = 0$ and $\mathrm{P6SEL0.0} = 1$, P6.0 is configured as the transmitting pin for UART. Additionally, baud rate is set for the highest baud rate possible when the MSP430FR5994 is operating at 1 MHz. This highest baud rate is 115200 Hz according to [16, p.782].
When the MSP430FR5994 UART buffer is ready for transmission, it will generate an interrupt flag in register $\mathrm{UCTXIFG}$ [16, p.771].
Data can be sent over UART by loading a byte into the $\mathrm{UCA3TXBUF}$ register, the MSP will clear the interrupt flag on $\mathrm{UCTXIFG}$ to indicate it is transmitting and will transmit whatever is in register $\mathrm{UCA3TXBUF}$ over P6.0 via the UART.

### 7.3.4. I2C

Due to time constraints I2C could not be implemented on the MSP430FR5994 in time.

### 7.3.5. ADC

Due to time constraints ADC could not be implemented on the MSP430FR5994 in time.

### 7.3.6. Finite state machine

The software on the microcontroller should implement the behaviour stated in Chapter 3. It should generate several outputs and respond to inputs at the correct time. This is achieved using the Finite

State Machine (FSM) shown in Figure 7.1. Below each state will be discussed with its function and corresponding implementation on the MSP430FR5994.

#### Idle state

The MSP430FR5994 will spent most of its time in this state. This is the state which is occupied between measurements and because the MSP430FR5994 does not need to communicate with anything during this state it is imperative that its power consumption is as low as possible during this state. The MSP430FR5994 will therefore be put into LPM3.5 during idle state. An RTC will count until 1 hour has passed. It will then exit LPM3.5 and return to active mode and move to the measure state.

#### Measure state

In this state the microcontroller sends out a signal over I2C to the sensors to initiate a measurement. It will then receive measurement data from the sensors over I2C and store these measurements in its FRAM memory.

#### Send data state

In this state, the MSP430FR5994 must send all measurement data to the nRF52810 for transmission over BLE. This is done using the UART protocol. The MSP is configured to use UART as described in Section 7.3.3.

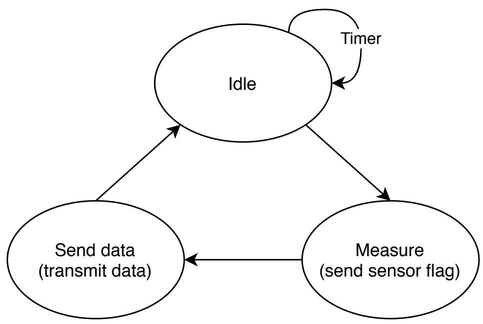



Figure 7.1: Finite state machine of Microcontroller

## 7.4. nRF52810 programming

In this section, the implementation of the program for the nRF52810 will be described. For the nRF52810, the SEGGER Embedded Studio IDE is used for programming. This IDE is downloaded according to the instructions from Nordic Semiconductor [35].

### 7.4.1. Testing of connection

For testing of the connection and data transfer, a smartphone is connected to the nRF52810 via Bluetooth. This is done using the ’nRF Connect’ app from Nordic Semiconductor [36]. Using this app, a precise log of the connection can be produced and this enabled better debugging of errors.

### 7.4.2. Code implementation

For the nRF52810, example code from the software development kit has been used. This enabled quick development and testing time since it is not practical to write a whole BLE software solution from scratch in the time allotted for this project. An example project was used which enable the nRF52810 to receive data over UART and transmit this data over BLE to a central device. The code which was used for this implementation can be found in Appendix C.3. A few modifications were made to the original code of the example project. The baud rate was set to match that of the MSP420FR5994, namely 115200 Hz. Additional modifications were needed to prevent the nRF52810 from overflowing during transmission. This was due to the fact that without modifications, the nRF52810 would overflow from the data from the MSP430FR5994. Two solutions to help this where implemented:

Connection interval
The connection interval was lowered in order for the UART buffer of the nRF52810 to be emptied as fast as possible. The suggested connection interval from the 'nRf Connect' app for a high priority connection was used. This suggested connection interval is 11.25-15 ms. Therefore the minimum and maximum allowed connection intervals of the nRF52810 where modified to assume this suggested range for intervals. This modification can be found in Listing C.1

UART flow control
To prevent the UART receiving buffer of the nRF52810 from overflowing, flow control between the two devices was enabled. Using flow control, the nRF52810 can give a signal to the MSP430FR5994 to indicate it is ready to receive information. This way, the MSP430FR5994 is aware of when the receiving buffer is full and will temporarily stop transmitting to prevent data loss. When the nRF52810 indicates it is ready to receive information again, the MSP430FR5994 will resume transmitting. The modification to enable flow control can be found in Listing C.2

## 7.5. Results

### 7.5.1. MSP430FR5994

The MSP430FR5994 software needed to implement a finite state machine. It was supposed to stay in an idle state and wake up using the RTC, measure data/receive data via I2C or ADC and transmit data by communicating to the nRF52810 using UART and then return to the idle state.
Due to a lack in time not all software was implemented. What has been implemented is a program which can stay in LPM3.5 for around 8 minutes, then wake up using a RTC interrupt and transmit 120kbits, that were pre-programmed in the FRAM memory, to the nRF52810 via UART and then return to LPM3.5. The code for this can be found on the GitHub described in C.2. What has not been implemented is the communication with the sensors using I2C, measuring the ADC.
Then Energytrace was used. Energytrace is a program made by Texas Instruments that can be used on development kits to measure the current consumption of the microcontrollers. With Energytrace, it was possible to do measurements on the existing code. The goal is to make a comparison between the predicted currents in Chapter 4 and the actual measured currents. Since only UART, low power, and active mode current were implemented, only a comparison of these currents can be made.
Figure B.5 shows the power consumed during two UART transmission events, Figure B.6 zooms in on one of these events. With the given baud rate of 115200Hz, the transmission should take approximately 1.5 seconds. In the plot this is 1.8765 seconds. During this interval the average power is 626$\mu$W. At this interval, the microcontroller is in active mode and uses UART. In chapter 4 it was said that the active mode current was 118$\mu$A/MHz and the UART current was 6.3$\mu$A/MHz. At 1 MHz and 3V this results in a power of 372.9$\mu$W. So the UART and active mode power is 1.68 times higher than predicted through the datasheets.
Figure B.7 shows the power measured for LPM3.5, which is 1.640$\mu$W. Chapter 4 predicted that the LPM3.5 current would be 350nA at 3V, which is 1.05$\mu$W. So the LPM3.5 power is 1.56 times higher than predicted through datasheets.
Several measurements are missing in this analysis due to time constraints. However, it can be noted that the power consumption will be approximately 1.6 times higher than predicted in Chapter 4. This can probably be solved partially by optimising code. Code Composer Studio offers an ultra low power advisor that can help solve these problems, but they have not yet been tackled in this software.

### 7.5.2. nRF52810

It was decided that the MSP430FR5994 would communicate with the nRF52810 using UART. This was successfully implemented but due to a lack of time the code could not be fine tuned. Thus the nRF52810 is able to receive the code via UART and then transmit it to a mobile phone using BLE. Accurate current measurements were not possible due to lack of necessary measurement equipment. Current measurements for this device would however not be relevant in any case. That is because the nRF52810 is used for verification and in the actual design, the better performing RSL10 should be used.

## 7.6. Future work / improvements

Further improvements to the software implementation are required, such as:

1. The BLE module should be put to sleep when the microcontroller goes into LPM. Furthermore, the microcontroller should then 'wake up' the BLE module when it exits LPM and is ready to transmit measurement data.
2. I2C should be implemented so that communication with the sensors is possible
3. The ADCs should be implemented so that they can measure
4. The interval of the FSM should be changed to 1 hour
5. The MSP430FR5994 software should be further optimised with Ultra Low Power software of Texas Instruments
6. The software for the RSL10 should be developed once this module has been added to the design
7. More research needs to be done to which protocols are more efficient: I2C, UART, SPI

# 8 Results

## 8.1. Microcontroller and communication design

After having performed simulations, tests and measurements, a conclusion can be made about the achieved results in the scope of the microcontroller and communication subsystem. To summarise:
The optimal off-the-shelf components for the subsystem have been determined: for the microcontroller this is the MSP430FR5994 [15] and for the BLE module this is the RSL10 [32]. These two components are the most power efficient in their respective areas. Additionally they meet all mandatory requirements as stated in Chapter 2.
A first iteration of the software for the MSP430FR5994 has been developed. It is able to bring the MSP430FR5994 in Low Power Mode 3.5 which is its mode of operation which uses the least amount of power, while still running a RTC. It is also able to communicate large amounts of data via UART to another device and return to the LPM after executing its tasks. An overview of the achieved system as implemented in Chapter 7 is given in Figure 8.1. The inputs *Analogue sensors* and *Digital sensors* have been simulated by randomly generated data.

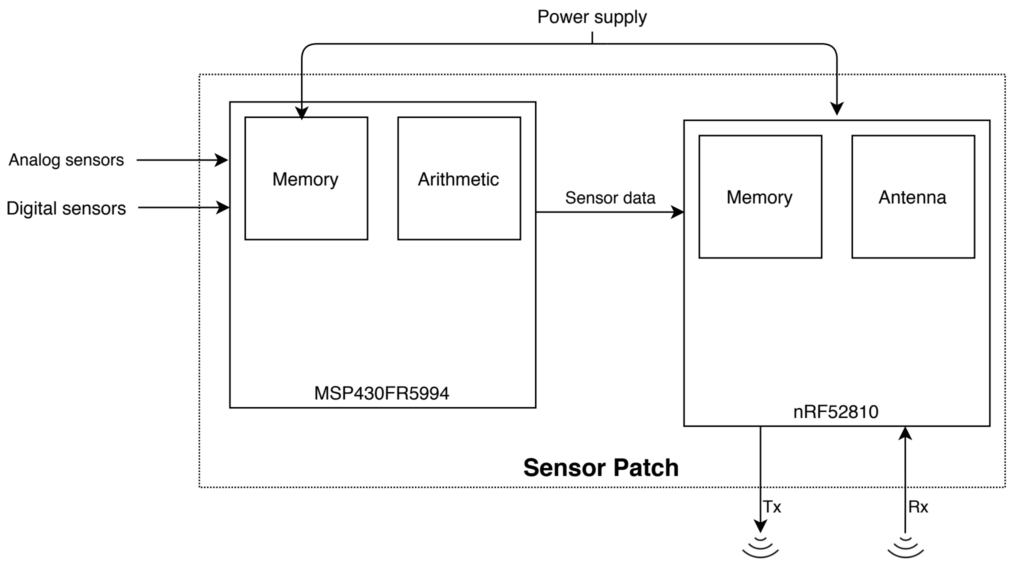



Figure 8.1: Schematic overview of designed system

## 8.2. Overall design

This chapter discusses what the prototype should look and perform like. First, the physical characteristics will be discussed. Then a section will be devoted to how the system should be used.

## 8.3. Physical characteristics prototype

Figure 8.2, 8.3 and 8.4 show the top view, side view and bottom view respectively of the design of the sensor patch. The total size should be 3x7cm since this was recommended by Dr. Dudink [11]. The orange parts are flexible wires that interconnect the different parts. These flexible wires allow the prototype to be closer to the skin of the patient. Since the device needs to be reusable, it should be possible to clean the device. Therefore the device should be water- and alcohol proof. To this end, a silicon layer will be applied over the entire device. This will result in a flexible reusable device. Additionally there will be 2 contact points on the silicon casing through which the battery can be charged.

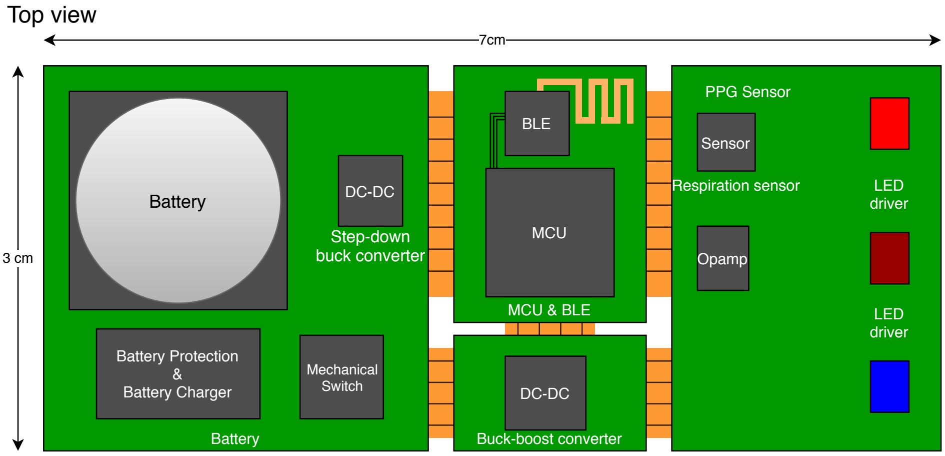



Figure 8.2: Top view of physical prototype

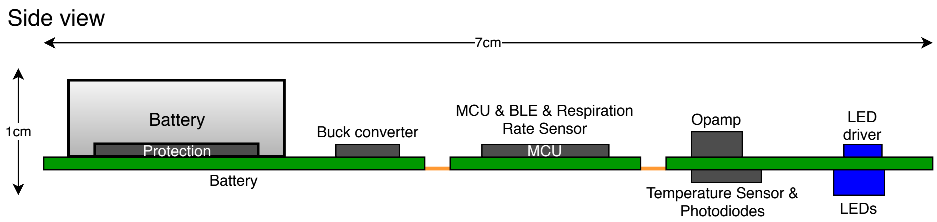



Figure 8.3: Side view of physical prototype

## 8.4. Results of final design

The photoplethysmography (PPG) sensor measuring heart rate, oxygen saturation and bilirubin levels has been designed to make accurate measurements theoretically possible. This was based on the following: sensor properties from papers describing similar PPG sensors, LED radiated power, attenuation and diffusion factors of the skin, photo-diode sensitivity and output currents and noise due to the amplification of the photo-diode output. However as no physical prototype has been made, no measurements could be performed to prove the accuracy and measurement capabilities when performing measurements on patients.
The temperature sensor (Si7051) is able to perform accurate measurements within 0.1 degrees Celsius deviation between measured skin temperature and actual skin temperature. There is a correlation between skin temperature at the chest and internal core temperature [37]. Heart rate and chest movements are used for the correlation model and enable noninvasive methods to replace invasive methods to monitor patient core temperature.
The acceleration sensor (IIS2DLPC) produces 3 dimensional acceleration data with a acceleration resolution of 0.224mg. Using algorithms that remove large body movement components, a method for estimating respiration rate can be used to measure respiration rate [38]. Studies have shown

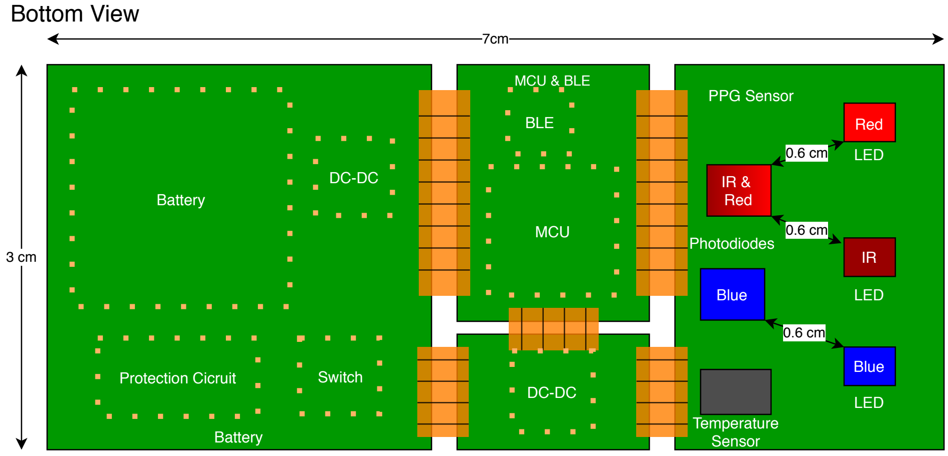



Figure 8.4: Bottom view of physical prototype

accurate respiration rates can be measured when using a 3D acceleration sensor with a sensitivity of 0.2mg and the appropriate signal processing techniques [39].
The total system consumes an estimated 52J in 3 days when it is attached to a newborn while measuring once every hour. This energy consumption is the sum of deep sleep, wake up, measurement, processing and transmission stages of every device separately. A visualisation can be seen in Figure D.3. The sensor patch can last an estimated 3 weeks on one battery cycle.
The system is able to communicate its measurements wirelessly to a Bluetooth receiving device. This can either be a mobile phone, or some sort of base station which should be developed further.

## 8.5. Conceptual use case

### 8.5.1. Data access

In terms of usability, there are a few factors which should be considered. Firstly, this system will measure and transmit sensitive medical data. Only doctors or nurses, who would otherwise work in the maternity ward of the hospital shall be allowed insight into this information. Parents should not be able to view the measurements from the patch since they are not trained to interpret this medical information and the risk of unnecessary panic would be great. The purpose of this project is to replace maternity care and offer the possibility of medical monitoring from home, but this monitoring shall still only be interpreted by medical professionals.

### 8.5.2. Application of patch on skin

The first time the sensor patch is attached will be in maternity care. This will be done by a nurse. This way it is ensured that the sensor patch will be applied properly the first time, before the baby goes home. As described in further detail in Section 9.1, the patch will have an indicator which indicates if it is applied properly. Because of this indicator, parents at home could also reapply the patch properly in case this is needed.

### 8.5.3. Charging

The charging of the sensor patch will be done by trained individuals, to avoid accidental overcharging or damage to the system. This charging shall be done at the same location as the sterilisation of the devices as described in Section 8.6.

## 8.6. Sterilisation methods

The sensor patch can be categorised as a Semi-critical instrumentation, which means that it needs to be cleaned and disinfected [40]. Sterilisation often happens with alcohol and/or steam. The device will have a silicon layer around it as mentioned earlier. For this application, alcohol can be used to sterilise because silicone has good chemical resistance for alcohol [41].

## 8.7. Conclusion

As can be seen in Figure 8.2 the final design of the sensor patch does not exceed the required dimensions mentioned in 2. Furthermore, the designed sensor patch is able to measure the heart rate, temperature, oxygen saturation, respiration and bilirubin. The battery is able to supply enough power to perform measurements every 2 hours for 3 days. Moreover, the sensor patch is able to communicate wirelessly with an external device in order to send the measured data to a hospital where a doctor can access this data.

## 8.8. Future work

Some aspects of the design have not yet been addressed in this paper. In order to implement the sensor patch successfully, some aspects still need to be researched. They will be discussed in this chapter section.

### 8.8.1. Patch contact indicator

In order to receive accurate measurements from the sensor patch, it needs to be attached properly. Otherwise, the sensors might might be too far away from the skin for accurate measurements. For this purpose, further work needs to be done into the development of some sort of indicator which will inform the user whether the patch is applied properly to the skin, so that the user can expect correct measurements.

### 8.8.2. PCB

A concept for the construction of a PCB and component placement has been made, however a design and testing procedure must take place in the future to realise a PCB with the correct placement and operation of components. The PCB design has to take into account aspects of comfort and durability to fulfil design requirements. Flexible PCBs or flexible cables need to be explored to see if the sensor patch can be somewhat flexible to make it fit better to the skin of the patient.

### 8.8.3. PCB housing

In order to protect the sensor patch from external influences, the patch needs a casing. This case must be completely closed and non transparent in order to block the ambient light which could influence the PPG sensors. However, this casing should not interfere with the Bluetooth communication. Furthermore, the casing should protect the electronics of the sensor patch from moisture like sweat. In order to increase user comfort the casing should be flexible. Also, charging points will need to be integrated in the PCB housing.

### 8.8.4. Method of attachment

No testing has been performed on how to apply the sensor patch to the patient. Some solutions have been conceived such as: using an adhesive material or tape to secure the sensor patch. These methods need to be explored and tested in the future to determine their capabilities of attaching the sensor patch for a period of three days. Water resistance can be a useful property for the method of attachment as this would enable the patient to take a bath while wearing the sensor patch and would prevent the patch from losing grip if the patient sweats.

### 8.8.5. Clinical trials

In order to calibrate the designed sensors, clinical trials on patients should be performed. The trials can be used to remove systematic errors in the measurements. Furthermore trials would also verify the performance of the sensors, for example the accuracy and repeat-ability of measurements. The trials will reveal potential issues in the design and may lead to the need of design alterations.

# 9 Recommendations

After concluding the results of this project, a few recommendations have come to mind for future research:

1. Micro controllers offer the option to turn off Real time clock (RTC) which lowers the power consumption by 80-90% in some cases. Using this in combination with a low power RF wake up module might result in lower power consumption.
2. Make a comparison about the ideal clock frequencies. This research chose a low clock frequency to minimise the current drawn, but this results in devices staying on longer. A good analysis about what clock frequencies result in the total lowest energy consumption is recommended.
3. Further research into the benefits of processing of the data on the microcontroller should be conducted. By performing arithmetic on the measurement data before transmission instead of after, the use of the wireless communication module is reduced. This could result in even lower power consumption overall.
4. Further research into the benefits of data compression should be performed. By compressing the measurement data before transmission, less data needs to be transmitted, thus lowering the power consumption of the BLE module. This most likely does result in an increase in microcontroller power usage. Research into the possible benefits of this should be done.
5. For the purposes of fast and easy prototyping, a microcontroller with very large memory was selected. However, additional models of the MSP430FR5994 exist with less memory. These modules could be a better choice in case the program is not very large.

## 9.1. Future work for overall design

### 9.1.1. Patch contact indicator

In order to receive accurate measurements from the sensor patch, it needs to be applied properly. Otherwise, it could be that the sensors do not make sufficient contact with the skin for accurate readings. For this purpose, further work needs to be done into the development of some sort of indicator which will inform the user if the patch makes sufficient contact with the skin in order to perform accurate measurements.

### 9.1.2. Low power indicator

To inform the user ahead of time if the battery has low charge, an indicator should be implemented. This can be as simple as a LED. This will inform the user on time if the sensor patch is about to fail and will prevent accidental loss of measurements.

# A
# Communication

Table A.1: Comparison of communication modules, with data from datasheets

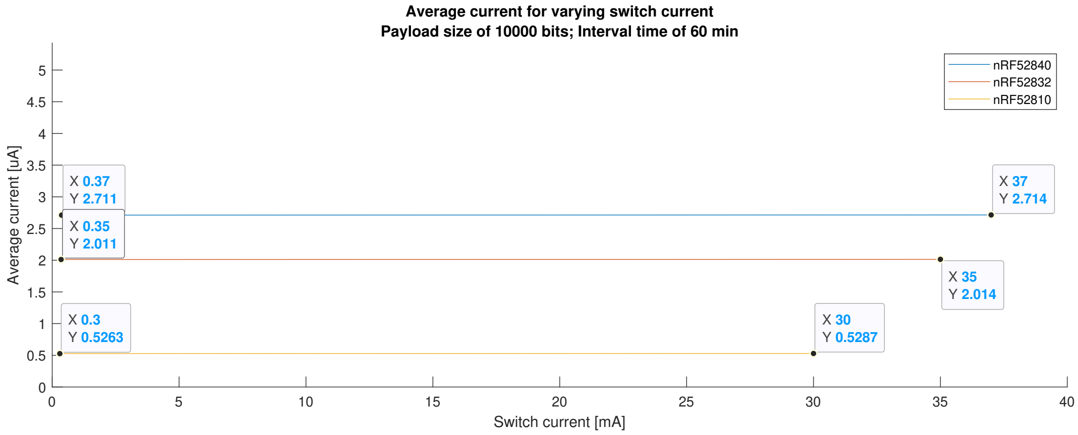
| | Dimensions | | | Power | | Results | | |
|---|---|---|---|---|---|---|---|---|
| | Height (mm) | Width (mm) | Operating Voltage (V) | Sleep current (uA) | TX Current (mA) | Area (mm^2) | TX Power | Sleep power |
| **BLE** | | | | | | | | |
| Tayo Yuden EYSHSN | 3,25 | 8,55 | 3 | 1,2 | 7,5 | 27,7875 | 2,25E-02 | 3,60E-06 |
| SmartBond TINY DA14531 | 1,7 | 2 | 3 | 0,75 | 3,5 | 3,4 | 1,05E-02 | 2,25E-06 |
| MtM M905 | 6,5 | 6,5 | 3,3 | 1,2 | 7,1 | 42,25 | 2,34E-02 | 3,96E-06 |
| ISP091201 | 8 | 12 | 3 | 2 | 12,7 | 96 | 3,81E-02 | 6,00E-06 |
| Silabs BGM13S22F512GA | 6,5 | 6,5 | 3,3 | 1,4 | 8,9 | 42,25 | 2,94E-02 | 4,62E-06 |
| Silabs EFR32BG22 | 4 | 4 | 3 | 1,4 | 4,1 | 16 | 1,23E-02 | 4,20E-06 |
| u-blox ANNA-B112 | 6,5 | 6,5 | 3 | 2,2 | 5,3 | 42,25 | 1,59E-02 | 6,60E-06 |
| Wi2Wi WC7220B0 | 8 | 12 | 3,3 | 0,9 | 8 | 96 | 2,64E-02 | 2,97E-06 |
| | | | | | | | | |
| **Zigbee** | | | | | | | | |
| Digi XBee 3 Zigbee 3.0 | 13 | 19 | 2,85 | 2 | 40 | 247 | 1,14E-01 | 5,70E-06 |
| E75-2G4M10S | 16 | 26 | 3,3 | 1 | 30 | 416 | 9,90E-02 | 3,30E-06 |
| Telit ZE51 | 26 | 15 | 3,3 | 1 | 35 | 390 | 1,16E-01 | 3,30E-06 |
| EFR32MG22 (SoC) | 4 | 4 | 3 | 1,4 | 4,1 | 16 | 1,23E-02 | 4,20E-06 |
| | | | | | | | | |
| **Wi-Fi** | | | | | | | | |
| GS2200M | 13,5 | 17,85 | 3,3 | 8 | 147 | 240,975 | 4,85E-01 | 2,64E-05 |
| Silabs AMW006 | 20,3 | 15,2 | 3,3 | 3,8 | 11,4 | 308,56 | 3,76E-02 | 1,25E-05 |
| TI CC3200 | 9 | 9 | 3,3 | 4 | 166 | 81 | 5,48E-01 | 1,32E-05 |

Table A.2: BLE modules used in the simulated comparison

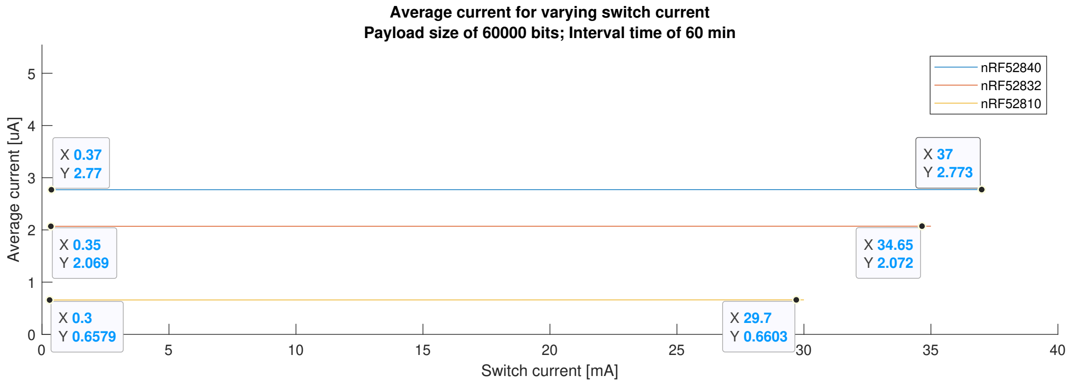
| Model name | Max PHY Speed [Mbps] | DLE | Voltage [V] | Transmit current [mA] | Receive current [mA] | Sleep current (no ram retention) [$\mu$A] |
|---|---|---|---|---|---|---|
| **nRF52840** | 2 | TRUE | 3,00 | 6,40 | 6,26 | 0,40 |
| **nRF52832** | 2 | TRUE | 3,00 | 7,20 | 6,50 | 0,30 |
| **nRF52810** | 2 | TRUE | 3,00 | 5,80 | 6,10 | 0,30 |
| **nRF8001** | 1 | FALSE | 3,00 | 12,70 | 14,60 | 0,50 |
| **AMS001** | 1 | FALSE | 3,00 | 10,80 | 12,80 | 0,65 |
| **EYSHSNZWZ** | 2 | TRUE | 3,00 | 7,50 | 5,80 | 0,30 |
| **QN908x** | 2 | TRUE | 3,00 | 3,50 | 3,50 | 1,00 |
| **DA14531** | 2 | TRUE | 3,00 | 3,50 | 2,20 | 0,27 |
| **M905** | 2 | TRUE | 3,00 | 7,10 | 6,50 | 0,70 |
| **BGM13S** | 2 | TRUE | 3,30 | 8,90 | 10,50 | 0,07 |
| **ATBTLC1000-QFN** | 2 | TRUE | 3,30 | 3,91 | 5,24 | 0,05 |
| **EFR32BG22** | 2 | TRUE | 3,30 | 4,10 | 3,60 | 0,17 |
| **ANNA-B112** | 2 | TRUE | 3,30 | 5,30 | 5,40 | 0,30 |
| **CC2640** | 1 | TRUE | 3,00 | 6,10 | 5,90 | 0,10 |
| **RSL10** | 2 | TRUE | 3,00 | 4,60 | 3,00 | 0,05 |

# B
# Simulation results

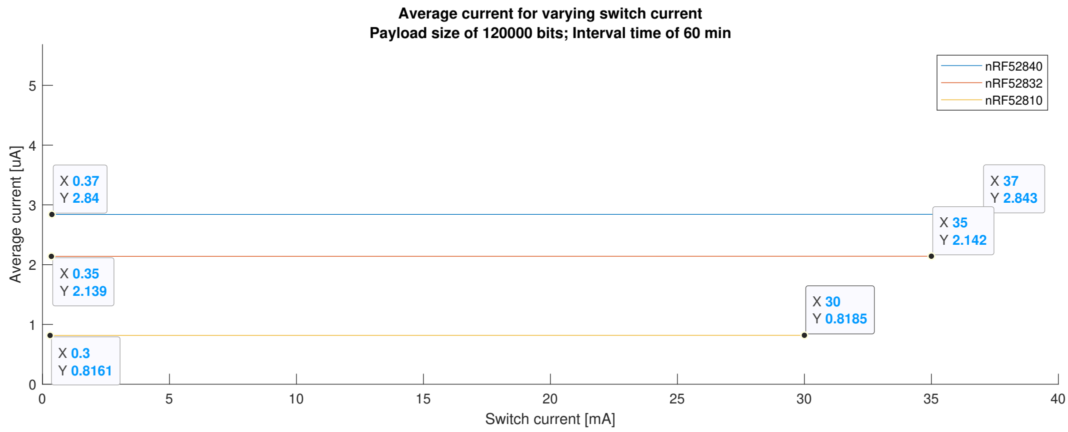
## B.1. Bluetooth Low Energy simulations




(a) Payload size of 10.000 bits

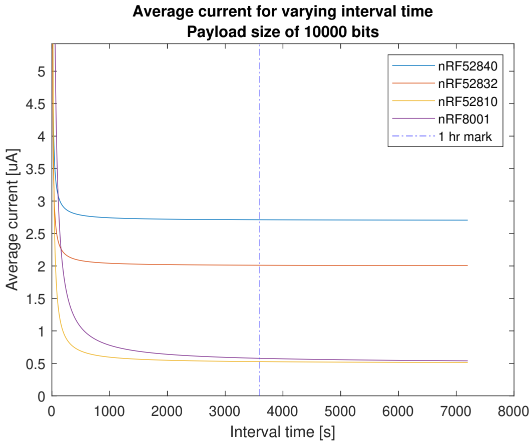



(b) Payload size of 60.000 bits

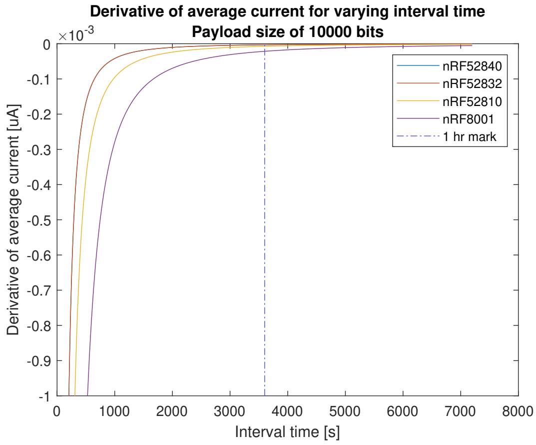



(c) Payload size of 120.000 bits

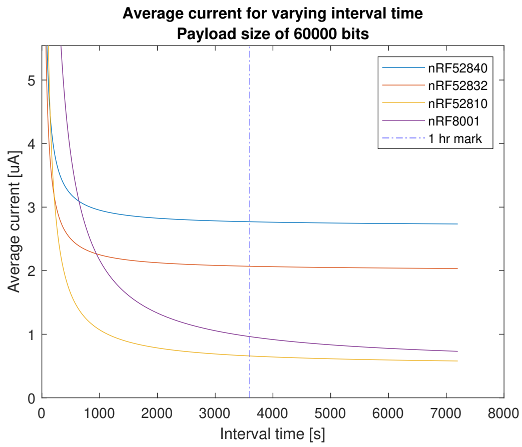
Figure B.1: Simulations of average current vs. switch current for nRF52840, nRF52832 and nRF52810



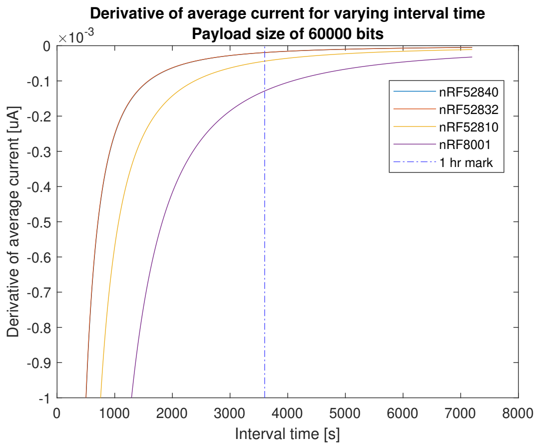

(a) Payload = 10.000 bits



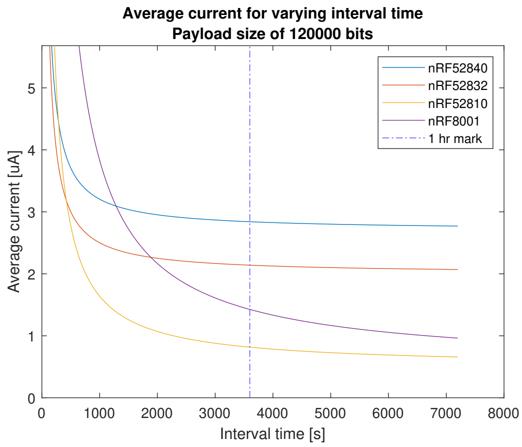

(b) Derivative of average current



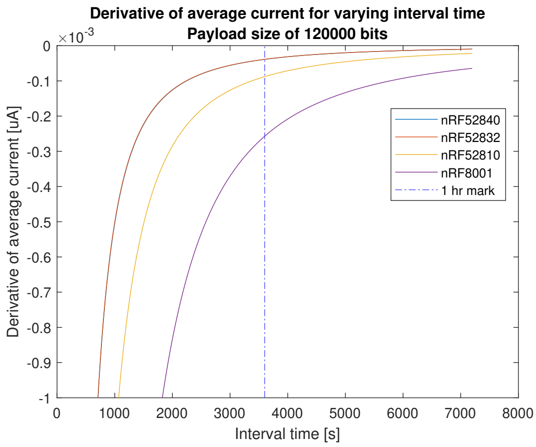

(c) Payload = 60.000 bits




(d) Derivative of average current




(e) Payload = 120.000 bits




(f) Derivative of average current

Figure B.2: Simulations of average current vs. interval time for nRF52840, nRF52832, nRF52810 and nRF8001

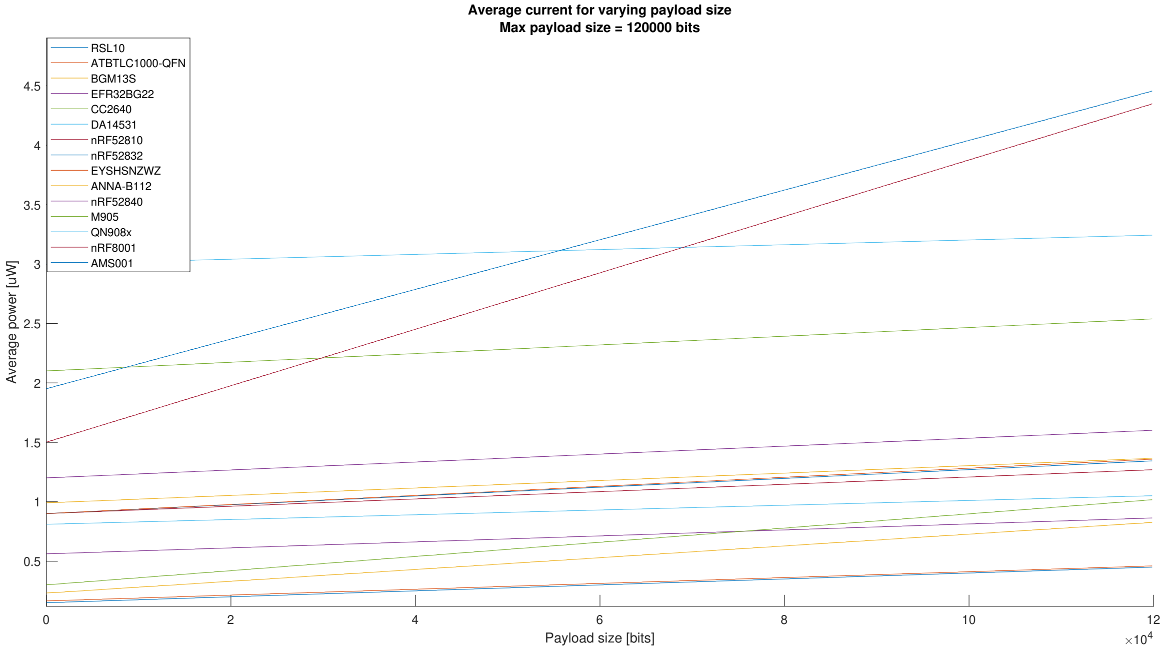



Figure B.3: Simulated average power for 15 different BLE modules

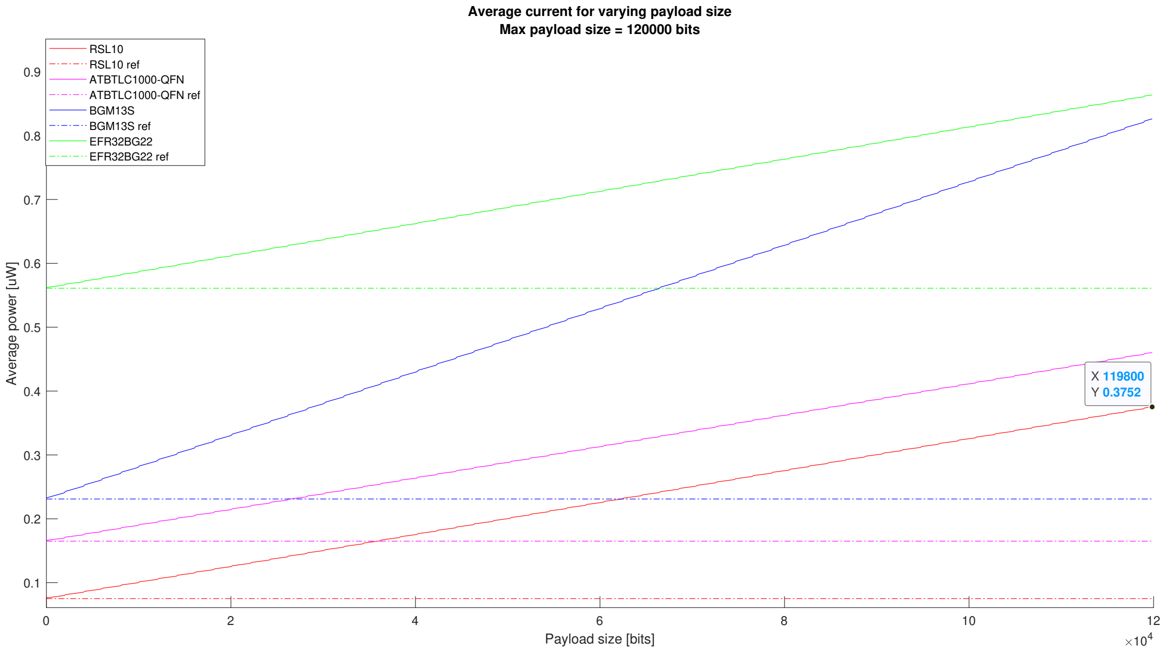



Figure B.4: Simulated average power for 4 most efficient BLE modules

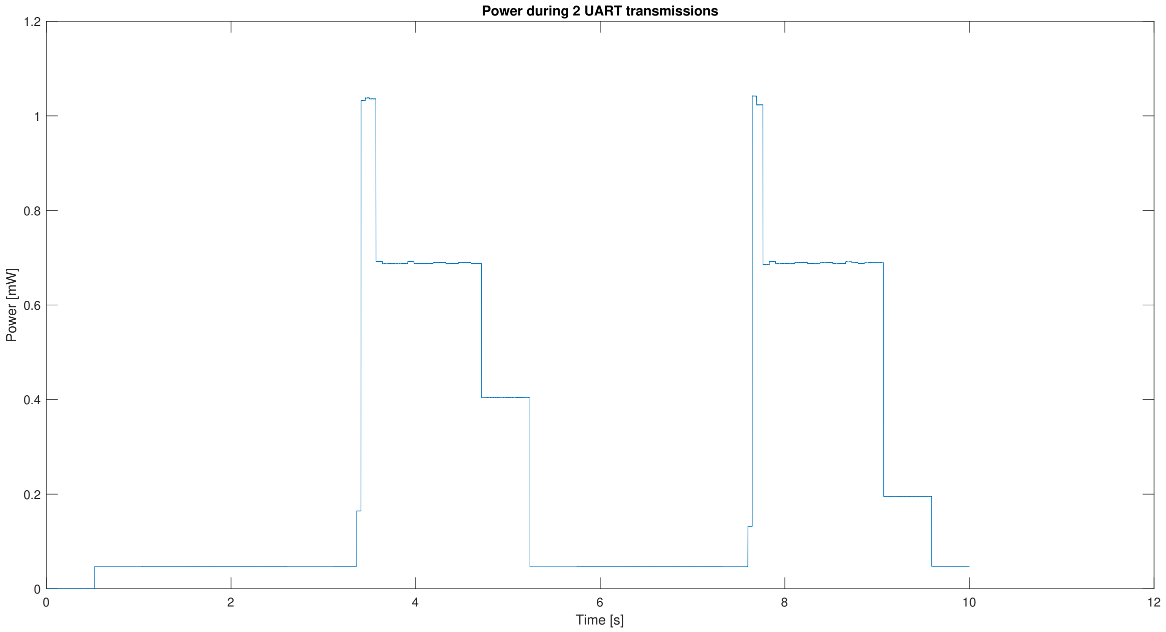



Figure B.5: Measured power consumption MSP430FR5994 when in Active Mode using UART. Plot shows transmitting twice via UART

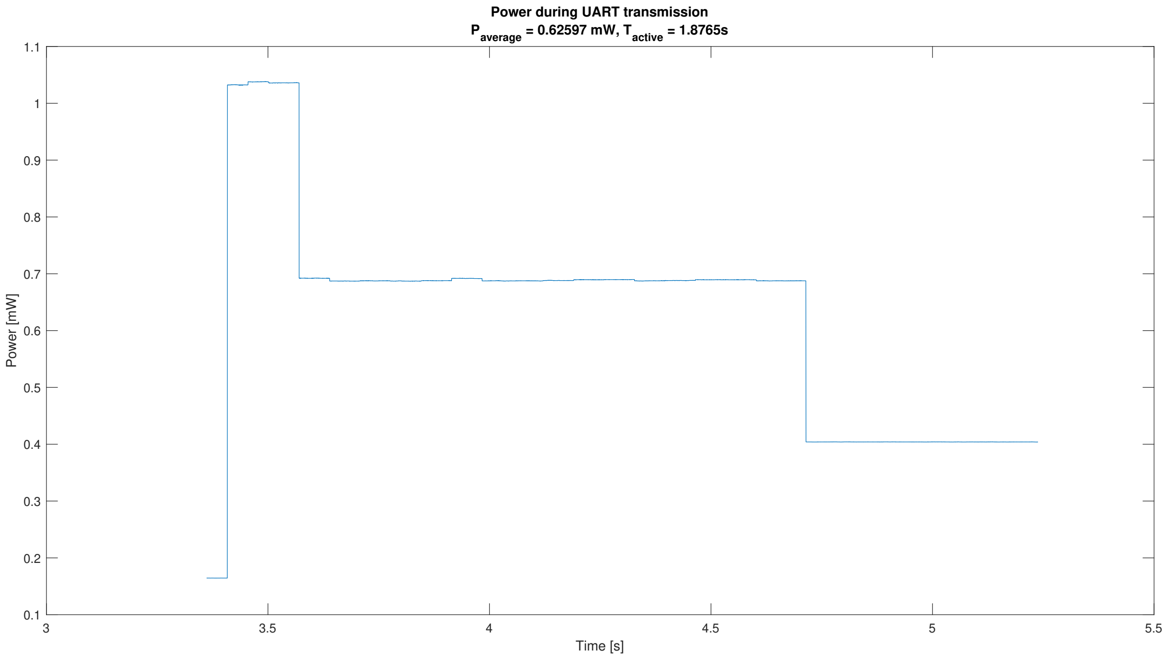



Figure B.6: Measured power consumption MSP430FR5994 when in Active Mode using UART, zoom-in on one UART event

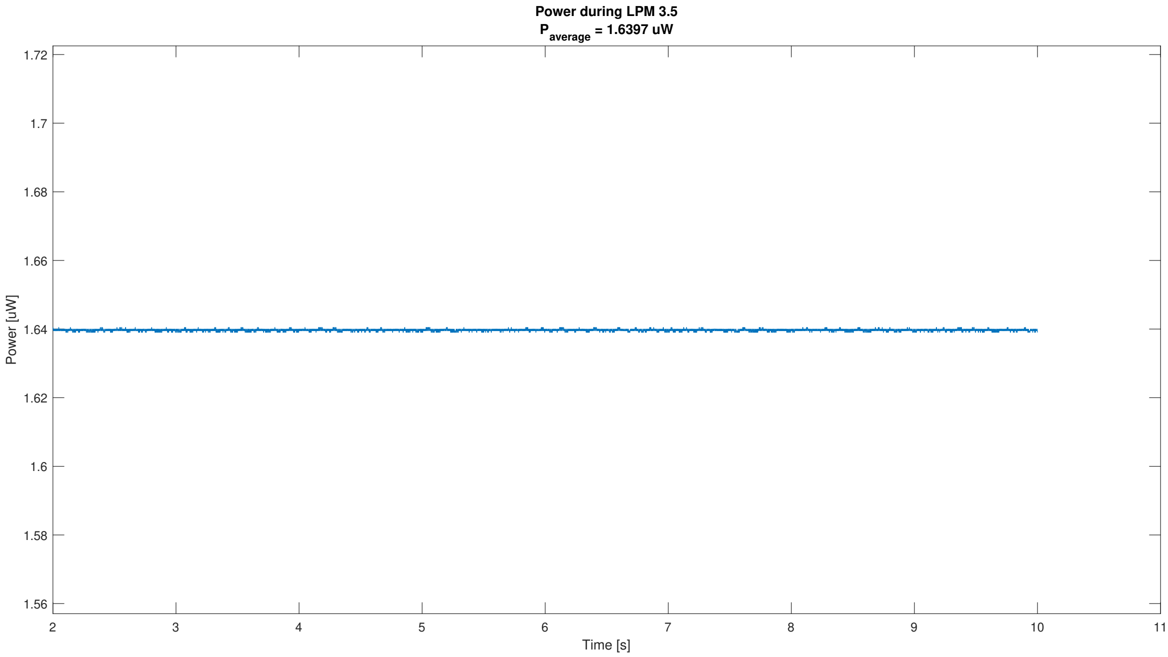



Figure B.7: Measured power consumption MSP430FR5994 when in LPM 3.5

# C
# Program code

## C.1. MATLAB

### C.1.1. BLE simulations

The MATLAB script files used for simulating the power consumption of BLE modules can be found at: https://github.com/ggianfran/Sensor-patch-communication-and-microcontroller/tree/master/Matlab/BLE

### C.1.2. MCU simulations

The MATLAB script files used for simulating the power consumption of MCUs can be found at: https://github.com/ggianfran/Sensor-patch-communication-and-microcontroller/tree/master/Matlab/MCU

## C.2. MSP430FR5994

The Code Composer Studio project files can be found at: https://github.com/ggianfran/Sensor-patch-communication-a tree/master/MSP430FR5994

## C.3. nRF52810

The Software Development Kit for the nRf52810 can be downloaded from: https://www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK/Download

The project files can be found under: examples/ble_peripheral/ble_app_uart/pca10040e/s112/

In listing C.1 and C.2, the only modified lines of code are shown.

Listing C.1: Defines for nRf52810 program

```
#define DEVICE_NAME                     "Sensor Patch"                               /**< Name of device. Will be included in the advertising data. */
#define NUS_SERVICE_UUID_TYPE           BLE_UUID_TYPE_VENDOR_BEGIN                  /**< UUID type for the Nordic UART Service (vendor specific). */

#define APP_BLE_OBSERVER_PRIO           3                                           /**< Application's BLE observer priority. You shouldn't need to modify this value. */

#define APP_ADV_INTERVAL                64                                          /**< The advertising interval (in units of 0.625 ms. This value corresponds to 40 ms). */

#define APP_ADV_DURATION                10000                                       /**< The advertising duration (180 seconds) in units of 10 milliseconds. */

#define MIN_CONN_INTERVAL               MSEC_TO_UNITS(10, UNIT_1_25_MS)             /**< Minimum acceptable connection interval (20 ms), Connection interval uses 1.25 ms units. */
```

```
#define MAX_CONN_INTERVAL               MSEC_TO_UNITS(15, UNIT_1_25_MS)            /**< Maximum acceptable connection interval (75 ms), Connection interval uses 1.25 ms units. */
#define SLAVE_LATENCY                   0                                          /**< Slave latency. */
#define CONN_SUP_TIMEOUT                MSEC_TO_UNITS(4000, UNIT_10_MS)            /**< Connection supervisory timeout (4 seconds), Supervision Timeout uses 10 ms units. */
#define FIRST_CONN_PARAMS_UPDATE_DELAY  APP_TIMER_TICKS(5000)                      /**< Time from initiating event (connect or start of notification) to first time sd_ble_gap_conn_param_update is called (5 seconds). */
#define NEXT_CONN_PARAMS_UPDATE_DELAY   APP_TIMER_TICKS(30000)                     /**< Time between each call to sd_ble_gap_conn_param_update after the first call (30 seconds). */
#define MAX_CONN_PARAMS_UPDATE_COUNT    3                                          /**< Number of attempts before giving up the connection parameter negotiation. */

#define DEAD_BEEF                       0xDEADBEEF                                 /**< Value used as error code on stack dump, can be used to identify stack location on stack unwind. */

#define UART_TX_BUF_SIZE                1024                                       /**< UART TX buffer size. */
#define UART_RX_BUF_SIZE                1024                                       /**< UART RX buffer size. */
```

Listing C.2: Initialisation of UART for nRf52810 program

```
static void uart_init(void)
{
    uint32_t                     err_code;
    app_uart_comm_params_t const comm_params =
    {
        .rx_pin_no    = RX_PIN_NUMBER,
        .tx_pin_no    = TX_PIN_NUMBER,
        .rts_pin_no   = RTS_PIN_NUMBER,
        .cts_pin_no   = CTS_PIN_NUMBER,
        .flow_control = APP_UART_FLOW_CONTROL_ENABLED,
        .use_parity   = false,
#if defined (UART_PRESENT)
        .baud_rate    = NRF_UART_BAUDRATE_115200
#else
        .baud_rate    = NRF_UARTE_BAUDRATE_115200
#endif
    };

    APP_UART_FIFO_INIT(&comm_params,
                       UART_RX_BUF_SIZE,
                       UART_TX_BUF_SIZE,
                       uart_event_handle,
                       APP_IRQ_PRIORITY_LOWEST,
                       err_code);
    APP_ERROR_CHECK(err_code);
}
```

# D

# Figures

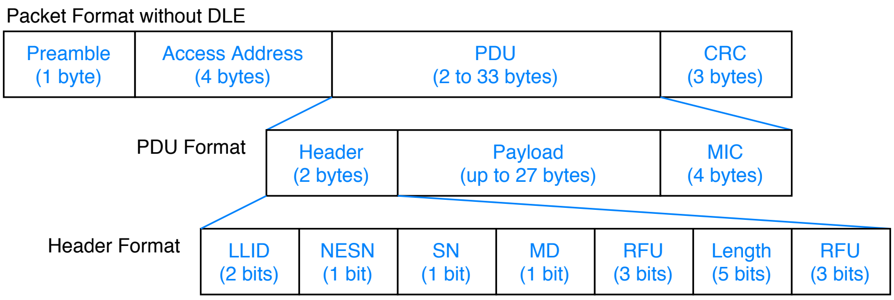



Figure D.1: LL packet make-up without DLE

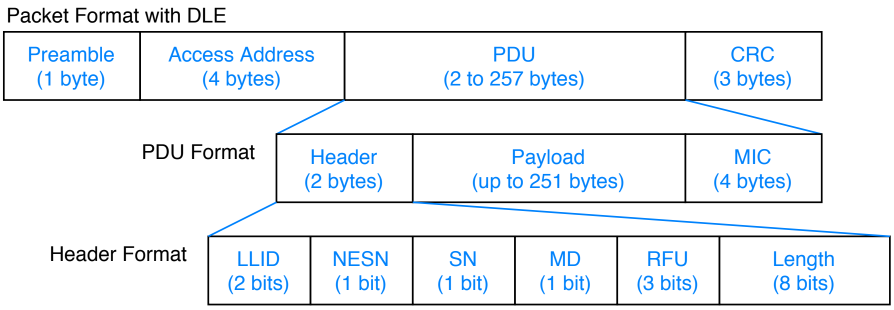



Figure D.2: LL packet make-up with DLE

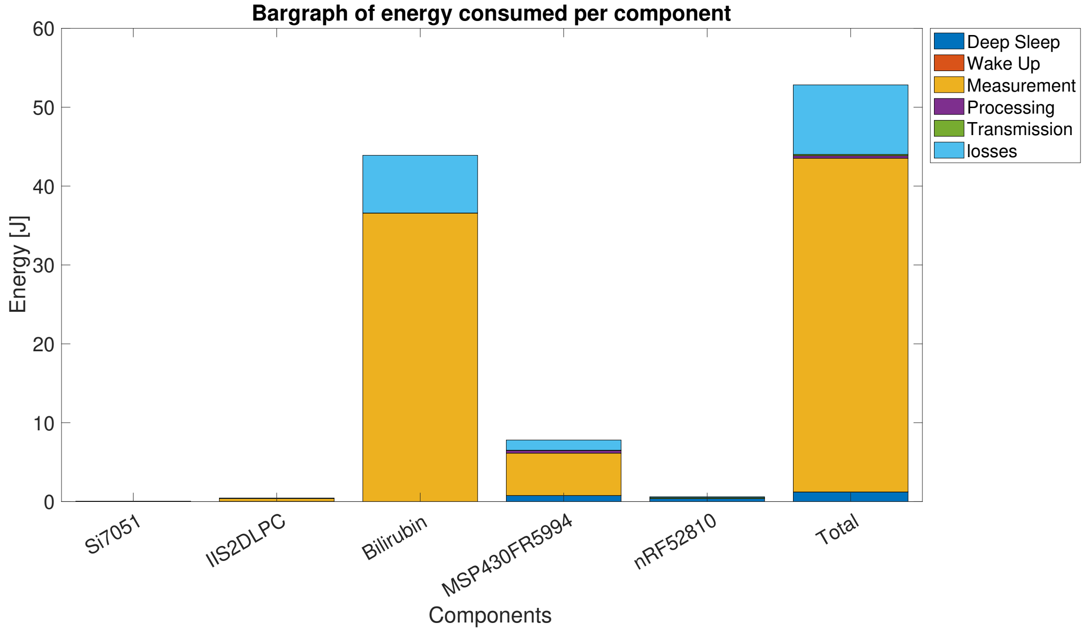



Figure D.3: Energy usage per component divided per stage in the full system simulation.

# Bibliography


[1] N. Rutter and L. Doyal, "Neonatal care and management of pain: historical and ethical issues," *Seminars in Neonatology*, vol. 3, no. 4, pp. 297–302, 1998.

[2] R. Flacking, L. Lehtonen, G. Thomson, A. Axelin, S. Ahlqvist, V. H. Moran, U. Ewald, F. Dykes, and the SCENE group, "Closeness and separation in neonatal intensive care," *Acta Paediatrica*, vol. 101, no. 10, pp. 1032–1037, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1651-2227.2012.02787.x

[3] T. Elfaramawy, C. L. Fall, S. Arab, M. Morissette, F. Lellouche, and B. Gosselin, "A wireless respiratory monitoring system using a wearable patch sensor network," *IEEE Sensors Journal*, vol. 19, no. 2, pp. 650–657, 2019.

[4] C. S. Abella, S. Bonina, A. Cucuccio, S. DAngelo, G. Giustolisi, A. D. Grasso, A. Imbruglia, G. S. Mauro, G. A. M. Nastasi, G. Palumbo, S. Pennisi, G. Sorbello, and A. Scuderi, "Autonomous energy-efficient wireless sensor network platform for home/office automation," *IEEE Sensors Journal*, vol. 19, no. 9, pp. 3501–3512, 2019.

[5] J. K. Brown, D. Abdallah, J. Boley, N. Collins, K. Craig, G. Glennon, K. Huang, C. J. Lukas, W. Moore, R. K. Sawyer, Y. Shakhsheer, F. B. Yahya, A. Wang, N. E. Roberts, D. D. Wentzloff, and B. H. Calhoun, "27.1 a 65nm energy-harvesting ulp soc with 256kb cortex-m0 enabling an 89.1ţw continuous machine health monitoring wireless self-powered system," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 420–422.

[6] N. Wuthibenjaphonchai, M. Haruta, T. Noda, K. Sasagawa, T. Tokuda, M. Sawan, S. Carrara, and J. Ohta, "Battery-free. sticker-like, device for health monitoring, operated by optical power transfer," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2018, pp. 1–4.

[7] A. M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Energy-efficient long-term continuous personal health monitoring," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 2, pp. 85–98, 2015.

[8] T. Elfaramawy, C. Latyr Fall, M. Morissette, F. Lellouche, and B. Gosselin, "Wireless respiratory monitoring and coughing detection using a wearable patch sensor network," in *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2017, pp. 197–200.

[9] A. Sekretaryova, "Chapter 4 - powering wearable bioelectronic devices," in *Wearable Bioelectronics*, ser. Materials Today, O. Parlak, A. Salleo, and A. Turner, Eds. Elsevier, 2020, pp. 89 – 132. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780081024072000059

[10] A. Sabban, "Small new wearable antennas for iot, medical and sport applications," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, 2019, pp. 1–5.

[11] J. Dudink is a specialist in neonatology and neoroscience. From 2004 to 2016 he has worked as a neonatologist and researcher of neonatal neuroimaging at Erasmus Medical Center. He has further worked as a pediatrician in Leiden and neonatologist in the Sophia children's hospital of Rotterdam. J. Dudink is presently working at the Department of Neonatology at University Medical Center Utrecht. [Online]. Available: https://www.umcutrecht.nl/en/research/researchers/dudink-jeroen-j

[12] W. H. Organization. (2018) Health-care waste. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/health-care-waste

[13] M. Ionascu and M. Marcu, “Energy profiling for different bluetooth low energy designs,” in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2, 2017, pp. 1032–1036.

[14] *Benchmarking MCU power consumption for ultra-low-power applications*.

[15] *MSP430FR599x, MSP430FR596x Mixed-Signal Microcontrollers datasheet*, Texas Instruments, 2018, rev. C. [Online]. Available: https://www.ti.com/lit/gpn/msp430fr5994

[16] *MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User’s Guide*, Texas Instruments, 2020, rev. P. [Online]. Available: https://www.ti.com/lit/pdf/slau367

[17] Radiation. (30-04-2020). [Online]. Available: https://www.who.int/health-topics/radiation#tab=tab_3

[18] E. F. Lee, “Investigation of using radio frequency identification (rfid) system for gear tooth crack detection,” 2014.

[19] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of things (iot) communication protocols: Review,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690.

[20] J. Lee, Y. Su, and C. Shen, “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, 2007, pp. 46–51.

[21] J. Lee, M. Dong, and Y. Sun, “A preliminary study of low power wireless technologies: Zigbee and bluetooth low energy,” in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 135–139.

[22] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, “Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario,” in *2013 IEEE International Wireless Symposium (IWS)*, 2013, pp. 1–4.

[23] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, “How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4,” in *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2012, pp. 232–237.

[24] C. Coleman. A practical guide to ble throughput. (30-04-2020). [Online]. Available: https://interrupt.memfault.com/blog/ble-throughput-primer

[25] ZigBee Alliance. Zigbee - zigbee alliance. (02-06-2020). [Online]. Available: https://zigbeealliance.org/solution/zigbee/

[26] Intel Corporation. Different wi-fi protocols and data rates. (02-06-2020). [Online]. Available: https://www.intel.com/content/www/us/en/support/articles/000005725/network-and-i-o/wireless-networking.html

[27] SiLabs. Bluetooth le fundamentals. (12-05-2020). [Online]. Available: https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

[28] Bluetooth SIG Inc. Exploring bluetooth 5, how fast can it be? (27-05-2020). [Online]. Available: https://www.bluetooth.com/blog/exploring-bluetooth-5-how-fast-can-it-be/

[29] J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica, “Performance Evaluation of Bluetooth Low Energy: A Systematic Review,” *Sensors (Basel)*, vol. 17, no. 12, Dec 2017.

[30] N. Iancu-Puiu and M. Marcu, “Highly accurate power profiling of bluetooth low energy communication,” in *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2018, pp. 417–424.

[31] Nordic online power profiler. (06-06-2020). [Online]. Available: https://devzone.nordicsemi.com/nordic/power

[32] *RSL10 - Bluetoothő 5 Radio System-on-Chip (SoC)*, ON Semiconductor, 2020, rev. 4. [Online]. Available: https://www.onsemi.com/pub/Collateral/RSL10-D.PDF

[33] *nRF52810 Product Specification*, Nordic Semiconductor, 2019, v1.3. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52810_PS_v1.3.pdf

[34] Texas Instruments. Code composer studio. (02-06-2020). [Online]. Available: https://www.ti.com/tool/CCSTUDIO

[35] Nordic Semiconductor. Installing the nrf connect sdk through nrf connect for desktop. (04-06-2020). [Online]. Available: https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/gs_assistant.html

[36] *nRF Connect for Mobile*, Nordic Semiconductor. [Online]. Available: https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-mobile

[37] L. S, R. V, O. T, and et al, “Individualized estimation of human core body temperature using noninvasive measurements,” *J Appl Physiol (1985)*, vol. 124, no. 6, pp. 1387–1402, 2018.

[38] Q.-S. Z. B.-Y. H. Guan-Zheng Liu, Yan-Wei Guo and L. Wang., “Estimation of respiration rate from three-dimensional acceleration data based on body sensor network,” *Telemedicine and e-Health*, vol. 17, no. 9, pp. 705–711, 2011.

[39] T. Reinvuo, M. Hannula, H. Sorvoja, E. Alasaarela, and R. Myllyla, “Measurement of respiratory rate with high-resolution accelerometer and emfit pressure sensor,” in *Proceedings of the 2006 IEEE Sensors Applications Symposium, 2006.*, 2006, pp. 192–195.

[40] N. H. Genootschap. (2018) Reiniging en desinfectie of sterilisatie van instrumentarium. [Online]. Available: https://www.nhg.org/themas/publicaties/reiniging-en-desinfectie-sterilisatie-van-instrumentarium

[41] *Chemical Resistance Guide for Silicone Hose*, Parker, -. [Online]. Available: https://www.parker.com/Literature/Fluid%20Transfer%20Hose%20-%20Europe/chemical%20table%20silicone.pdf