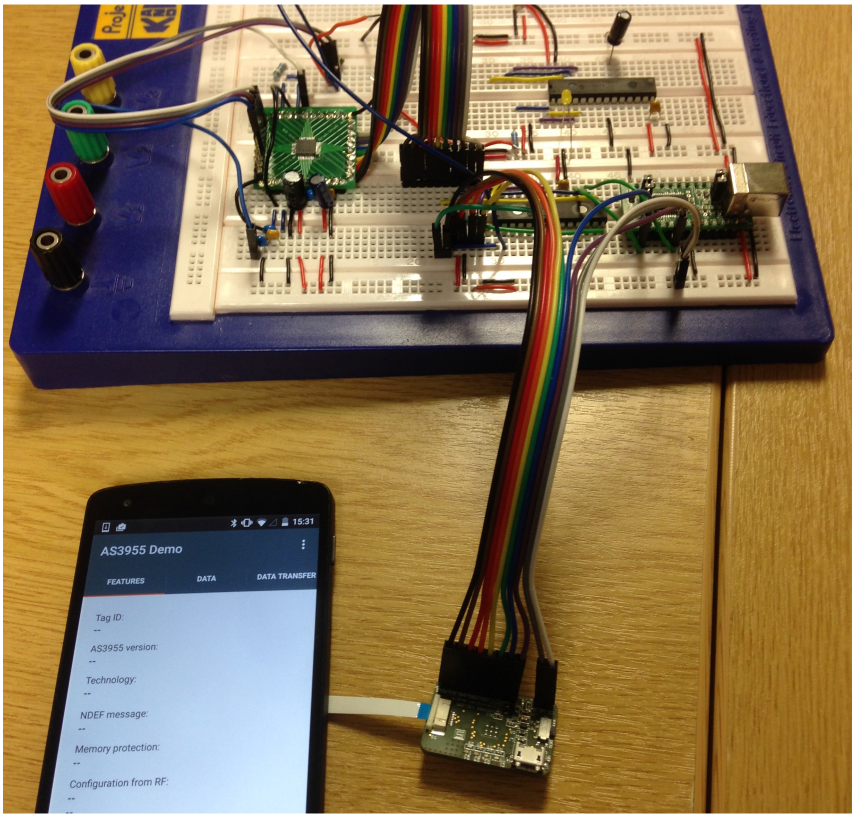
Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2016



Project Title:	**An Autonomous Potentiostat for Point-of-care Multimetabolite Measurement in Diabetes**
Student:	**Pascal Loose**
CID:	**00694642**
Course:	**EEE4**
Project Supervisor:	**Dr Pantelis Georgiou**
Second Marker:	**Dr Konstantin Nikolic**

Abstract

The aim of this project is to build a device that is capable of measuring the concentration of glucose in a solution and sending the measurements via NFC to a smartphone, while also being powered by the smartphone. The device consists of a microcontroller that interfaces with a potentiostat and a RFID tag and the report details the implementation. Using a breadboard as prototype, the functionality and testing is demonstrated. The report concludes with highlighting required further work and future development.

Acknowledgements

Overall this project was a very rewarding experience. Even though at times it became very difficult to sustain motivation, I am very glad I have done this project and I feel that I am going to graduate as a proper electronic engineer. I would like to express my gratitude to:

- My supervisor Dr Pantelis Georgiou, for his relentless patience and advice
- My co-supervisor Dr Sara Ghoreishizadeh, for her constant help and encouragement
- Mohamed Fayez El Sharkawy, who helped me so much and shared his microcontroller programming experience with me
- My family and friends

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Diabetes is one of the most prevalent chronic diseases in the UK with nearly 6% of the British population being diagnosed with diabetes [11]. Diabetics are in danger of having too much glucose within their blood and are therefore required to continuously measure their glucose levels. Several glucose meters are available that allow the user to measure their glucose levels. These devices can be either invasive, where a blood sample of the patient is required, or noninvasive, where the device can measure the levels without puncturing the skin. In most cases the glucose meters require a battery for powering, requiring patients to change it whenever it's depleted.

Simultaneously, smartphones become more and more omnipresent, with over 3/4 of the UK's population owning a smartphone [12]. Modern smartphones incorporate various wireless communication tools, such as Bluetooth, Wi-Fi and Near Field Communication (NFC). A relatively new technology, NFC allows to wirelessly send data as well as power small devices wirelessly. Major technology companies have begun to incorporate NFC in the design of new wireless glucose meters. Google for example have patented contact lens sensors that measure the glucose levels via a contact lens. The lens is powered and communicates via

NFC to send the data to a smartphone [13].

Glucose is one of various metabolites, others include lactate or glutamate, which can be indicative of a patient's well-being depending on the concentration of the respective metabolite. By design glucose meters are limited to only measure glucose, they don't measure other metabolites. A common device to use to determine the glucose levels is a potentiostat, which measures a very small current that is generated through chemical reactions. A potentiostat allows users to measure other metabolites in the same fashion. This potential combined with the features of NFC forms the baseline of this project.

1.2 Project Objectives & Specifications

The main aim of this project is to design a baseline prototype of a device that is able to detect very small currents and send the data via NFC to a smartphone. The device needs to consume as little power as possible and is designed to be battery free as it is powered via the NFC of the smartphone. The fourth specification is that the device requires to be as inconspicuous as possible and therefore it should be designed to be very small, not bigger than 3x3 cm.

In order to achieve these specifications there are several objectives that need to be met. The first objective was to research, compare and select appropriate components that are required for this device. The device should be built using only off-the-shelf hardware; the components necessary for this project are based on a potentiostat, a NFC tag and a microcontroller. The second objective is to integrate these components and design the required hardware. Where possible, pre-existing libraries for the devices are used and / or modified in order to streamline the development of the prototype, compared to building custom libraries.

To validate this concept, the device should be able to detect very small currents in the range of nA with a single sensor. At this point it is not necessary to test if it can measure multiple metabolites, as the potentiostat system for measuring for example glucose is identical for measuring other metabolites. Therefore the system is validated if it can detect the current that would be generated when measuring the concentration of glucose. Another objective for this project is for it to be able to send the data from the potentiostat via the NFC tag to a smartphone. The last objective is for the device to be able to be powered via NFC. The capabilities of this platform are tested and recommendations are suggested for any future work.

1.3 Report Structure

In this report, the design choices, their implementation and the results are described. Chapter 2 explains in detail the technology that is used and outlines related work that has been undertaken. The related work forms a comparison for this project to be evaluated against. Chapter 3 then explains the structure of the system and how it is implemented. In Chapter 4 the platform is tested and the results are explained. Chapter 5 then outlines any further work than can be done to improve the platform. In Chapter 6 the project is evaluated against the objectives and specifications and Chapter 6 draws the project to its conclusion.

Chapter 2

Background

2.1 Potentiostat

When assessing the state of a patient, most electrochemical devices measure the concentration of their respective metabolites, which are a family of molecules relevant to the metabolism. In the case of diabetics, the concentration of glucose in the blood is the factor, which is indicative of the status of their condition. In order to measure the concentration, sensors use the enzyme glucose oxidase [9]. By applying a voltage to the cell, the sensors break down the glucose via the enzymes [9]. This chemical reaction, oxidation, releases electrons which creates a current that is measured by a potentiostat. The basic setup of the sensors with a potentiostat is shown in Figure 2.1.

The electrochemical cell consists of three electrodes, the working electrode (WE), the reference electrode (RE) and the counter electrode (CE). In a solution of glucose a voltage is applied at WE, which results in a transfer of charge to and from the solution as oxidation takes place. The RE is used to determine and stabilise the voltage of WE, while not allowing any current to flow through it. The third electrode CE is put in place to balance the charge added or removed by WE. This counterbalancing is the measured current, shown as the current source in Figure 2.1, which is proportional to the concentration of glucose [9].

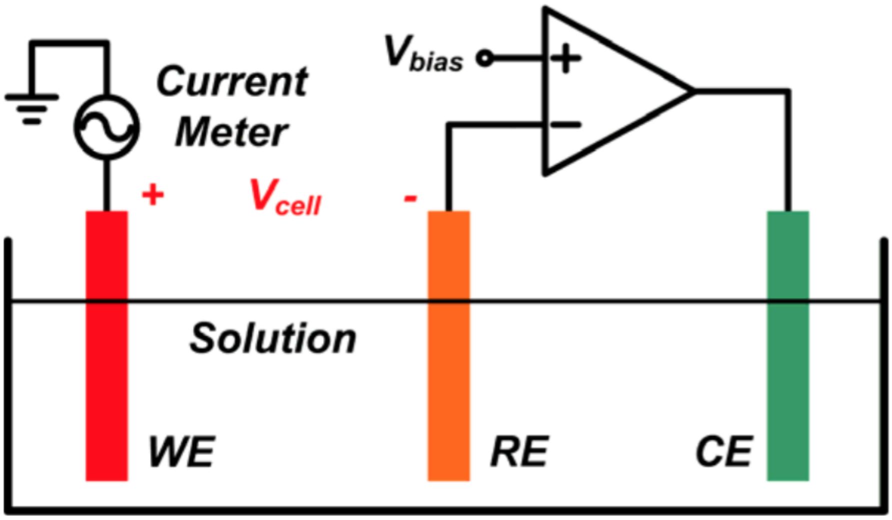


Figure 2.1: Concept of a electrochemical cell with a potentiostat [1]

Each metabolite has its own oxidation voltage. By applying the appropriate oxidation voltage between RE and WE, shown in Figure 2.1 as V_{cell}, and by using the corresponding sensors, the respective metabolite concentration can be measured as described above for glucose. These voltages are summarised in table 2.1. Therefore it's possible to target specific molecules in a solution containing metabolites by applying the required oxidation voltage.

Target molecule	Applied voltage (mV)
Glucose	+ 520
Lactate	+ 650
Cholesterol	+ 700
Glutamate	+ 600

Table 2.1: Oxidation voltages for respective metabolites [9]

In order to read out the current, the device requires a transimpedance amplifier (TIA), which is a current to voltage converter, an analogue to digital converter (ADC) to process the signal so that it can be used further in the device and a digital to analogue converter (DAC) to set the voltage at WE. The combination of a TIA and ADC forms a readout circuit.

2.2 Comparison of Data Exchange Standards

The aim for this project is for the user to only require a smartphone in addition to the device. Modern smartphones incorporate various communication formats, whereas NFC is the only communication protocol allowing to power other devices wirelessly. Other communication methods include Bluetooth and Wi-Fi, which are compared below.

Bluetooth was established long before NFC and is widely used to exchange data and connect smartphones with external devices such as speakers. Nowadays manufacturers either use the protocol Bluetooth 3.0 or the protocol Bluetooth 4.0, Bluetooth Low Energy (BLE). BLE was introduced to enable low power devices with Bluetooth and BLE has a data rate of 1 Mbps and range of under 50 m[14]. Many commercially available glucose meters use BLE to connect to smartphones [15] [16]. BLE is only a communication protocol and in order to use it the device would need a battery-powered BLE receiver.

Another common communication standard is Wi-Fi. Wi-Fi has the longest range up to 100 m and largest data rate, up to 600 Mbps [14]. Devices that use Wi-Fi either need to be connected to an access point or to other Wi-Fi enabled devices, in this case the smartphone. It does however use too much power for any small low-power devices, consuming current at a rate of 92 mA in receiving mode [14]. Like BLE, a device that uses Wi-Fi would need a battery in order to power it.

NFC has a data rate of up to 424 kbit/s [17] and is limited to a read range of 4 cm [2]. It is however the only communication protocol that also allows to power the device wirelessly. Additionally NFC is the most secure method of connection as it requires such close proximity between devices.

2.3 NFC

2.3.1 RFID

NFC has its roots in radio frequency identification (RFID). RFID is used for short-distance communication and when using RFID as a communication protocol it requires the use of a tag and of a reader, sometimes called initiator [18] [2]. By utilising radio waves, RFID allows the reader to scan a tag, which usually returns an unique identifier number (UID) and associated information. There are two types of tags, which also form the communication mode: active and passive [18] [2]. Passive tags don't have any on-board power and are powered via electromagnetic induction at low frequencies by the signal emitted by the reader [18]. Passive tags can't communicate unless read by a reader and usually consist of an antenna and some circuitry, containing the ID. Active tags include on-board power, typically a battery, which extends the read range, and active tags can initiate communication [18]. Depending on the frequency used for RFID communication, different frequencies are used, which also influences the read range. RFID is usually used in smart cards and monitoring inventory.

2.3.2 NFC Overview

NFC is an extension of RFID, operating in the 13.56 MHz band. While it's possible to perform the same operation as RFID, such as reading RFID tags and receiving the UID with NFC, NFC enables more complex exchanges [2]. Found in many modern smartphones, it allows short range wireless communication, it's not designed for long range communication. NFC is regulated and complies to standards regarding to data formats and transfer rates, making it interoperable and easily integrated within systems [10]. The same channel used for NFC is half duplex, meaning it can be used for both transmit and receive [10]. Data transfer rate varies from 106, 212, 414 kbps [10]. Similar to RFID having active and passive tags, NFC has active and passive communication: Active communication has two devices sending RF signals, passive communication has only one device emitting RF signals [10]. In passive communication the second device uses load modulation to transfer the data back to the interrogating device [10].

NFC devices can operate in three communication modes [19]:

- Read / Write: A reader can read data from a target and write to it
- NFC card emulation: Similar to RFID, a reader can replace a contactless smartcard
- Peer to peer: Data is exchanged in both directions

In the read / write mode the tag acts as a dual memory port: one port is connected to the reader and the second port is connected to the embedded system [19]. As the aim of this project is to read and write data to a NFC tag, the read / write mode is the preferred mode.

2.3.3 NFC Tags

NFC tags are able to store data within the integrated circuit, often using electrically erasable programmable read-only memory (EEPROM). The benefit of using EEPROM is that the data can be written to the tag even though the system is not receiving power. The system would receive the data later via one of the connections. Similarly, the microcontroller of the system can write to the EEPROM even when NFC is not present.

One limitation of NFC is that it's not possible to have one reader communicate with multiple targets simultaneously. It's only possible to communicate on a one-to-one basis. Theoretically, if two targets were to enter the RF field of a reader, the reader wouldn't be able to distinguish between them and their responses [2]. This is called collision [10]. In order to prevent multiple NFC targets interfering with each other, NFC tags can include anti-collision support. Once the tag with anti-collision has established a communication with the reader, other tags are not able to interfere [19].

There are four different types of tags, with different capabilities. They are summarised in the table below:

Type	Capability	Memory	Speed (Kb/s)	Anit-collision
Type 1	Read-only or read/write	96 bytes to 2 Kbytes	106	No protection
Type 2	Read-only or read/write	48 bytes to 2 Kbytes	106	Anti-collision support
Type 3	Read-only or read/write	2 Kbytes	212 or 424	Anti-collision support
Type 4	By factory either read-only or read/write	32 Kbytes	106, 212 or 424	Anti-collision support

Table 2.2: Different types of tags [10]

Tag 1, 2 and 4 all use the same standard and are the most commonly used types of tags. Tag Type 1 and Type 2 are identical apart from the anti-collision support. Although it is unlikely that the system is going to have tags interfering with one another, it forms good practice to include anti-collision support. Therefore when researching for appropriate NFC tags for this project, type 2 tags are the preffered types.

NFC tags are usually connected to the main processor of a device either via serial peripheral interface (SPI), universal asynchronous receive-transmit (UART), integrated circuit communication (I2C) or universal serial bus (USB).

2.3.4 NDEF

NFC data exchange format (NDEF) is the most common way to send messages between NFC devices or between a NFC reader and tag [10]. NDEF is a binary message that includes one or more records, where each record contains a header and a payload, the message's content, as shown in Figure 2.2 [10]. The record header contains metadata about the record, such as UID and length of the payload [10].

Each record can be of various types and common NDEF records types include:

- Simple text records: Generally text strings, which can include metadata and instructions for the target device
- URIs: Uniform resource identifier records contain network addresses, which are sent with an instruction
- Smart posters: Data that might be associated with a poster, such as websites
- Signatures: Contain trustworthy information about the origins of data

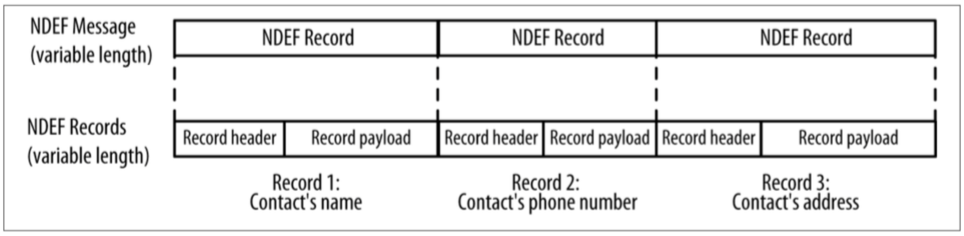


Figure 2.2: NDEF structure [2]

By using the text format the smartphone is able to receive any data, such as the glucose concentration, coming from the device in a string format. Vice versa, any instructions for the device would be sent encoded in the text string.

The record header contains all the information that allow the reader to interpret the message and its structure is shown in Figure 2.3.

The header is divided into bytes and its length depends on its content. The initial byte contain the message flags of the record and has following layout [2]:

- Message begins (MB), used for first record in a NDEF message
- Message ends (ME), used for last record in a NDEF
- Chunk flag (CF), used for then the record is chunked, as described below
- Short record (SR), used if the payload length is either 1 byte or 4 bytes
- ID length (IL), used if the length of the ID is used in the header
- Type Name Format (TNF), defining the type of the payload

If the message is very short, the MB, ME and SR bits are set. If no ID is required, the IL is set low. The size of the NDEF message depends on the memory size of

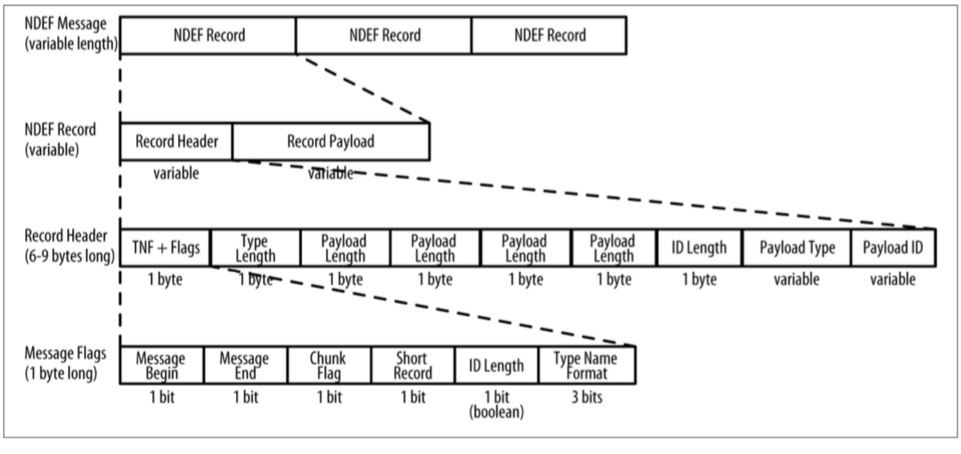


Figure 2.3: NDEF message structure [2]

the tag. If the message needs to broken down it can be chunked and the chunked bit in the header is used. During the prototyping process the device will only send very short messages and won't require an ID, so it won't be necessary to chunk the NDEF or set the IL bit.

The second byte of the record header contains information about the length of the payload type. The payload type that is being used for this device is going to be of type text 'T', which only requires one byte. Therefore the second byte of the header is set to 0x01. The payload length can have a length of either 1 or 4 bytes. If the payload length was set to 1 byte it would mean that the payload would have a maximum of 255 bytes of content, which is enough space to send the short messages. Since no ID is required and the IL bit is set low, the record doesn't include the ID length or the last optional payload ID bytes [2].

There are eight different TNFs, including media-types and URIs. The only applicable for this project is the format TNF 01, which is defined as the well-known format. When setting the payload as type text 'T' the payload needs to start with the bytes 0x02, 0x65, 0x6e. This configures the payload text as the English language code and the common standard ASCII to Hex conversion can be applied.

2.4 Selection of Components

The project required two main components, a potentiostat and a NFC tag. Various components from different manufacturers were considered and compared.

2.4.1 Selection of Potentiostat

There aren't many potentiostats available to purchase as a chip. In the end 3 potentiostat chips were considered and are compared in Table 2.3. When looking for suitable potentiostats following parameters were considered:

- Current range: The potentiostat needs to be able to detect very small currents, therefore the smaller the current it can detect, the more desirable it is.
- Supply voltage: The supply voltage is ultimately coming from the NFC tag. The system needs to consume as little as power as possible, therefore smaller supply voltages were preferred.
- Size: Since the final design is ideally as small as possible, the potentiostat is also required to be as small as possible.
- ADC: The current of the potentiostat is going to be converted via an ADC. The larger the resolution of the ADC, the more precise the potentiostat is.
- Communication interface: SPI and I^2C are the most common interfaces.

In the end the MAX1329 chip was chosen as the potentiostat for this project. It has the lowest current range, starting from 1 nA and the lowest supply voltage at 1.8 V. The LMP91000 chip includes a programmable TIA gain, allowing to change the gain of the transimpedance amplifier. The MAX1329 chip doesn't have one, but its TIA can be changed with a configurable resistor.

2.4.2 Selection of NFC Tags

To allow quick prototyping, NFC development boards were considered instead of only using NFC tags. NFC development boards often include pre-configured

Name	Manufacturer	Features
MAX1329 [6]	Maxim	• 1.8 V to 3.6 V supply voltage • Current range from 1 nA • SPI interface • 12-bit ADC • 6 mm x 6 mm size
LMP91000 [20]	Texas Instrument	• 2.7 V to 5.25 V supply voltage • Current range of 5 A to 750 A • Programmable TIA gain • I^2C interface • 4 mm x 4 mm size
ADuCM350 [21]	Analog Devices	• 2.5 V to 3.6 V supply voltage • Current range up to 33.5 μA • 16-bit ADC • SPI, I^2C and UART interface • 8 mm x 8 mm size

Table 2.3: List of potentiostats

microcontrollers that can be modified according to the objectives. The main criteria for applicable NFC development boards were that the NFC tags were passive and that they could harvest energy from the reader. An overview of five available development boards is shown in Table 2.4.

When researching NFC development boards following properties were compared:

- Tag type: As described in Section 2.3.3, the tag is required to be type 2 tag. Type 1 tag was acceptable too.
- Memory: The size of the memory needs to be as large as possible as the microcontroller is going to write to the tag and store data on it.
- Data rate: A faster data rate would result in a quicker exchange with the NFC reader and is therefore desirable.
- Communication: SPI or I^2C communication are the most common one and there wasn't a preference for one. Ultimately it should have been the same communication that is used for the potentiostat so that they can be easily integrated.
- Supplied power: The most critical property determines how much power the tag is able to deliver in terms of voltage and supplied current.
- Size: Since the final design is ideally as small as possible, the tag is also required to be as small as possible.

The development board that was chosen was AS3955. AS3953A has a data rate at 848 Kb/s, depending on which memory is used. Compared with the AS3955, which has a data rate of 106 Kb/s, the AS3953A is quicker, but this drawback can be circumvented by sustaining a longer communication time with the reader. Ultimately AS3955 provides the most power at 5 mA and 4.5 V. The board includes the microcontroller PIC24FJ128GB20, and uses both SPI

Name	Manufacturer	Microcontroller	Features
NTAG I^{2}C [22]	NXP	Not included	• Type 2 Tag • 2016 bytes EEPROM and 64 bytes of SRAM • I^{2}C interface • Data rate 106 Kb/s • Power supply of 5 mA and 2 V
AS3953A [23]	AMS	PIC27FJ-64GB002	• Type 4 Tag • 1 kbit EEPROM • SPI interface • Data rate from 106 Kb/s to 848 Kb/s • Power supply of 4 mA and 1.8 V • Size of 45 x 40 mm
AS3955 [8]	AMS	PIC24FJ-128GB20	• SPI and I^{2}C interface • Type 2 or 4 Tag • 4 kbit EEPROM • Data rate 106 Kb/s • Power supply of 5 mA and 4.5 V • Size of 32 x 45 mm
M24LR04E-R [24]	ST	STM8L-152C6T6	• I^{2}C interface • 4 kbit EEPROM • Data rate 26.48 Kb/s • Power supply of 6 mA and 1.7 V
RF430FRL-152HEVM [25]	TI	MSP430	• SPI and I^{2}C interface • 2 kbit of FRAM • Data rate 26.48 Kb/s • Power supply of 100 μA and 2 V

Table 2.4: Table caption

and I^2C interface protocols, which provides more options for programming the microcontroller [26]. The demonstration kit includes three different tags of different size, the biggest one having a dimension of 32 x 45 mm, which is slightly larger than the envisioned size of the device. The board itself has a dimension of 32 x 21 mm, which is just about the desired size of 3 x 3 cm.

2.5 Comparison of Mobile Operating System

The leading smartphone mobile operating system are Apple's iOS and Google's Android. The latest devices offered by both incorporate NFC, however Apple uses NFC solely for its Wallet product and therefore only uses the smart card emulator as described in the Section 2.3.2 [27]. Secondary mobile operating systems were considered too, but ultimately Android provides much more online support compared to others. Therefore, the selected operating system was Android for this project. [28].

2.6 Existing Work

This section explores relevant work that incorporate some of the objectives, and give a guideline what the system needs to ultimately achieve.

2.6.1 NFC-WISP

Zhao, Smith and Sample developed a sensing platform that uses NFC from a smartphone to power and communicate with the platform [3]. The NFC-WISP (Wireless Identification and Sensing Platform) is based on a PCB that has the RFID antenna printed within the board itself and uses an ultra low power microcontroller to handle the NFC protocol and sensing. A power harvesting unit on the board rectifies the incoming energy into DC voltage to power the system and stores any unused energy within a capacitor. A block diagram of the NFC-WISP, which gives an overview of the building blocks, is shown in Figure 2.4. This platform forms a baseline validation for powering a device solely with an NFC signal.

2.6.2 Wireless potentiostat for mobile chemical sensing

Kassal, Kerekovi and Steinbergs developed a mobile sensor that uses NFC and a potentiostat [4]; the principle of the device is displayed in Figure 2.5. The mobile sensing platform uses an antenna that is similar to the aforementioned NFC-WISP, printed within the PCB, but uses a 3 V lithium coin cell to power the platform. The potentiostat is able to measure a current in the range of 15 to 4394 nA [4], and so achieves a dynamic range that is close to the one that is desired here.

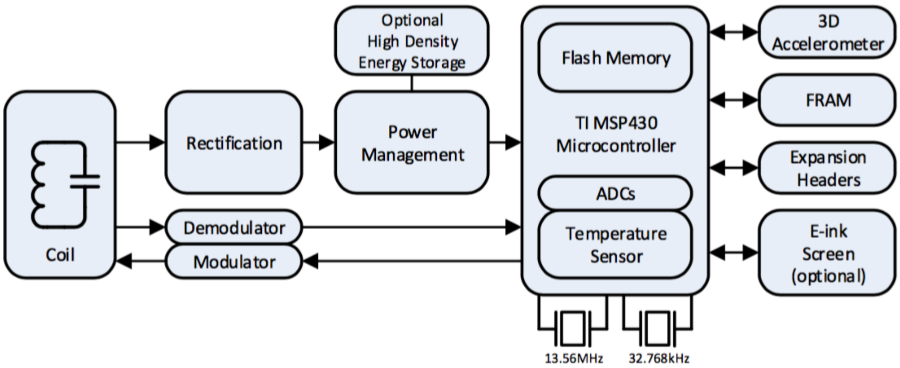


Figure 2.4: Block diagram of NFC-WISP [3]

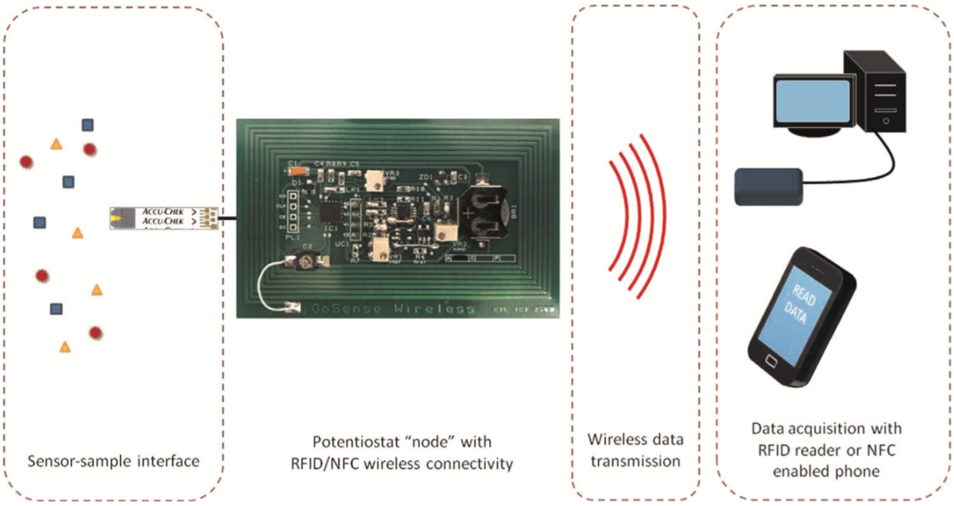


Figure 2.5: Principle of mobile potentiostat by [4]

2.6.3 Dual-channel wide input range interface circuit

Liu, Chen, Li and Ren built a amperometric potentiostat within a PCB that allows for detection of a current within a dynamic range of 120 dB, measuring currents in the range of 1nA to 1μA [29]. An off-the-shelf potentiostat wasn't used but rather a bespoke device based on current integrators and transimpedance amplifiers. Overall, the sensor consumes 3.3 V at 4.2 mA, making it quite expensive in terms of power consumption.

2.6.4 Continuous Glucose Monitoring System

Another system was developed by Cai, Cao, He and Wang, which consists of smartphone application, a monitor centre and a Bluetooth equipped sensor [30]. The sensors are planted underneath the skin and continuously measure the glucose levels, the data is then sent via Bluetooth to a smartphone. Via the 3G mobile network, the smartphone connects to the monitor center, which can act accordingly in response to the incoming data.

2.6.5 Implantable, NFC enabled Glucose Monitor

Anabtawi, Freeman and Ferzli recently published their work on an implantable glucose monitor. [5]. The system-on-chip (SoC) uses an amperometric glucose sensor interface, NFC and a power management unit for on board charging or supply regulation [5]. The system level architecture is shown in Figure 2.6. In the presence of an RF field, the system harvests energy from NFC to generate a digital and analogue supply, which charge the battery. Once the reader is removed the device is supplied by the battery. Overall the system consumes either 47 mW or 24 μW for charging enabled or charging disabled, respectively.

The objectives achieved in this work are very similar to the to the vision for this project. However its aim is to be implantable, whereas the device being developed in this project is intended to take the form of a patch worn on the skin. Furthermore, while SoCs are usually more resourceful in terms of power management and size, they are more expensive to develop, as they don't use off-the-shelf components.

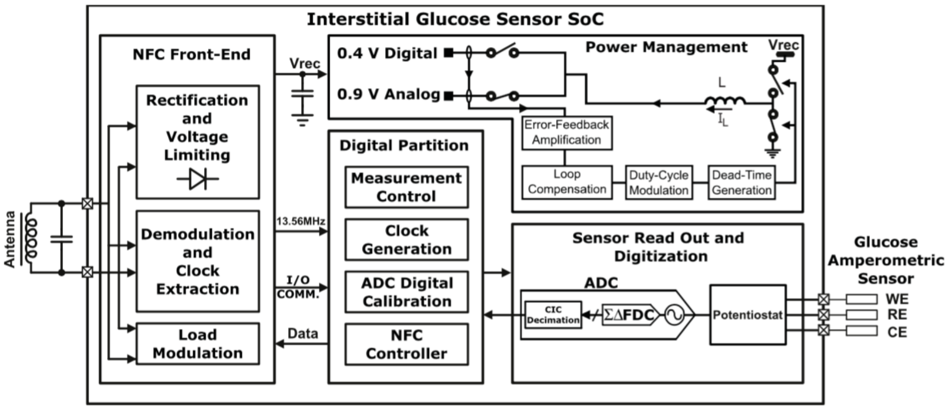


Figure 2.6: NFC enabled glucose monitor SoC by [5]

Chapter 3

Implementation

3.1 Design Overview

Overall the system is based on two components, the NFC tag and the potentiostat. The high level design of the device is shown in Figure 3.1. The design shows how the components of the system interact with each other. The NFC tag receives instructions from a smartphone, which it passes on to a microcontroller via SPI. The tag also harvests the energy from the smartphone to supply power to the microcontroller and the potentiostat. Based on the instructions received from the smartphone, the microcontroller configures the potentiostat, which then reads the concentration of a metabolite via its sensors. The results are sent back to the microcontroller via SPI, which are passed on to the NFC tag, where they can be read from a smartphone. As both peripherals communicate using SPI, it's easy to integrate both interfaces onto the microcontroller.

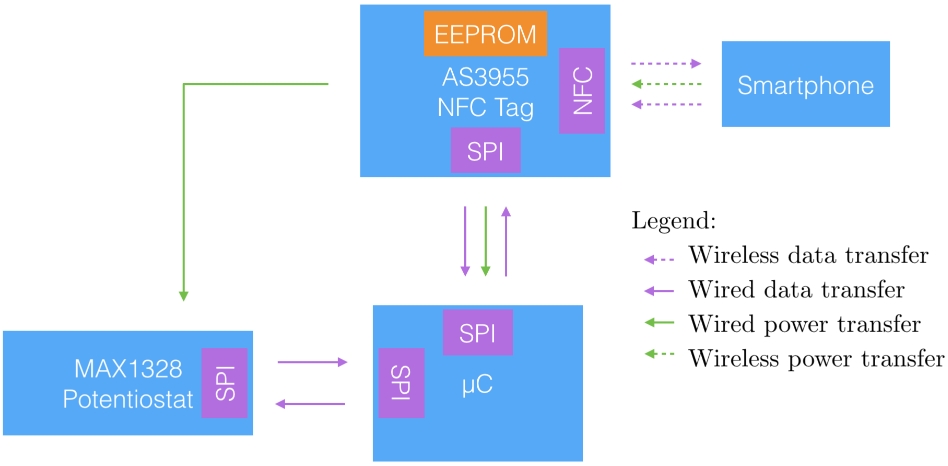


Figure 3.1: Block diagram of system design

During the implementation process of the prototype, two microcontrollers were used to configure both front-end components, as explained below. The main specifications of this prototype are to:

- Measure small current via a potentiostat
- Write the measurement to the EEPROM of the NFC tag
- Harvest energy via NFC from the smartphone

3.2 Microcontroller Overview

Some of the benefits of using the PIC24 microcontroller are its low-power features. These include various sleep and idle modes that turn off peripherals, minimising the power consumption when the microcontroller is not active, as well as very fast wake up times [31]. Furthermore it includes alternate clock modes, which, when used, reduce the clock speed, decreasing the power consumption further. While these features are not made use of during the prototyping stage, they are worth considering when testing the power consumption of the project in Section 4.3.

3.3 Configuration of the Potentiostat

3.3.1 Overview

As described in Section 2.4.1, the MAX1329 was selected as the potentiostat for the system. The first step in building this prototype was to configure and set up the chip with a microcontroller. The microcontroller being used is responsible in setting up the SPI configuration, in initialising the chip and in receiving any data. The MAX1329 is defined as a configurable data acquisition system with multiple configuration options, including the potentiostat that is required for the device being developed [6]. The potentiostat configuration from the MAX1329 chip is shown below in Figure 3.2.

The potentiostat can be configured in two different modes, either in the single channel mode or the dual channel configuration. In the single channel configuration the set-up is as it was described in Section 2.1, whereas in the dual channel configuration the reference electrode is used as a second working electrode [6]. For the purpose of this project the single channel configuration is used. Within the single channel configuration the second DAC, DACB, is not used and only the first DAC, DACA, and Op Amp 1, OA1, are used. When ignoring DACB, the single channel configuration can translated into the diagram as shown below in Figure 3.3.

In this configuration, the current I_A is then the sensor current that is being measured and is converted via R_A into a voltage [6]. The REFADC voltage, set at 1.25 V, is used as the reference voltage for the RE of the potentiostat.

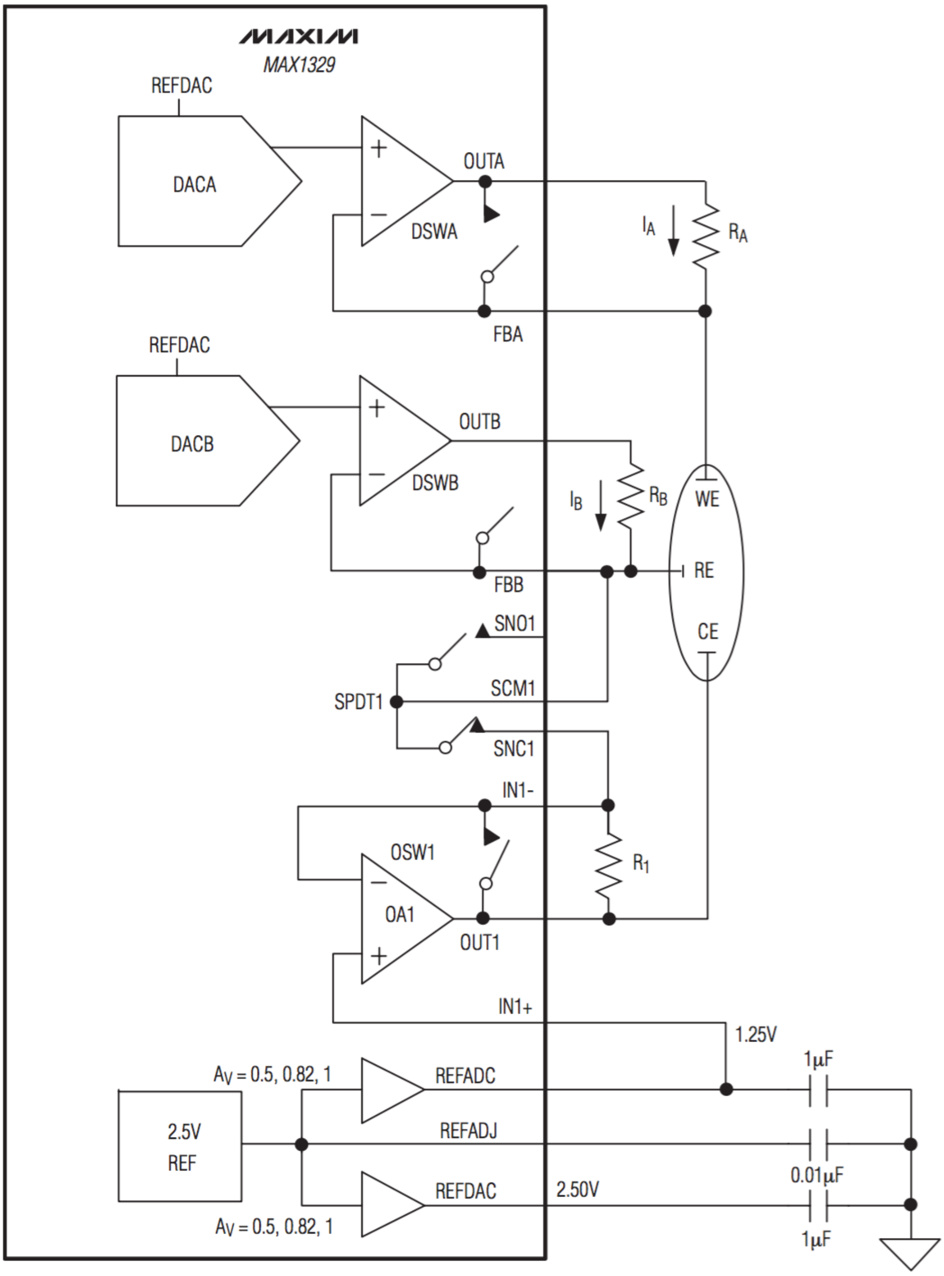


Figure 3.2: Potentiostat configuration of Maxim 1329 [6]

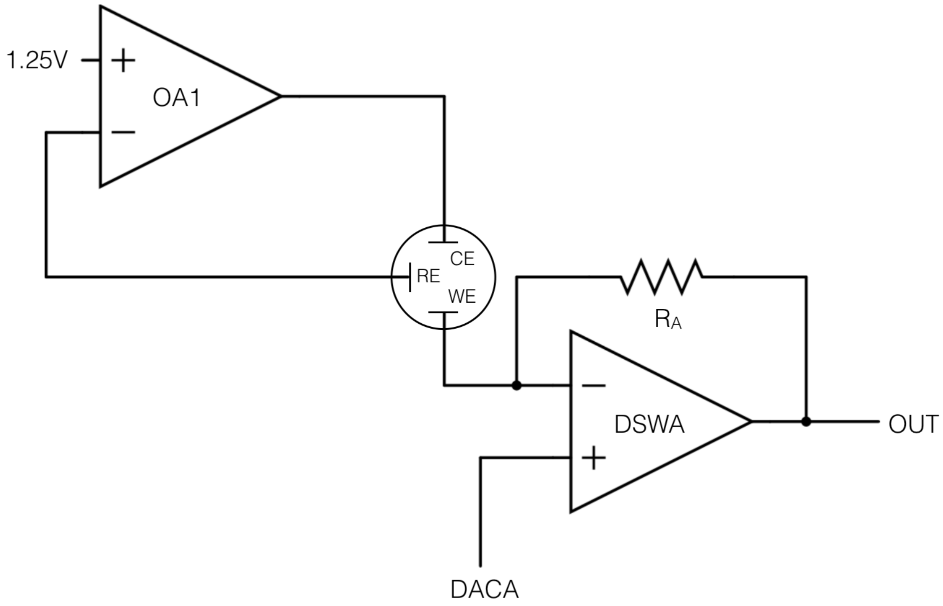


Figure 3.3: Simplified potentiostat configuration

R_A is an external resistor that can be changed accordingly. The corresponding oxidation voltage is set via at DACA, and any current that is generated by the oxidation with the sensors results in a voltage drop across R_A, which can be measured by reading the value at the output pin OUTA.

3.3.2 DAC Set-up

To set the DACA, the MAX1329 uses direct-mode commands to set the DACs directly [6]. The MAX1329 chip uses SPI for communication and initially it was tested with a PIC24 microcontroller, as the NFC development board uses a PIC24 to interface to the NFC tag. However, the PIC24 has reprogrammable SPI pins. The PIC18F25J11 microcontroller that was used for building the prototype has SPI configured pins, which proved to be more effective when using SPI compared to the PIC24 [32].

In this set-up, the PIC18 acts as the SPI Master and the MAX1329 as the SPI Slave. An example timing diagram of the direct-mode command is shown in Figure 3.4. The Chip Select (CS) communication line is set by the PIC18 and needs to be set low in order to allow any data exchange. One of the earlier shortfalls was when the switching of the CS line was attempted to be set manually. By using one of the pre-configured SPI commands that manages the switching of CS line, this obstacle was overcome. The SCLK clock is provided by the PIC18 and the microcontroller was configured to operate in SPI mode 0. In this mode, any data is captured on the rising edge of the clock and written on the falling edge of the clock.

In order to write as a direct-mode command to the DACs, it is required to send a write instruction 010b first. The fourth bit, AB, is to select the corresponding DAC, 0 to select DACA and 1 to select DACB. The twelve preceding bits, D_{11} to D_0, are the resolution bits of the DAC. The digital value of the DACA is described by the following equation:

$$D = V_{DACA} * \frac{2^{12}}{V_{REFDAC}} \tag{3.1}$$

where V_{DACA} is the required output voltage of the DACA and V_{REFDAC} is by design 2.5 V, as explained in Section 3.3.3 [6].

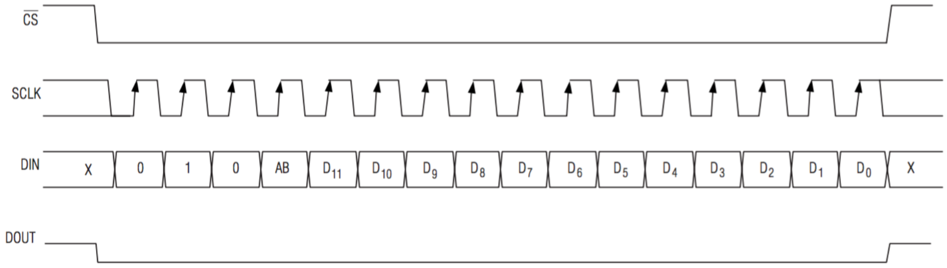


Figure 3.4: Timing diagram of the DACA diagram [6]

3.3.3 Control Registers Set-up

To obtain the configuration as shown in Figure 3.2, there are three control registers that need to be set: ADC Control, DAC Control and Switch Control. The control registers are set via SPI in the same configuration as in Section 3.3.2 and follow the same initial five-bit format to indicate the mode, either read or write, and address of the corresponding register. An overview of the control registers is shown in Figure 3.5.

The MAX1329 uses an internal unbuffered 2.5 V reference voltage, noted as REFADJ. The voltage of REFDAC V_{DACA} is used to set DACA, whereas the voltage of REFADC is used as the bias voltage for WE and RE for the electrochemical sensors. Depending on the gain, the voltages of REFADC and REFDAC can be set through the respective ADC and DAC control register. The reference voltage for the DAC requires a gain of 1 to set it to 2.5 V. The voltage of REFADC requires to be 1.25 V, therefore the gain in the ADC Control register needs to be set to 0.5.

The AUTO bits within the ADC Control register define if the ADC is used automatically at a specific time interval. Currently it is not used and therefore set to 0. The APD bits set the mode of the ADC, and since the ADC is not used at this point are set to 0. To set the gain of the internal buffer reference voltage to 0.5 the bits AREF are set to 01, whereas the last bit REFE enables the internal buffer voltage REFADJ when set to 1.

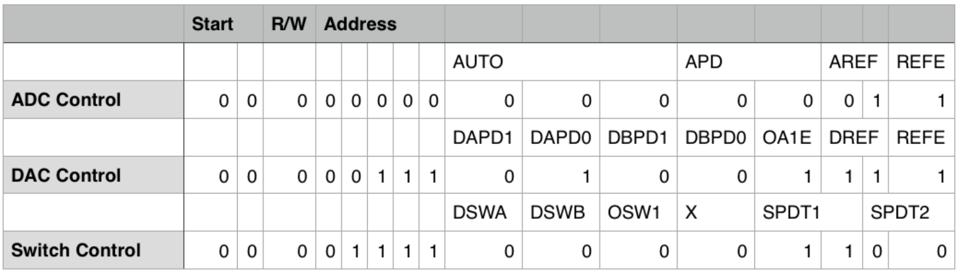
The DAPD bits of the DAC Control register either power the DACA up or down. Since the DACA is used, it is powered by setting the bits to 01b. The following two bits for DBPD control the power state of DACB, and as DACB is not used, they are set to 0, as explained in Section 3.3.4. The OA1E enables the op amp 1 as required for the configuration and the DREF bit sets the gain of the internal reference buffer of the DAC. To obtain a gain of 1 they are set to 11. REFE, identical to the set-up in ADC Control, is set to 1 to enable REFADJ.

The Switch Control register is addressed at 01111b and defines whether the single or the dual channel configuration is used. It sets the state of DACA via the DSWA switch, which needs to be set to 0. SPDT1 modifies a single-pole, double-throw switch which needs to be closed, therefore it set to 11.

The control registers are only required to be set once need to be only set once to configure the potentiostat. Therefore they are configured when the `main()` loop is invoked.

3.3.4 ADC Set-up

The PIC18 includes a direct ADC pin, which was used to read out any data that was coming in from the MAX1329 chip. The PIC18 ADC was used instead of the ADC of the MAX1329 chip to allow rapid prototyping; the ADC of the MAX1329 required further configurations in order to be used. The ADC was initialised with the `setup_adc` command and any data was written to a `adc_read` variable. As described in the Section 4.2, the output was fluctuating heavily, so it was sampled at a 200 ms interval and then averaged for every 10 readings. The frequency of 200 ms was chosen as a trade-off between quick readings and limiting the power consumption by not sampling too quickly.



	Start		**R/W**	**Address**												
									AUTO			APD		AREF		REFE
ADC Control	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
									DAPD1	DAPD0	DBPD1	DBPD0	OA1E	DREF		REFE
DAC Control	0	0	0	0	0	1	1	1	0	1	0	0	1	1	1	1
									DSWA	DSWB	OSW1	X	SPDT1		SPDT2	
Switch Control	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0

Figure 3.5: Control register overview

The readings of the ADC were sent to a PC by using a UM232R UART Serial module [33]. The TX and RX pins of the PIC18 were connected to the RX and TX pins of the UM232R module respectively. The BAUD rate was set to 9600 Bps, which is the most commonly used data rate.

3.4 Overview of the NFC Tag

3.4.1 NFC Demonstration Board

Before fully integrating the AS3955 tag within a single device, the tag was tested and used in system development with its corresponding demonstration board. The demonstration kit supplied by ams contains the demonstration board, three different sized tags and a ribbon cable to connect one of the tags with the demonstration board. The PIC24FJ128GB204 microcontroller is used in the board to configure the tag, its layout is shown in Figure 3.6.

The board contains three connection ports, P1, P2 and P3. P1 provides the necessary connection pins required to program the PIC24, it includes the MCLR, VDD, GND, PGD1 and PGC1 pins [34]. Additionally it gives two extra pins RP2 and RP3 that are used to provide a UART connection, where RP3 is configured as the RX pin and RP2 as the TX pin. These two pins allow the microcontroller to be interfaced either to a PC or to an external, second microcontroller. Here, the UART interface is used to connect the PIC24 to the PIC18, which is controlling the potentiostat.

P3 provides all the connection to interface the microcontroller to the tag, which is interfaced via SPI. Therefore the microcontroller PIC24 is the Master and the tag is the Slave. The code for the PIC24 that is provided by ams is written in C using the MPLAB X IDE and the XC16 compiler. The microcontroller was

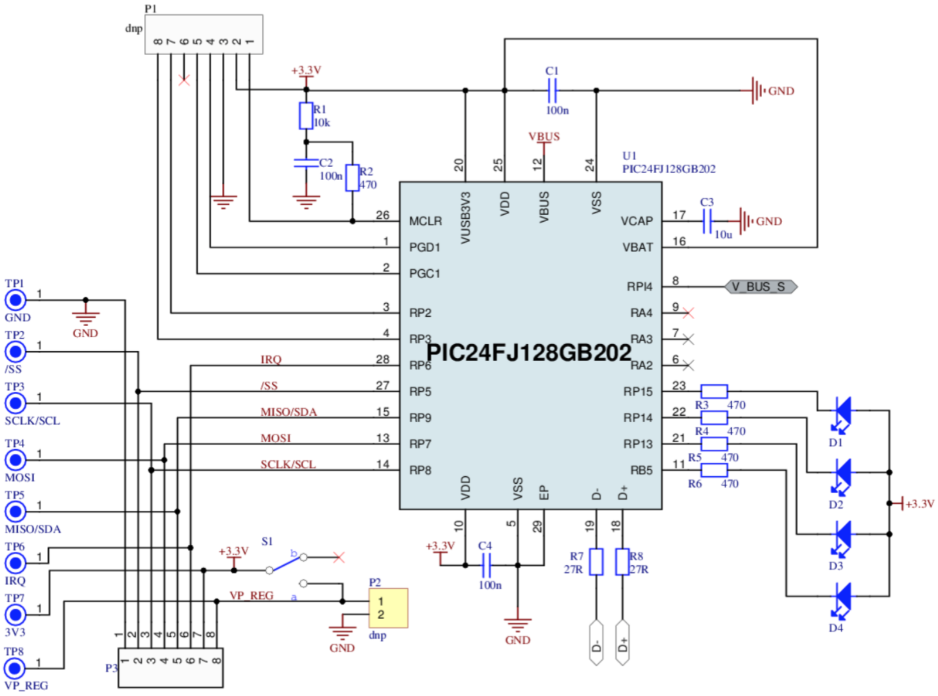


Figure 3.6: PIC24FJ128GB204 connection [7]

debugged with the debugging tool PICkit 3.

The last port, P2 provides two output pins, VP_REG and GND, which are used as the external voltage pins, as described in Section 3.4.2.

3.4.2 Power Modes

The AS3955 tag supports two power sources, either being powered from an external power source, such as a battery, or to be powered by harvesting energy from the NFC [8]. As one of the main objectives is for the tag to harvest energy to power the system, the latter option is chosen. Figure 3.7 shows the analogue front end (AFE) of the tag.

The AFE includes a rectifier and two regulators, one external and one internal. The internal regulator provides the power for the circuitry within the tag,

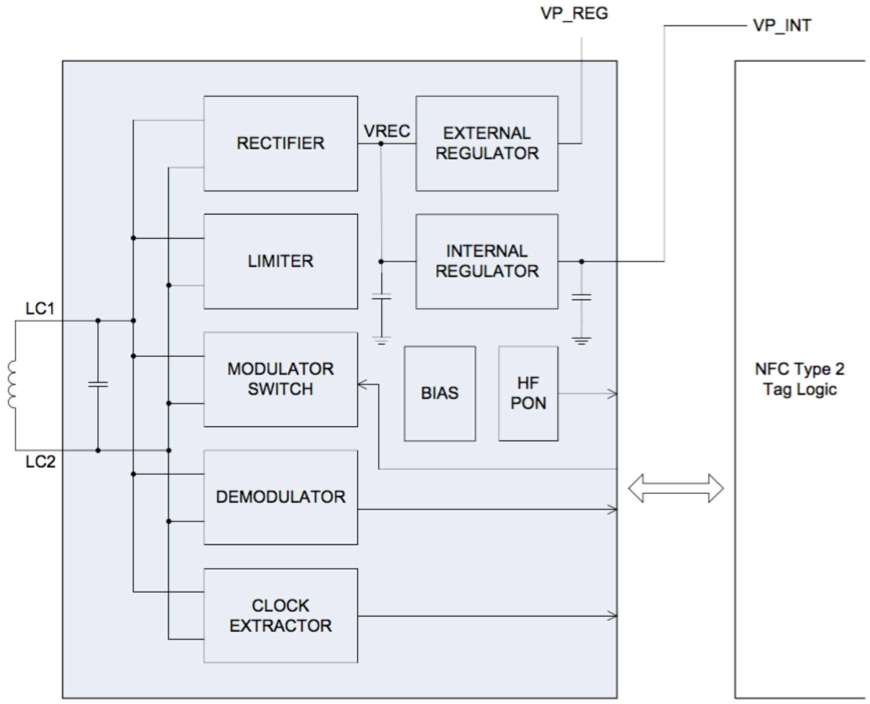


Figure 3.7: AFE of AS3955 [8]

powering the EEPROM and SPI interface. The external regulator provides the power at VP_REG, which is used to power the system. The output voltage can be adjusted using the EEPROM setting, as described in Section 3.4.3.

AS3955 includes four different power modes, depending on whether external power is provided at pin VP_IO or whether the power is fully harvested from the NFC of the phone. The power mode that is required for this project is power mode 1, only using the energy harvested from the NFC [8]. This mode is written within the EEPROM configuration as well.

The range of the output voltage is from 1.8 to 4.5 V and can be set in steps of 100 mV. The regulated output voltage is set by 5 bits, where 0b indicates 1.8 V and 11011b indicates the maximum 4.5 V. The digital value of the output voltage is defined by the following equation:

$$D = V_{REG} * \frac{27}{4.5} \tag{3.2}$$

where D is the digital value of the bits to set the voltage V_{REG} at pin VP_REG. The potentiostat and the microcontrollers all require a supply voltage of 3.3 V, therefore the output voltage is set to 3.3 V.

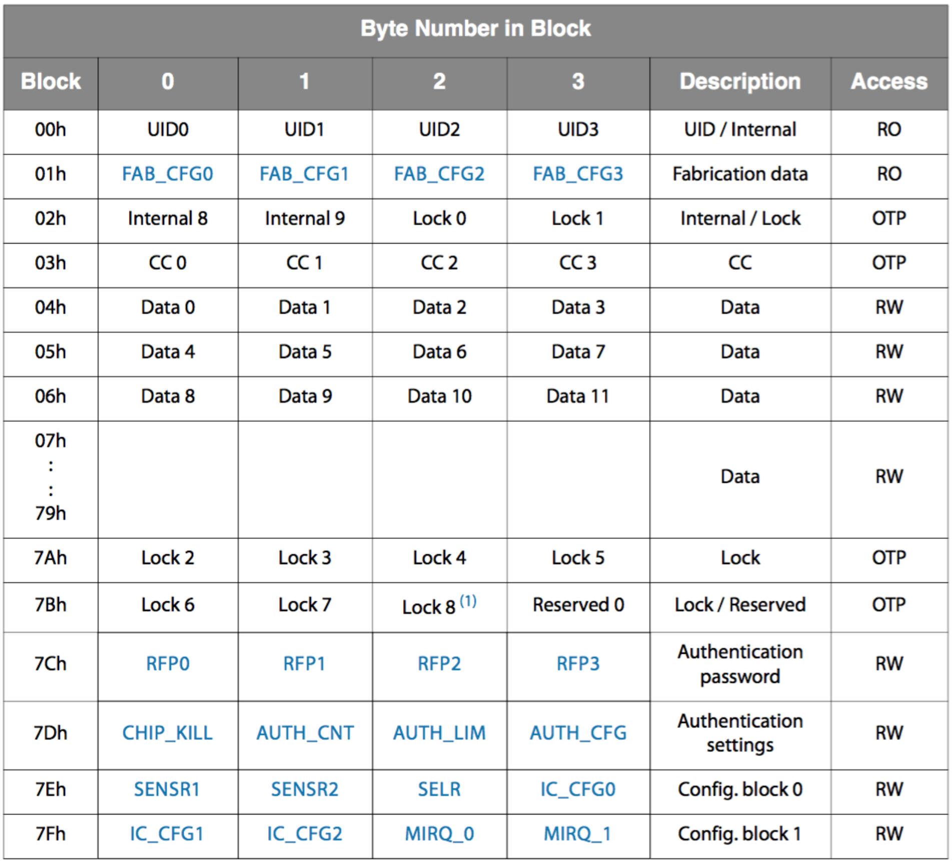
The amount of current that can be supplied, depends on the size of the antenna [8]. The AS3955 includes three different tags, which all have the same function, but different antenna size. In order to receive the maximum amount of current, the largest of the three tags is used. As mentioned in Section 1.2 the system is ideally not bigger than 3 x 3 cm. The size of the large tag has a dimension of 3.2 x 4.5 cm and therefore exceeds the desired proportions.

3.4.3 EEPROM Settings

The AS3955 tag that was used in this system has an EEPROM with a size of 4 kbits. The 4026 bits, or 512 bytes, of data are organised in blocks, each block containing 4 bytes [8]. The first byte is the address of the block, whereas the remaining 3 bytes either store configuration settings of the tag or data. Figure 3.8 shows the block diagram of the EEPROM of the 4 kbits AS3955 tag.

The EEPROM can be accessed via two methods: Either via NFC of the phone or via the PIC24 microcontroller via SPI. For this project only the configuration bytes in block 7Fh need to be set. The byte IC_CFG1 contains the bits to set the regulated output voltage, whereas IC_CFG2 contains the bits to set the power mode. The bytes MIRQ_0 and MIRQ_1 are the bytes for the volatile memory for the Mask Interrupt Registers 1 and 2, which are not used at this point.

The data that is being read from the potentiostat is going to be stored in the



Byte Number in Block						
Block	**0**	**1**	**2**	**3**	**Description**	**Access**
00h	UID0	UID1	UID2	UID3	UID / Internal	RO
01h	FAB_CFG0	FAB_CFG1	FAB_CFG2	FAB_CFG3	Fabrication data	RO
02h	Internal 8	Internal 9	Lock 0	Lock 1	Internal / Lock	OTP
03h	CC 0	CC 1	CC 2	CC 3	CC	OTP
04h	Data 0	Data 1	Data 2	Data 3	Data	RW
05h	Data 4	Data 5	Data 6	Data 7	Data	RW
06h	Data 8	Data 9	Data 10	Data 11	Data	RW
07h : : 79h					Data	RW
7Ah	Lock 2	Lock 3	Lock 4	Lock 5	Lock	OTP
7Bh	Lock 6	Lock 7	Lock 8 (1)	Reserved 0	Lock / Reserved	OTP
7Ch	RFP0	RFP1	RFP2	RFP3	Authentication password	RW
7Dh	CHIP_KILL	AUTH_CNT	AUTH_LIM	AUTH_CFG	Authentication settings	RW
7Eh	SENSR1	SENSR2	SELR	IC_CFG0	Config. block 0	RW
7Fh	IC_CFG1	IC_CFG2	MIRQ_0	MIRQ_1	Config. block 1	RW

Figure 3.8: EEPROM byte layout [8]

data blocks, addressed from 04h - 79h. This allows to store 472 bytes of data within the EEPROM.

3.4.4 Writing the NDEF Message

With the SPI commands there are two modes which are used to communicate with the EEPROM, used either to write to it or to read from it. In order to evaluate the design, the first aim was to write any incoming data to the first data block. Any command via the SPI requires 6 bytes to be sent first. The first byte is the mode command, which must be set to 01000000b or 0x40 to use the SPI in write mode. The second byte is the block address byte as shown in Figure 3.8. The last 4 bytes in the SPI write command are the relevant data, implying that 4 bytes of data can be sent all at once, or updated incrementally, one by one.

The addresses only need 7 bits to be defined. The byte that is used in SPI to write to a particular address starts with the MSB and ends withs a 'Don't Care' bit, as shown in Figure 3.9. Therefore when writing to a particular block, it's address is shifted by a bit. For example, if it is required to write to the address 0x04, the byte 0x08 is used.

	B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
EEPROM Block Address	WA6	WA5	WA4	WA3	WA2	WA1	WA0	x

Figure 3.9: EEPROM SPI address byte [8]

As discussed in the Section 2.3.4, it is necessary to write the data in a suitable NDEF format. The AS3955 uses the TLV format for writing NDEF messages. The TLV format consists of three blocks:

1. T for the type of the TLV block. To send NDEF messages it is set to 0x03.

2. L for the length of the value field.

3. V for the value, which is the NDEF message.

As an example to send the NDEF message [0xD1, 0x01, 0x06, 0x54, 0x02, 0x65, 0x6e, 0x00, 0x10] it would need to be preceded by the bytes [0x03, 0x09]. These are 11 bytes in total and can be written with three SPI commands using the commands [0x40, 0x08, 0x03, 0x09, 0xD1, 0x01], [0x40, 0x0A, 0x06, 0x54, 0x02, 0x65] and [0x40, 0x0C, 0x6e, 0x00, 0x10, 0x00]. Depending on the measurement of the potentiostat, the byte of the L field vary as well as the length of the payload. These lengths can be measured by using buffers and counters within the code, as is described in Section 4.4.

Chapter 4

Testing & Results

4.1 Overview

The design of the testing plan was different to the design of the full system as shown in Section 3.1. As described, two microcontrollers were used during the process, a PIC18 and a PIC24 microcontroller, to help with the implementation. Figure 4.1 shows a high level diagram of the testing plan. When using the microcontrollers, the power supply of the PICkit 3 was used.

4.2 Minimum detectable current

One of the specifications is to be able to measure glucose concentration in a solution. When measuring glucose with electrochemical sensors a very small current is generated, therefore it is necessary that very small currents are detectable in order for the system to be viable.

Before actually testing with glucose sensors and a glucose solution, the system was tested with a Keithley 6221 current source [35]. The system corresponded to Figure 4.2, where the Keithley 6221 provided the current source.

The DAC was set to correspond to the oxidation voltage of glucose at 520 mV, as shown in Table 2.1. Since RE is set at 1.25 by the internal reference voltage, the

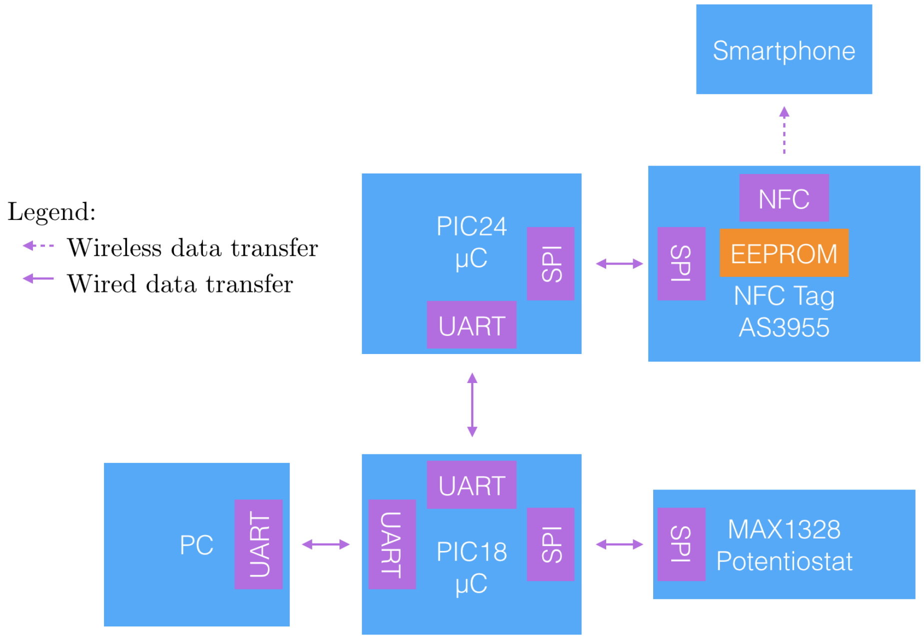


Figure 4.1: Block diagram of testing design

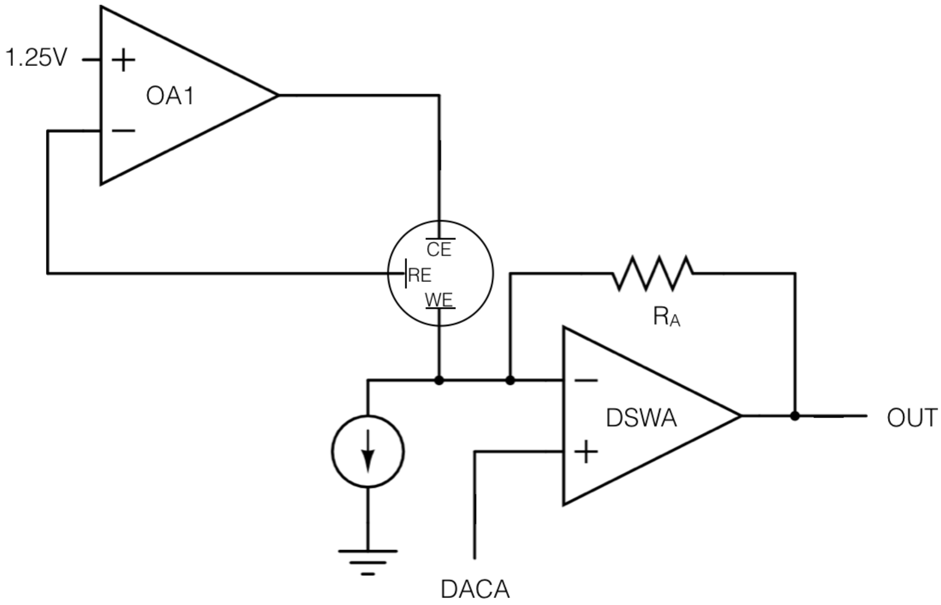


Figure 4.2: Diagram of the potentiostat with current source

DAC must be set to $V_{DACA} - 1.25 = 0.52$, therefore $V_{DACA} = 1.77$. According to equation 3.1, this would require to send to 2900 via the DACA Write command.

Device testing started with a 5 MΩ between the output of the DACA and the WE pin. The output of OUTA was fed to the ADC of PIC18, where it was converted directly with the PIC18's inherent ADC commands. The ADC values were sent to the Serial Development Module UM232R via UART. The UM232R was connected to a PC so that the values could be read. Ultimately the lowest detectable current depends on the resolution of the ADC and the resistor value used. The bit resolution of the ADC is 10 bits, therefore has a resolution of 1024. The range for the ADC is between 1.77 V and 3.3 V, which corresponds to 0 V and 1.53 V. Theoretically the lowest possible current is then $V_{RES} = \frac{1.53}{1024} = 1.5$ mV. Using a 5 MΩ resistor corresponds to a lowest detectable current of 0.3 nA. Conversely, the largest detectable current is $\frac{1.53}{5*10^{-6}} = 306$ nA.

In practice, the supply voltage was 3.1 V, leading to a minimum detectable current of 0.26 nA and maximum detectable current of 266 nA. When driving these currents with the current generator this prediction held true for the max current, as the ADC read 1023, corresponding to the maximum possible value. Any increase in current resulted in the same output.

Without any current flowing through the system the output was approximately 1.77 V, as set by the DAC, this was expected. However the lowest detectable current wasn't 0.26 nA, but instead 2 nA. This is due to the noise. When reading the ADC the digital value fluctuated by around 15 integers, corresponding to noise fluctuation of $15 * 1.3 * 10^{-3} \approx 20$ mV, which in turn $\frac{20*10^{-3}}{5*10^{6}} = 4$ nA. This variation is too large for the system to be viable as such a large fluctuation could seriously affect the readings of any concentration. When the output was sampled at an interval of 200 ms and the average taken at every 10 samples, the output only fluctuated by 3 integers, which corresponds to $3 * 1.3 * 10^{-3} \approx 4$ mV, which in turn $\frac{4*10^{-3}}{5*10^{6}} \approx 1$ nA. This is the output that was expected and justifies that the system is able to measure very small currents. It was reasonable to use the ADC of the PIC18 instead of the ADC of the MAX1329, because ultimately the smallest detectable current is dependent on the noise of the system. However, the power consumption of the PIC18 was too high, possibly due to it's sampling frequency and constant activation mode.

4.3 Energy Harvesting

The NFC is required to power the whole system. The amount of output voltage the tag can supply depends on its setting within its EEPROM, whereas the amount of supply current depends on the size of the antenna. All three tags were tested and all three tags had their output voltage set to 3.3 V. The results are summarised in Table 4.1. None of the tags were able to supply a regulated output voltage of 3.3 V as indicated. Furthermore, depending on how the phone was placed on the tag, the output voltage dropped as low as 2.4 V for the large tag. This implies that the smartphone needs to be placed carefully in order to get the maximum regulated output voltage. For a device that should be simply powered by placing the smartphone on the tag, this variation in supply voltage is less than ideal.

Tag Size	Voltage Range	Smallest Voltage
Small	2.9 to 3.0 V	2.5 V
Medium	2.9 to 3.0 V	2.6 V
Large	2.9 to 3.0 V	2.4 V

Table 4.1: Regulated output voltage

One method to compensate for this was to theoretically set to the regulated output voltage to 4.5 V, the maximum possible supply voltage, and then use an additional voltage regulator to scale down the voltage to 3.3 V. For this test the ZSR330 regulator was used, which is a 3.3 V regulator [36]. However, as this regulator requires an input voltage of larger as 7.3 V, it was never possible to achieve a regulated output voltage 3.3 V that was independent on how the phone was placed on the tag.

The drop in the regulated output voltage can be attributed to the LEDs that are used on the demonstration board. The board uses several LEDs that consume 20 mA at a supply voltage 3.3 V [37]. When further developing the prototype, these LEDs should be removed to decrease the power consumption. Another solution in order to operate the device within the power budget is to operate all devices at a power supply lower than 2.6 V rather than the current 3.3 V. Comparing the smallest possible supply voltages of the components, the PIC24 has the largest one at 2.0 V [31]. Therefore it would be possible to use a supply voltage as low as 2.6 V.

For the prototype the components current consumption was measured and the results are summarised in Table 4.2. Overall the power consumption of the board was 5 mA at a supply voltage of 3.3 V.

Device	Voltage	Current
PIC18	3.3 V	3 mA
MAX1329	3.3 V	1 mA
PIC24	3.3 V	1 mA

Table 4.2: Current consumption

At this point the power consumption of the prototype is too large, especially of the PIC18. This could be attributed to the fact that the PIC18 constantly in active state and was sampling at a frequency of 5 Hz. There are three possible solutions to this: One is to not use the ADC of the PIC18 and instead use the

ADC of the MAX1329 chip. The second suggestion is to sample less frequently, for example at a frequency of 2 Hz. The ideal sampling frequency is ultimately a trade-off between precision and power consumption. This solution can be combined with setting the PIC18 to sleep whenever it is not used. The final solution is to remove the PIC18 and only use the PIC24, an approach that is discussed further in Section 5.1.

4.4 Writing to EEPROM

To test if it was possible to write incoming values from the potentiostat to the EEPROM of the NFC tag, the two microcontrollers PIC18 and PIC24 were used. The first step was to write a random integer variable defined within the `main.c` code to one of the data blocks within the EEPROM. The variable `TEST` was defined and when compiling the code on the PIC24, the variable was written to the block 0x20 via the `as3955WriteEepromWordFull` command. The data within the block was read out using the AS3955 Android demonstration app.

The next step was to connect the two PICs and exchange data via the UART communication. PIC18 was sending the ADC values to the RX pin of PIC24. The UART driver of the PIC24 was modified by creating an interrupt service routine function, which was created to trigger an interrupt as soon as the PIC24 received any data from the PIC18. Any data that was coming would have been stored in the `TEST` variable and written to the EEPROM as described above. However, the interrupt service routine was never evoked when testing. Assuming that all the connections were correct, there could have been two reasons for this. One was that the PIC24 never received any data at its RX pin from the PIC18. This is unlikely, as the PIC18 was still able to send data to the UM232R logger. The second reason was that the interrupt routine was never invoked, because the interrupt enable bit was accidentally reseted elsewhere in the code. More debugging time was required.

Once this step would have been achieved then an algorithm to detect the length of the incoming data would have been employed. As described in Section 3.4.4, the NDEF message that is written depends on the length of the data. Two lengths need to be determined, the length of the payload and the length of the NDEF message. These algorithms would use counters and buffers to store the values temporarily. Once the lengths were determined, the NDEF message could have been correctly written within the EEPROM.

Chapter 5

Further Work

5.1 Combination & PCB

During the implementation, two PICs were used, PIC18 and PIC24. As demonstrated in Section 4.3, this uses too much power and additionally takes too much space. Both the potentiostat and the NFC tag can be accessed via SPI, therefore the first step when improving the prototype is to only use PIC24 as the SPI Master and the two components as SPI Slave 0 and SPI Slave 1. In this configuration, the MAX1329 and AS3955 share one SPI SCK clock, one SPI MOSI line and one SPI MISO line. Both components are selected via the respective CS line of the PIC24.

Careful consideration needs to be taken regarding when the CS line is enabled in order to avoid both peripherals trying to access the shared MISO and MOSI line. When the smartphone is placed on the tag, the CS line for the NFC tag is enabled, the microcontroller can read any incoming data of the EEPROM. Once the data transfer is completed, the CS line for the NFC tag goes high, and the CS line for the potentiostat chip goes low. The MAX1329 and the PIC24 exchange data via SPI and after the transfer, the CS line is disabled for the MAX1329. To complete the loop, the microcontroller writes back to the NFC tag, enabling the CS line once again. This algorithm loop needs to be constructed when merging the PICs.

Once the devices have been integrated onto one microcontroller and the libraries have been merged, the system can be implemented on a PCB. The code for the PIC24 microcontroller can be easily migrated onto the PCB. When designing the PCB, one of the critical design decision to ensure is to that the analogue supply voltage AV_{DD} and the digital supply voltage DV_{DD} are separated into two different supply rails.

5.2 Sensors and Other Metabolites

The system has proven to be able to detect the current generated when measuring the concentration of glucose. The next step would be to actually test the device with electrochemical glucose sensors. Once this has been established, the system needs to be optimised to be applicable for other metabolites. In order to measure other metabolites, the system has to change the DAC voltage accordingly and implement a resistor, which value can be changed accordingly.

5.3 Development of Android App

A bespoke Android app needs to be built for the system. At the moment, all the data was read out with the included AS3955 Android demo app. The bespoke app would allow the user to read any data coming in from the system and send instructions to the device. Any incoming instructions need to be processed by the PIC and change its settings accordingly. So if for example the user wishes to measure a different metabolite, an NDEF message is sent to the device, which includes the instructions. The instructions for the system would be written in the described text 'T' format. The PIC24 is then able to decode the NDEF message and change the voltage of the DAC and the value of the resistor. A lot of online support can be found regarding developing apps for Android that use NFC [28].

Chapter 6

Evaluation

6.1 Comparison to Objectives

The main objective of this project was to develop a low-power device for measuring the concentration of metabolites and using NFC for data and power transfer. The selection of components and the development of libraries formed the basis of this device. The design was tested and comments regarding the test results were provided, which form the improvements that should be incorporated when building this device further by implementing the recommendations. In the end, the completed system is going to use a single microcontroller to communicate with the peripherals, which were able to detect small currents and transmit data. For the device to be tested fully using human interaction, the required sensors need to be put in place. Overall, the objectives were met, validating the design.

6.2 Comparison to Specifications

The specifications of this project were for the device to be able to detect small currents, to use NFC to communicate to a smartphone, to harvest energy via NFC from a smartphone and for the device to be not bigger than 3 x 3 cm. Comparing the device to the related work discussed in Section 2.6.3, which was able to measure currents as small as 1 nA, the prototype proved to be capable of measuring currents to a minimum of 2 nA, which is very close to the related work. The power consumption of the prototype at the moment is too large, mainly due to inefficient use of the given capabilities, such as the low power mode of the PIC24, or the incorporation of power draining components, such as LEDs. The system is able to harvest energy from the smartphone, but the current power consumption of the system needs to be diminished before it can be solely powered via NFC. The system was able to exchange data via NFC to a smartphone, therefore meeting this specification. The majority of the devices used for the prototype had a size smaller than 3 x 3 cm, therefore allowing to build a device that fits within the size criteria.

Chapter 7

Conclusion

During this project multiple milestones were achieved:

1. Using only off-the-shelf components, a prototype was designed and the corresponding libraries were built
2. The system was able to detect very small currents that are similar to the current generated by the electrochemical reactions with metabolites. When sensors are going to be applied to the potentiostat, it should be able to measure the concentration of glucose in the solution.
3. The system communicated via NFC with a smartphone by using a Type 2 Tag. By doing so, it was able to write any data within the memory for the NFC tag, so that the data could be read out via a smartphone.
4. The system was able to harvest energy from a smartphone using NFC. Even though the energy wasn't enough to power the whole device, explanations and improvements were detailed on how this can be achieved.
5. As only very small components were selected, the device can be built within a dimension of around 3 x 3 cm

Although it wasn't possible to meet all the specifications within this prototype, this project has shown that it is possible to design a device that can include all of these objectives. Once the further work has been implemented it will be possible to build a state of the art device that allows diabetic patients to measure their glucose levels with the device and their smartphone. The device will not require a battery and can therefore run theoretically indefinitely.

Shortcomings were encountered at the beginning of the project. When researching for RFID and how to harvest energy, a lot of time was spent on trying to back-engineer the technology. Instead of finding applicable development boards, initial suggestions on how to design this project included designing a PCB, which incorporates the RFID antenna, to extract energy for the system. Another difficulty was to understand on how to use the MAX1329 chip. Apart from its datasheet there isn't any documentation available that could have helped with its development. For example it took multiple days to get the SPI interface working. This situation was exacerbated by the fact that it was initially attempted to program the chip with a PIC24. As described, using the SPI on the PIC24 was initially more difficult than using the SPI on a PIC18. These shortcomings are expected as an engineer and contributed to a valuable experience. Hopefully, when this project may be picked up in the future, this report will form a helpful start.

Bibliography

[1] Wei-Song Wang, Wei-Ting Kuo, Hong-Yi Huang, and Ching-Hsing Luo. Wide dynamic range cmos potentiostat for amperometric chemical sensor. *Sensors*, 10(3):1782, 2010.

[2] Dan Coleman Tom Igoe and Brian Jepson. *Beginning NFC*. O'Reilley Media, Sebastopol, CA, 2014.

[3] Yi Zhao, J.R. Smith, and A. Sample. Nfc-wisp: A sensing and computationally enhanced near-field rfid platform. In *RFID (RFID), 2015 IEEE International Conference on*, pages 174–181, April 2015.

[4] Matthew D. Steinberg, Petar Kassal, Irena Kerekovi, and Ivana Murkovi Steinberg. A wireless potentiostat for mobile chemical sensing and biosensing. *Talanta*, 143:178 – 183, 2015.

[5] N. Anabtawi, S. Freeman, and R. Ferzli. A fully implantable, nfc enabled, continuous interstitial glucose monitor. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 612–615, Feb 2016.

[6] Maxim. Max1329. `https://datasheets.maximintegrated.com/en/ds/MAX1329.pdf`.

[7] AMS. As3955 demo kit manual. `http://ams.com/eng/content/download/746267/1895469`.

[8] AMS. As3955. `http://ams.com/eng/content/download/765130/1929267/389115`.

[9] S. S. Ghoreishizadeh. *Integrated Electronics to Control and Readout Biosensors for Implantable Applications.* PhD thesis, Ecole Polytechnique Fdrale de Lausanne (EPFL), 2015.

[10] Ian Poole. Nfc near field communication tutorial. `http://www.radio-electronics.com/info/wireless/nfc/near-field-communications-tutorial.php`.

[11] Diabetes UK. Diabetes: Facts and stats. `https://www.diabetes.org.uk/Documents/About20Us/Statistics/Diabetes-key-stats-guidelines-April2014.pdf`.

[12] Deloitte. Mobile consumer 2015: The uk cut. `http://www.deloitte.co.uk/mobileuk/assets/pdf/Deloitte-Mobile-Consumer-2015.pdf`.

[13] F. Honoré, B. Otis, and A. Nelson. Reader communication with contact lens sensors and display device, December 30 2014. US Patent 8,922,366.

[14] Khurram Shahzad and Bengt Oelmann. A comparative study of in-sensor processing vs. raw data transmission using zigbee, ble and wi-fi for data intensive monitoring applications. pages 519 – 524, Barcelona, Spain, 2014. BLE;Bluetooth low energies (BTLE);IEEE 802.15.4;Sensor processing;Wireless sensor;.

[15] iHealth. Wireless smart gluco-monitoring system. `https://ihealthlabs.com/glucometer/wireless-smart-gluco-monitoring-system/`.

[16] OneTouch. Onetouch verio sync meter. `http://www.onetouch.com/veriosync`.

[17] Roland Minihold. Near field communication (nfc) technology and measurements. `https://cdn.rohde-schwarz.com/pws/dl_downloads/`

dl_application/application_notes/1ma182/1MA182_5E_NFC_WHITE_PAPER.pdf.

[18] Stephen Weis. Rfid (radio frequency identification): Principles and applications. http://www.eecs.harvard.edu/cs199r/readings/rfid-article.pdf.

[19] NXP. Nfc for embedded applications. https://www.nxp.com/documents/brochure/75017587.pdf.

[20] Texas Instrument. Lmp91000. http://www.ti.com/lit/ds/symlink/lmp91000.pdf.

[21] Analog Devices. Aducm350. http://www.analog.com/media/en/technical-documentation/data-sheets/ADuCM350.pdf.

[22] NXP. Nt3h1101/nt3h1201. http://www.nxp.com/documents/data_sheet/NT3H1101_1201.pdf.

[23] AMS. As3953a. http://ams.com/eng/content/download/325044/1140864/78495.

[24] ST. M24lr04e-r. http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00038059.pdf.

[25] Texas Instrument. Rf430frl152hevm. http://www.ti.com/lit/ds/slas834c/slas834c.pdf.

[26] Quick2Wire. I2c and spi. http://quick2wire.com/articles/i2c-and-spi/.

[27] Apple. Wallet. https://developer.apple.com/wallet/.

[28] Android. Near field communication. https://developer.android.com/guide/topics/connectivity/nfc/index.html.

[29] Yuntao Liu, Min Chen, Zhichao Li, and Mingyuan Ren. A dual-channel wide input range interface circuit for electrochemical amperometric sensors. *IEICE Electronics Express*, 12(17), 2015. High resolution;Input and outputs;Input current;Interface circuits;One-to-one mappings;Sigma delta ADC;TIA;Wide range;.

[30] Hai Guo, Y.Y. Cai, D. Cao, X.H. He, and Q.X. Wang. 2012 international workshop on information and electronics engineering continuous glucose monitoring system based on smart phone. *Procedia Engineering*, 29:3894 – 3898, 2012.

[31] Texas Instrument. Pic24fj128gb204. `http://ww1.microchip.com/downloads/en/DeviceDoc/30005009c.pdf`.

[32] Microchip. Pic18f25j11. `http://ww1.microchip.com/downloads/en/DeviceDoc/39932D.pdf`.

[33] FTDI Chip. Um232r. `http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM232R.pdf`.

[34] Microchip. Mplab x - using pickit 3 in-circuit debugger. `http://ww1.microchip.com/downloads/en/DeviceDoc/50002010B.pdf`.

[35] Keithley. Model 6221 ac and dc current source. `http://www.tek.com/sites/tek.com/files/media/media/resources/6220-6221.pdf`.

[36] Zetex. Zsr330 datasheet. `http://www.diodes.com/_files/datasheets/ZSR.pdf`.

[37] KINGBRIGHT. Smd chip led lamp. `http://www.farnell.com/datasheets/2046637.pdf`.