ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO DE TELECOMUNICACIÓN

Título del proyecto:

DESIGN AND IMPLEMENTATION OF A HARDWARE PROTOTYPE FOR PERSONAL HEALTH DATA IDENTIFICATION

Adrián Catón Oteiza

Supervisor: Dr. Ing. Luis Serrano Arriezu

Examinador 1: DI Ferenc Gerbovics

Examinador 2: Dr. Stefan Sauermann

Viena, 04/05/2012

FH Technikum Wien

MASTERS THESIS

for an academic degree
"Master of Science in Engineering"

Design and implementation of a hardware prototype for Personal Health Data Identification (PHDI)

written by **Adrián Catón Oteiza**
1190, Wien, Hutweidengasse, 35/8/3

Examiner 1: DI Ferenc Gerbovics

Examiner 2: Dr. Stefan Sauermann

Vienna, 18. July 2012

FACHHOCHSCHULE TECHNIKUM WIEN

Written at University of Applied Sciences Technikum Wien

Study Programme, Master Biomedical Engineering

Affidavit

"I hereby declare by oath that I have written this paper myself. Any ideas and concepts taken from other sources either directly or indirectly have been referred to as such. The paper has neither in the same nor similar form been handed in to an examination board, nor has it been published."

Vienna, 18. July 2012
Place, Date

Signature

Abstract

Background

The rapid aging of European population is increasing the demand for residential care thus, the expenditures are dramatically increasing. The costs of healthcare are getting unsustainable therefore technological devices are a necessary tool to extend the period of time during which elderly people can safely stay at their own homes.

These technological devices can also be used to encourage people to practice more sport, therefore reducing healthcare costs.

This paper presents a device that meets all the needs for elderly people care and for fitness applications.

Methods

Hardware designed to meet the requirements focusing on the needed components and their data sheets.

Results

A fully functional prototype has been achieved with reduced size, capable of measure acceleration, heights and temperature. With voice output, an RFID reader and ANT module for wireless communication. Also a decent power consumption has been achieved.

Conclusion

The main result that can be extracted from this thesis is the knowledge on how to design a Personal Health Data Identifier device which can have many different applications even as a Personal Health Device depending on the firmware implemented.

Resumen

Antecedentes

El rápido envejecimiento de la población europea está incrementando la demanda de cuidado residencial y por tanto, los gastos están creciendo dramáticamente. Los costes sanitarios empiezan a ser insostenibles, de ahí que los dispositivos tecnológicos sean una herramienta indispensable para extender el período de tiempo durante el cual la población anciana puede permanecer de forma segura en sus propios hogares.

Estos dispositivos tecnológicos también pueden ser empleados para motivar a la población a practicar más deporte y por tanto, reducir los costes sanitarios.

El presente documento presenta un dispositivo que cumple con todas las necesidades para el cuidado de personas mayores así como para aplicaciones deportivas.

Métodos

Se ha diseñado un hardware que cumple con todos los requerimientos. Para ello se ha hecho uso de las hojas de características de los diferentes componentes necesarios.

Resultados

Se ha logrado fabricar y probar un prototypo completamente funcional y de tamaño reducido, capaz de medir aceleraciones, alturas y temperatura. Además dispone de interfaz con el usuario vía mensajes de audio. También dispone de un lector de etiquetas RFID así como un módulo de comunicaciones inalámbricas de bajo consumo ANT. También se ha conseguido que el consumo global de energía sea aceptablemente reducido.

Conclusiones

El resultado principal que puede ser extraído de este proyecto fin de carrera es el conocimiento sobre cómo diseñar un dispositivo de identificación personal de datos médicos que puede tener muy diversas aplicaciones, incluso como dispositivo de salud personal (PHD) dependiendo del firmware implementado.

Keywords: Health, Data, Identification, Hardware, Prototype, PHDI, PHD

Acknowledgments

First of all, I would like to deeply thank my family and friends for being always there when I needed support or help (which happened to be everyday in this degree). Specially my mother, father and brother for stoically listen my many complains and for giving me hope to keep working hard until this degree and thesis were finished. Otherwise, none of this would have been possible.

I would also like to thank the professors that did a great job and put a lot of efforts to transmit us not only the knowledge but the passion towards it's subjects (sadly they turned to be just a few of them) because we will never be able to pay back what they gave us.

Special thanks to our project leader Dipl. Ing. Ferenc Gerbovics for his continuous support, guidance and encouragement. And for giving us the chance to participate in such an exciting project.

Dr. Stefan Sauermann, representative for the Biomedical Department, to make this internship at the Technikum Wien possible.

Michael Haller BSc for his support and continuous help.

All the PHDI project team members: Mona Janbozorgi, Xin Lu, Elsa Moriones, Jiliang Wang and Gregor Wiktorin for their great job and support.

Finally, Elsa Moriones, again, for everything else.

Table of Contents

1 Introduction

The fact that Europe population is getting older is a truth that has to be faced with sufficient anticipation to reduce the increasing costs of elderly care and the shrinking share of active population. [1] As seen in the next figure, half of the population over 15 years old is expected to be retiree by 2050. [2]

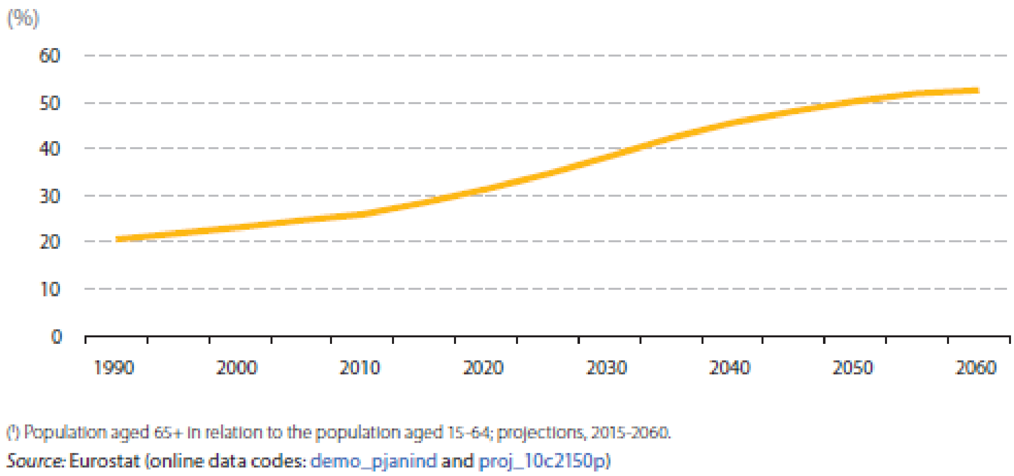


Figure 1: Old-age dependency ratio, EU-27. [2]

In addition, elderly people is prone to visit the general practitioner much more often than younger sectors of the population therefore generating extra costs in the system. [2]

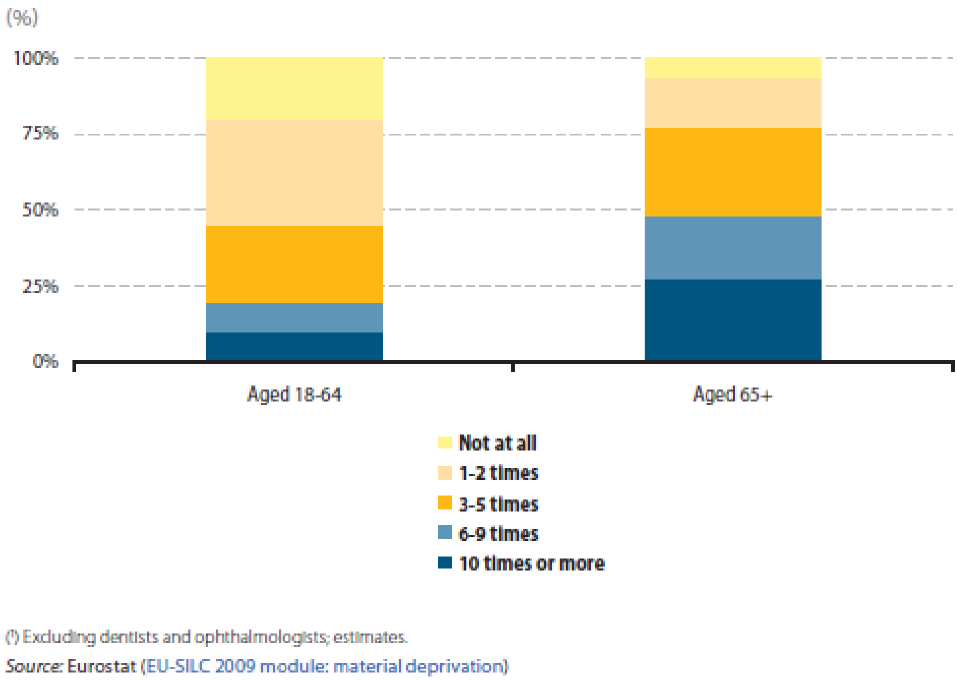
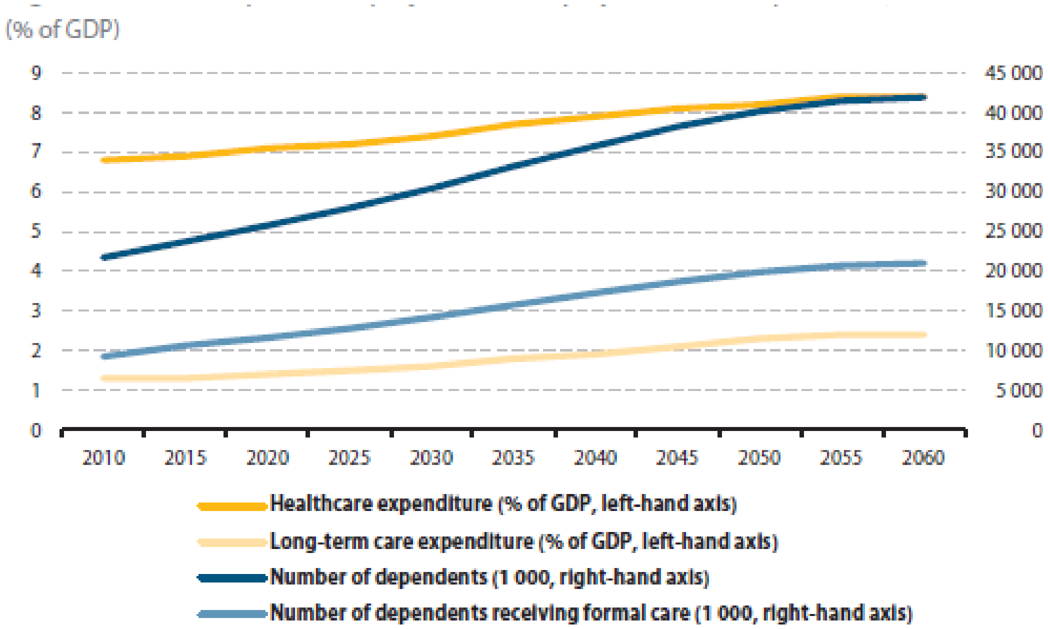


Figure 2: Frequency of consulting a GP or specialist by age, EU-27, 2009. [2]

For all the reasons explained above, health care expenditure is expected to rise up to more than 8 percent of the GDP of the European Union. Therefore, new ways for reducing costs have to be developed in order to keep the sustainability of the system.



Source: European Commission, European Economy, 2/2009 - Economic and budgetary projections for the EU-27 Member States (2008-2060)

Figure 3: Care expenditure projections for dependents, EU-27. [2]

One way for reducing costs is the home caring of elderly people using monitoring devices such as the PHDI, designed in this document. It is known that home care keeps families together, which is key for ill people. Also, home is where most people want to be when they are sick. Home care is personalized and keeps the individuals freedom intact while preventing the common depression caused by institutionalization. [3]

This kind of home care devices will extend the period of time during which elderly people can be at their own home, thus reducing the already crowded medical facilities as many governments are starting to do in Europe. [4]

This device will provide more self confidence to elderly people, probably reducing the frequency of attendance to the GP.

It will also allow a faster medical response in case of emergency or fall which will in many cases reduce the mortality of some events such as heart attacks, embolisms, etc. [5]

A second field of application of the PHDI is fitness. It is also a great way to reduce costs in health care since it has been proven that exercising regularly is a great and cheap medicine for increasing health [6]. In the figure below, the impact of practicing exercise on mortality is shown for participants with various preexisting risk factors.

Other sources also state that physical inactivity is a risk factor for cardiovascular diseases and its prevalence. [7]

Therefore, fitness encouraging systems such as the PHDI are key to make people keep practicing regular exercise. However, proper screening and evaluation are important to find persons with underlying cardiovascular disease prior begin practicing exercise at moderate to vigorous levels [7]. Thus, the PHDI also provides an easy way to integrate all the monitoring systems in a fitness facility, identifying all the medical data with the corresponding clients automatically with minimum user interaction.

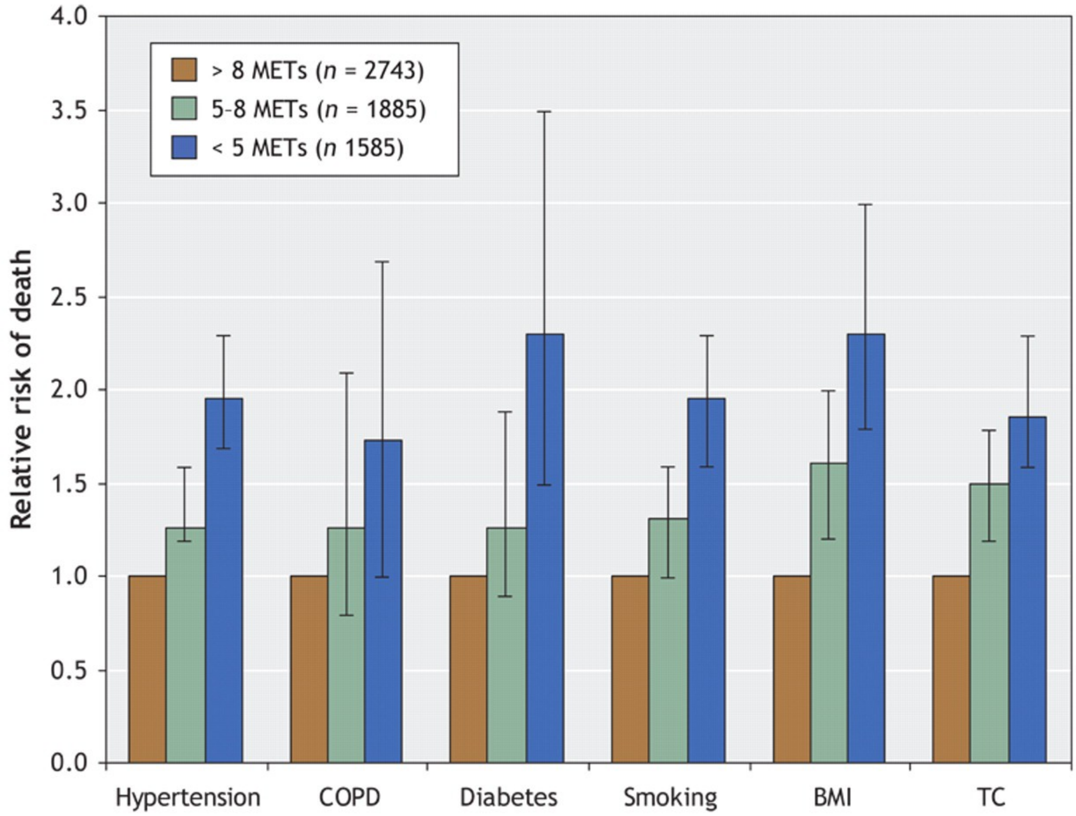


Figure 4: Relative risks of death from any cause among participants with various risk factors.

(e.g., history of hypertension, chronic obstructive pulmonary disease [COPD], diabetes, smoking, elevated body mass index [BMI ≥ 30] and high total cholesterol level [TC ≥ 5.70 mmol/L) who achieved an exercise capacity of less than 5 METs (metabolic equivalents) or 5–8 METs, as compared with participants whose exercise capacity was more than 8 METs. Error bars represent 95% confidence intervals. Adapted, with permission, from Myers et al. External Ref. (*N Engl J Med* 2002;346:793-801). [6]

Copyright © 2002 Massachusetts Medical Society. All rights reserved.

This diploma thesis aims to explain the design and implementation of a biomedical wearable device for Personal Health Data Identification (PHDI) capable of fitting all these needs. This wearable device is based on previous projects designed and developed by the Biomedical Engineering Sciences department of the FH Technikum Wien.

This work is only a small part of the whole PHDI project which is divided in smaller chunks each of which are completed by a team member. A basic summary of the different parts of the project can be seen in the figure below.

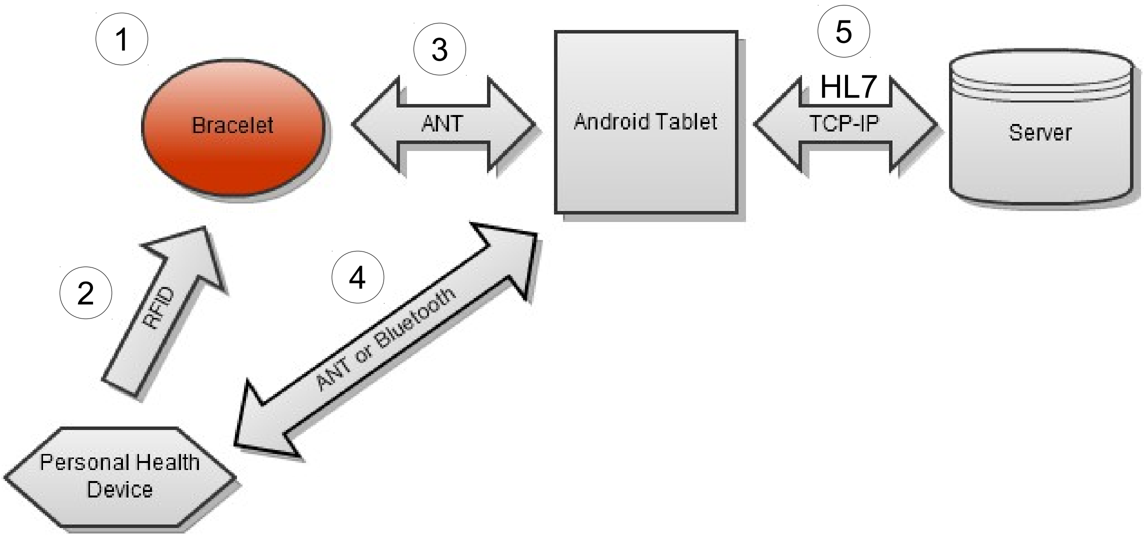


Figure 5: Summary diagram of the PHDI project.

In particular, the **bracelet hardware design** and some low level firmware implementation, which will be detailed in the upcoming sections, is covered in this thesis. Meanwhile other team members deal with most of the bracelet **firmware**, **Android** tablet programming and the **back-up server with web front-end**.

The PHDI acts as a personal identifier for the wearer in the fitness facility use case. When a new patient or client is added to the system, a patient ID is stored in the PHDI bracelet. This adding a new client process is done via the Android tablet, where the name of the client along with his or her medical data is stored.

When the user wants to use a Personal Health Device, he or she has to press the button located in the bracelet (1). Then the bracelet starts searching for the PHD ID number using the built in RFID reader (2). Once the bracelet has the PHD ID it sends the patients ID along with the PHDI ID to the Android tablet via ANT (3). Of course, the AT should be in the range of the PHDI.

ANT is a proprietary wireless protocol which main advantage is to be ultra-low power consuming and easy to use. *"Designed for 2.4 GHz operation, ANT is perfectly suited for any kind of low data rate sensor network topologies in personal area networks (PANs)"* [8]

When a PHD ID is received by the Android tablet via ANT it starts communication with that particular PHD via ANT or Bluetooth in order to get the health data and store it in the electronic health record (4). Therefore, the Android tablet is supposed to have a database where all the patients names, IDs and health records are stored and managed. The final part of the project is the back-up server. It communicates with the Android tablet via TCP-IP and stores the data properly (5). In addition, a web server is implemented in the back-up server as a second way to remotely access the records by patients or by staff.

The transmission of data between the AT and the server is done using the Health

Level 7 (HL7) standard. This standard is explained in more detail in the Health Level Seven (HL7) section.

Finally, for the elderly care use case, basic vital parameters such as movement and temperature are acquired (in a future version, pulse could be acquired as well) and transmitted to the remote Android Tablet. If a special event like a fall is detected by the PHDI, an alert is transmitted to the AT which deals with it accordingly, for example sending an SMS or placing a call to the contact number.

1.1 PHDI predecessor

The PHDI predecessor was called Health Data Hub [9]. Figure 6 shows the developing prototype. The HDH is a medical device which main objective was to measure patient's pulse and activity and transmit them in a reliable and standardized (HL7) way.

In order to achieve this, the hardware used was the following:

- Sensor modules
  1. Pulse meter for acquisition and amplification of the heart rate via a piezoelectric sensor element.
  2. Accelerometer for activity measurement
  3. Altimeter for additional information about the activity
  4. Temperature sensor to measure the environmental temperature of the device.
- Communication modules for wireless data transmission
  1. ANT+
  2. Bluetooth
- OLED display with touch-screen function
- Micro USB for data communication and power supply for recharging the battery.
- Micro SD card reader for local storage management
- Real-time clock unit
- Power supply (battery)

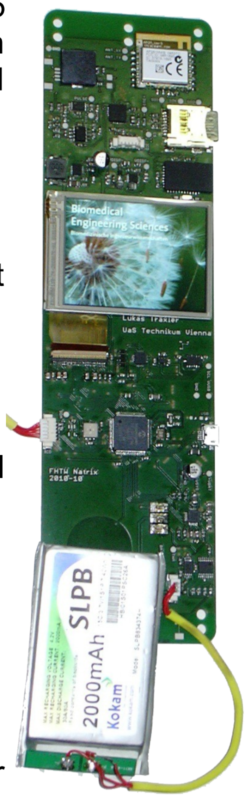


Figure 6: HDH first prototype

This configuration worked pretty well from the beginning besides of the artifacts detected in the pulse signal which are beyond the focus of this study. Nevertheless some issues were detected and have to be corrected in the next version of the device, the PHDI. Among the things that didn't work properly this thesis will focus on this four:

- Battery duration was too short
- Dimensions were too big
- Pulse measurement was noisy and not comfortable for a full day wearable

device

- Standardization of medical devices is very complicated and expensive

The design requirements of the PHDI are based on these findings to achieve better product specifications.

1.2 PHDI bracelet requirements

The prerequisites of the PHDI are:

- Manufacturing cost per unit lower than $100. When manufacturing more than 1000 devices.
- Sensors for movement and vital parameters.
- Small size.
- ANT and RFID communication protocols capable.
- No screen, only LEDs and one button.
- Speech capabilities as user interface from the bracelet to the wearer.

Optional goals to be achieved are:

- One week of battery duration.
- Waterproofness
- Fall detection algorithm
- Wake-up detection algorithm

1.3 Use cases

A device like the one described above can have many different applications on several fields. Moreover, the firmware can be modified as requested to fit almost any application. This first prototype aims at two specific use cases: a sports facility and elderly people care.

The first use case studied, the sports facility, is supposed to be any sports company with tens or hundreds of customers and several Personal Health Devices. The task of the PHDI in this situation would be to uniquely identify the person wearing it providing both access control to the different areas as well as identify the medical data coming from the PHDs. This identification information is transferred to a central computer to be processed and/or stored.

In the second use case, elderly people care, the task of the PHDI would also be identification of the patient, medication reminders via voice output, fall detection, wake up detection and emergency request via press button.

1.4 State of the art

The state of the art in personal health monitoring devices like the one described above is changing. Specially in the fitness field. Actors such as Nike are now developing interesting tools for activity monitoring and they have already achieved many of the goals described in the preceding section. [10]

Two use cases are the most common aim for these devices: sleep monitoring [11], [12], [13] and activity monitoring [10]. Even some of them do both things at the same time [14], [15].

The PHDI is also able to do both sleep and activity monitoring via it's accelerometer even though none of these are being developed for this first prototype. Fall detection of elderly people is one of the objectives of the PHDI even though some devices with that functionality are already in the market [16], [17], [18], [19].

Identification of the wearer for medical data manipulation and storage is the second big aim of the PHDI. In this field, not much has been done besides the widespread passive RFID tags that usually have much less functions and potential than the PHDI due to their lack of a processing core and more advanced communication protocols. [20]

1.5 Health Level Seven (HL7)

“Founded in 1987, Health Level Seven International (HL7) is a not-for-profit, ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services.” [21]

HL7 is also the name of one of their most important standards, a messaging protocol. The name HL7 refers to the seventh layer of the International Organization for Standardization (ISO) communication model for Open System Interconnection (OSI): the application layer. This means that HL7 relies on other communication protocols such as TCP-IP, Bluetooth... for proper message delivery and it only implements the last level of the OSI model, hence it's name. [9]

In the context of this project, HL7 has been used as a standardized way to transfer health data from the Android tablet to the server. Since it's implementation has been driven by other team members, no deeper study is done in this document.

1.6 Eagle CAD

The design of the hardware has been performed using Eagle CAD version 5.8.0. Eagle is a Computer-Aided Design software that offers an integrated environment to develop hardware, from the schematics to the PCB design. It also includes auto-route functionality and it performs back-annotation between the schematics and the board.

The **schematic editor** is a simple tool where electric schematics can be drawn. First, the symbols of the components in use are needed. This symbols can be either created with the **library editor** tool, downloaded from Internet, where many libraries from hobbyists are available; or taken from the built-in symbols that Eagle comes with. Once all the devices are in place, the wiring phase starts. This can be done wiring two pads together or using names and labels in between the pads. All the pads or wires with the same name are tied together creating what Eagle calls *signals.* This second method is the one mainly used in this paper for clearance and readability reasons. The last thing to do when designing a schematic with Eagle is to run the Electrical Rule Checker tool. This tool checks if all the wires are correctly connected and all the junctions correctly drawn.

Once the schematic is completed it has to be checked and rechecked (if possible for a different person). Only then the **board editor** tool can be used. This tool helps drawing the layout of all the components and the routing between them. Also the dimensions of the physical board can be drawn in a separate layer. First, all the signals are unrouted signals, called *air wires* in Eagle terminology. They are drawn by the software with a different color, just from the starting pad to the ending pad. The task of the designer is to place all the components in their position and to route all the *air wires.* This can be done also automatically with the auto-route tool however, this is not recommended due to the poor results that usually are obtained.

When all the signals are routed the Design Rules Checker tool can be used. The DRC checks if the design complies with the manufacturer capabilities in terms of route widths and spacing and also if there are design mistakes like overlaps.

Finally the Eagle files are quite standard in the industry and therefore can be directly sent to the manufacturer of the PCB.

1.7 Microchip's MPLAB X

To write the source code, compile and debug it, the software MPLAB X from Microchip was used. It provides several very useful tools that make the task of firmware development much easier. Among the tools, one can find:

- Tabbed browsing of source files
- Tree view of project source files
- Embedded compiler
- Embedded debugger with breakpoints and step by step capabilities.

2 Methods and implementation

This section is an explanation of the steps that were taken to design the hardware of the PHDI, as well as some of it's firmware.

2.1 Design of the schematics

Once the hardware requirements were set, the schematic of the first prototype started the design phase. This design is a succession of different revisions and fixes. Explaining all this issues and fixes is not the primary aim of this document. Even though some comments are done regarding some issues, only the final result is explained in detail in this document.

2.1.1 Microcontroller

The microcontroller is the heart of the system, it is responsible for everything that happen in the circuit and therefore many aspects have to be considered to choose the best microcontroller for a specific application. Starting with the architecture of the processor thus the manufacturer of the device finishing with the number and type of the peripherals embedded in the integrated circuit.

Nevertheless, there are two more rules to take into account, the first one states that using the same components on several boards reduces the costs of the purchases [22] as well as the time developing the firmware. The second rule asserts that *"if power consumption needs to be low, then RISC is probably the better (CPU) architecture to use"* [23].

Given the previous rules, the microcontroller chosen was from the well known **Microchip** brand because of the previous knowledge of the team members. The specific model is the **PIC32MX795F512H** of the PIC32 family which has a RISC core and has been used in previous designs such as the HDH.

Among the peripherals available in this microcontroller, one can find timers, Pulse Width Modulators, Serial Peripheral Interfaces, Inter-Integrated Circuit interfaces, Universal Asynchronous Receiver Transmitter interfaces, and many General Purpose Input Output ports. [24] This means that all the necessities of the project should be entirely covered.

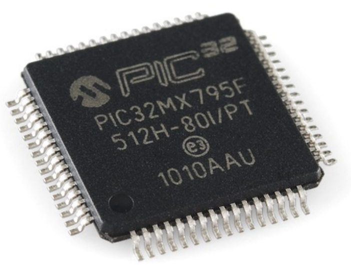


Figure 7: Microcontroller

Finally, the package of the microcontroller has to be chosen to comply with the size restrictions of the project. Therefore the TQFP package (Figure 7) and H version of the microcontroller is chosen which has 64 pins in contrast with the L version that has 100 pins.

The last reason to choose this 32 bits microcontroller is that this microcontroller has the ability to decode mp3 files via software, using a library provided by microchip.

However, during the early stages of development, only RAW audio or WAV files are going to be played for simplicity reasons.

Crystal oscillators

As seen in the schematic of Figure 31, the microcontroller uses two crystals for timing, one slow crystal of 32,768 kHz for the Real Time Clock and Calendar that allows a low power consumption mode while keeping the RTCC and a faster 8 MHz crystal for the normal operation.

Power and reset

A regulated 3.3 volts power source is provided to the microcontroller VDD pins 1 to 4 as shown in the Figure 31 as well as to the AVDD and VUSB pins used by the Analog to Digital Converter peripheral and the Universal Serial Bus peripheral of the microcontroller. Although the USB peripheral is not used, the microcontroller datasheet states that this VUSB pin should be connected to VDD. [24]

For power saving reasons a separate 1.85 volts power source has been applied to the VDDCORE pin. This pin is connected to the internal linear 1.8 volts regulator output. Due to the 0.05 higher voltage applied to this pin, the internal linear regulator is bypassed and hence not used. The reason for doing this is to avoid the higher power consumption of the linear regulator in comparison with the switching regulator that will be used to provide the power to the entire circuit as seen in the next section: Battery protection, power regulation and monitorization.

The MCLR (reset pin of the microcontroller) is connected to VDD using a 100kΩ resistor to limit the current sunk when pulling down the MCLR to call a reset. [24]

Decoupling capacitors

It is very common and necessary to use decoupling capacitors between power pins and ground in every digital Integrated Circuit and as close as possible to it [23]. The reasons are mainly two:

First, the capacitor decouples the noise from the power source and provides a path to ground for it.

Second, every digital system changes it's status on every clock rising (or falling) edge, changing thousands of transistors at a time thus consuming different amounts of power on each tic and generating noise that could be radiated by the PCB routes from the IC to the power supply. However, when a decoupling capacitor is used the power fluctuations that could radiate noise are sourced from the close capacitors instead of the main power source, avoiding the long routes (the longer the routes are, the more radiated energy [25])

2.1.2 Battery protection, power regulation and monitorization

The previous device designed by the Biomedical Engineering Department, the HDH, had three separated ICs for each one of this tasks. One for battery protection, one switching regulator and one battery gauge. Hence the first designs of the PHDI bracelet had three separated ICs as well.

However due to the dimensions restrictions that were to be faced, the **LTC3558** from Linear Technology appeared to be a much better option for our design, providing battery protection and power regulation in a single chip. In addition to that, the LTC3558 does not need any external MOSFET transistors to stop the charging process unlike the former battery protector making it even smaller. [26]

The battery charge current is programmed via a resistor between the PROG pin of the LTC3558 and ground. The design shows that a 1K74 Ohm resistor is used to provide a 500 mA current to the battery. However, during the testing phase other resistors have been used sometimes to test the inductive charger. The value of the resistor for a given current can be obtained using the following equation.

$$I_{BAT} = \frac{PROG \cdot 800}{R_{PROG}} \qquad (1)$$

Two data lines are connected from the LTC3558 to the microcontroller. Table number 1 shows how this lines are connected between the LTC3558 and the microcontroller. The PMIC_CHRG output data line provides the status of the charging process to the microcontroller. It advises of one of this four status: charging, not charging, bad battery or battery temperature out of range. The second data line, PMIC_HPWR, is an input that can be used by the microcontroller to change from fast charge mode to slow charge mode. In fast charge mode the current that is provided to charge the battery is the one calculated with the equation 1 while when the slow charge mode is selected, the current provided is a fifth of this programmed current.

Power regulator	Microcontroller
PMIC_CHRG	CN6/RB4
PMIC_HPWR	CN7/RB5

Table 1: Power regulator connections

As a battery monitor, the **LTC2942-1** from Linear Technology is used, providing the charging status and the voltage measurement in a small 6-Lead (2mm by 3mm) Plastic DFN package. The stored values can be accessed via the I2C bus using the address **1100100**.

The battery monitor schematic is provided also in the Figure 32. The association between the battery monitor logic lines and the microcontroller ports are summarized in the following table.

Battery monitor	Microcontroller
I2C bus	I2C1 peripheral

Table 2: Battery monitor connections

Battery

The main goals to be achieved by the battery are very long life (one week) and small dimensions. However, there is a tradeoff between capacity and dimensions that can not be avoided.

Given the dimensions of approximately 4 x 25 x 35 mm the **GM042535** battery from PowerStream Technologies can be used. The capacity of this battery is about 300 mAh. Consequently, an average consumption of 1.8 mA can be maintained during one week.

A fast calculation of the average power consumed by the microcontroller can be found in the formula 2. It is assumed that the microcontroller is used for 15 minutes per week in fast mode (80MHz clock), an additional 10% of the time in slow mode (4MHz clock) and the rest of the time in idle mode. The power consumption in each of this three states is given in the datasheet. [24]

$$(0.002*85)+(0.1*6)+(0.898*0.065)=0.82837\,mA \quad (2)$$

The theoretical result is an average power consumption of 0.83 mA by the microcontroller. Therefore, the rest of the components can consume an average of approximately 1 mA and still achieve the 1 week battery life.

The connection between the battery and the PCB is done using the same type of connectors than the ones used for the programming and explained in more detail in the section In Circuit Serial Programming (ICSP).

2.1.3 Tree axis accelerometer with gyroscope

A three axis accelerometer is required in this project to measure patient movements. Initially the device chosen for the PHDI was the ADXL346 also used in the former project, the HDH. However, this device does not provide gyroscopic functions whereas nowadays many devices do. That was the main reason to upgrade this device with the **MPU-6050** from InvenSense. [27]

The MPU-6050 is a small device that provides the three axis acceleration values in addition to the gyroscopic acceleration. It is accessible via the I2C bus using the address **1101000.**

As the schematic in Figure 33 shows, the typical application circuit extracted from the data sheet has been implemented. The connections between the accelerometer and the microcontroller are summarized in the next table.

Accelerometer + gyro.	Microcontroller
I2C bus	I2C1 peripheral

Table 3: Accelerometer + gyro. connections

2.1.4 Altimeter

An altimeter has been added to the design in order to have additional information to be able to determine when a fall event occurs. The same device as the one used in the previous project has been used this is the **BMP085** by Bosch.

The connections with the Micro-Processing Unit (MPU) are summarized in the next table. It uses the I2C1 bus with the address 1110111 for the data and two GPIOs for reseting the device and for reading the End Of Conversion signal from the altimeter. [28]

Altimeter	Microcontroller
I2C bus	I2C1 peripheral
ALT_$\overline{RST}$	RE0
ALT_EOC	RE1

Table 4: Altimeter connections

Adding a low noise linear regulator

Due to the high sensibility of the altimeter to noise and by recommendation of Michael Haller a spare linear regulator was added exclusively for the power supply of the altimeter. This can also be seen in the schematic in Figure 34.

2.1.5 ANT module

The first approach to the ANT feature was to add a new device called PAN1327. The greatest advantage of using this device instead of the one used in the previous project is the size. As shown in the pictures below the size of the PAN1327 is incredibly small (9 x 9.5 mm) in comparison with the ANTAP281MXIB (20 x 20 mm), the one used in the HDH.

In addition, the PAN1327 is also Bluetooth compatible which gives more freedom to our project.[29]

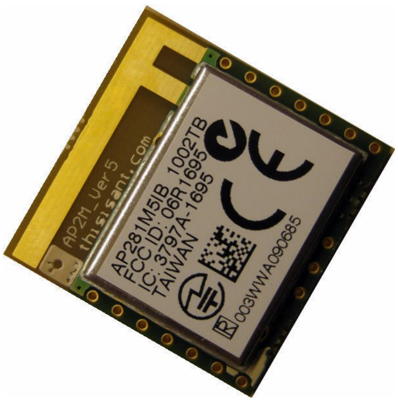


Figure 8: ANTAP281MXIB module

Figure 9: PAN1327 module

There were two problems to face with this new module: First, the firmware had to be redesigned from scratch and second, the availability of this module in Europe was very limited. Therefore the old ANTAP281MXIB module was finally used in this project as well.[30]

The final schematic can be found in the Schematics section, Figure 35. As seen in the schematic, two test points have been added to be able to debug the firmware via UART.

This module is connected using the UART-TTL interface, thus the PORTSEL pin is connected directly to ground. The speed is controlled with the BR1, BR2 and BR3 pins. This pins are connected to 3.3 volts, 3.3 volts and 0 volts respectively. Therefore the UART baud rate must be set in the microcontroller side to **50000 bps**.

In the next table there is a summary of the ANT module connections with the microcontroller.

ANT module	Microcontroller
UART bus with RTS	UART2 peripheral
ANT_SLEEP	RF0
ANT_SUSPEND	RD1
$\overline{\text{ANT\_RST}}$	RD0

Table 5: ANTAP281MXIB connections

2.1.6 RFID module

The RFID module chosen for this project was the smallest available in the market (at least that we know about), the M1-mini from Skyetek. It was used in a different project of this department, the Bluetooth thermometer. It's diameter is only 26 mm and will determine the width of the whole printed circuit board. The circuit implemented for this device is once again the one suggested in the data sheet for UART-TTL serial communication. The reason for choosing this UART-TTL communication protocol instead of the I2C or SPI is again simplify the task of the programmer, thus using the same protocol as in the thermometer. [31]

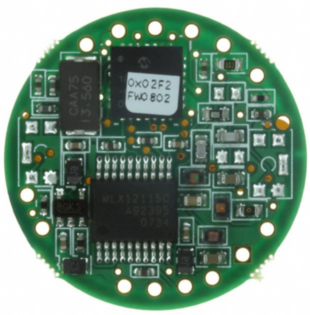


Figure 10: Skyetek's M1-mini RFID module.

The baud rate of the serial interface is software programmable and self adaptable. No hardware flow control is supported. The next table shows the available serial configurations.

4800 bps, N/8/1	+/- 0.3% error
4800 bps, N/8/1	+/- 0.3% error
4800 bps, N/8/1	+/- 0.3% error
4800 bps, N/8/1	+/- 0.3% error
4800 bps, N/8/1	+/- 0.3% error

Table 6: RFID module UART supported data rates

One of the drawbacks of this module is it's high power consumption. Even the sleep modes are not very power-saving. Therefore, the same solution as in the thermometer project has been implemented, this is adding a MOSFET transistor to switch the whole module power on and off.

Table 7 shows a brief description of the connections between the RFID module and the microcontroller.

RFID module	Microcontroller
UART bus	UART1 peripheral
RFID_RST	RE2
RFID_INT	RE3
$\overline{\text{RFID\_EN}}$	RB15

Table 7: RFID module connections

2.1.7 Micro SD Card

Due to the usage of WAV uncompressed audio files in this project for the speech feature, big amounts of non volatile memory are necessary. Therefore a micro SD card has been added to the design even though it is very big. On the other hand, it has the advantage of a very simple and known SPI interface that has been used in many previous projects.

This SPI drivers together with Chan's FatFs library for handling FAT32 or FAT16 file systems are a great tool for using standard SD cards.

The schematic is enclosed in Figure 36 and it is the same as the one used in th HDH. It is very important to be aware of the hardware connections of both the SPI bus with the microcontroller peripheral and the control lines of the SD Card reader. Above there is a chart summarizing this.

SD Card	Microcontroller
SPI bus	SPI2A peripheral
SDCARD_CD (Card Detect)	RB2[1]
SDCARD_CS (Chip Select)	RB3

Table 8: SD card connections

2.1.8 Audio output

The speech function is implemented with WAV audio files, this is the pure signal, sampled and quantified. Therefore, the easiest, cheapest and smaller way to play them is to use the Output Compare peripheral of the microcontroller in Pulse Width Modulator (PWM) mode in combination with a simple Low Pass Filter. [32]

A pulse width modulated signal is a square signal whose duty cycle is modulated according to an other signal called modulator signal.

The spectrum obtained applying the Fourier transform to a pulse width modulated signal is special because a DC component with an amplitude proportional to the duty cycle can be found in the modulated signal. Along with this DC component there will be sinusoids in the fundamental frequency (f) of the PWM signal as well as in the harmonics (2f, 3f, 4f,...). Therefore this PWM signal can be filtered with a proper low pass filter to obtain the modulator signal again getting rid of all this noise. [33]

A very simple LPF was designed mainly because of two reasons. On the one hand, the PHDI is supposed to play voice which frequency spectrum goes from a couple of Hertz to about four or five Hertz [34]. On the other hand the chosen fundamental frequency (f) for the pulse width generator will be high enough so that it goes out of the audible range of humans (more than 20 kHz). In addition, the common use amplifiers and speakers act as low pass filters as well, therefore there is no noise expected.

Figure 11 shows the theoretical frequency response of the designed filter. It is supposed to heavily attenuate everything above approximately 5 kHz.

1 Internal pull-up resistor has to be enabled

As seen in the data sheet of the microcontroller [24] the maximum current sourced by any I/O port is 25 mA which is insufficient to feed a speaker. Therefore an amplifier is mandatory. The selected amplifier is a 1 W mono channel audio amplifier called **TDA8551** from NXP. It comes in a small SO8 package. The final result can be found in the schematic in Figure 37. The connections to the microcontroller are summarized in the next table.

Amplifier	Microcontroller
VOICE_UP/DOWN	RE6
VOICE_SLEEP	RE7
VOICE_OUT	OC5

Table 9: Amplifier connections

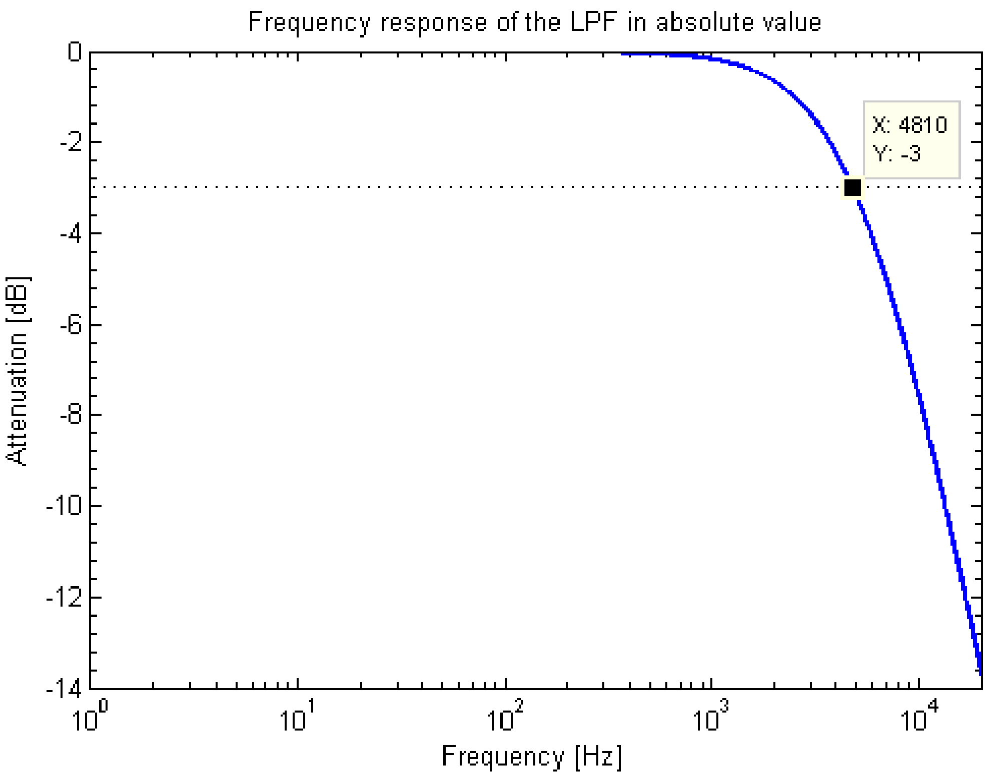


Figure 11: Frequency response of the LPF

2.1.9 In Circuit Serial Programming (ICSP)

The programming and debugging of the microcontroller firmware is done trough the ICSP connector and Microchip's ICD 3 device. Due to the size restrictions, a small connector from Molex has been used, the **53261 PCB Header** along with the **51021 Female plug** shown in the following two figures[2].

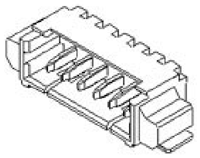


Figure 12: Molex PCB header

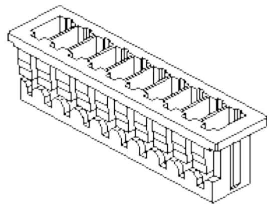


Figure 13: Molex female plug

This PCB Header has to be connected with the ICD 3 using it's RJ-12 plug, the following diagrams show how to connect them for proper function.

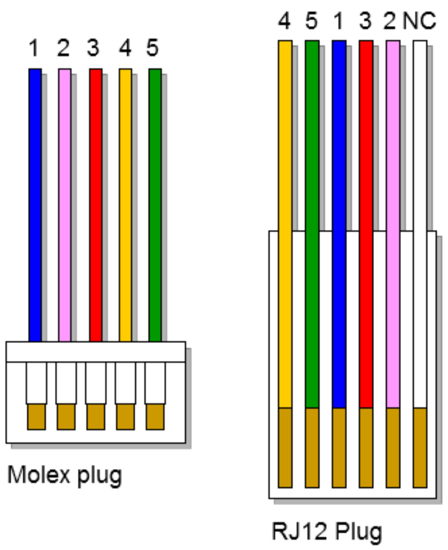


Figure 14: Molex plug – RJ12 plug pin mapping

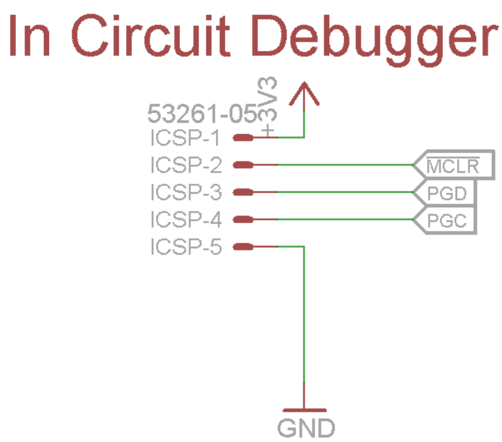


Figure 15: ICSP schematic

2 Note that figures 12 and 13 are for reference only. The actual number of pins is 5 on each.

2.1.10 User interface

Regarding to the user interface, a simple two LED with one button interface has been designed according to the requirements. The LEDs are connected with the microcontroller I/O pins listed below and the button is connected to a CN peripheral to be able to trigger an interrupt when pressed and therefore save power avoiding continuous polling.

UI	Microcontroller
LED1	RD7
LED2	RD6
SW	RD5/CN14

Table 10: UI connections

2.1.11 Shrinking it!

The final schematic comply with all the requirements but when trying to put all the components in the board it was too hard to get acceptable dimensions. Two components were found to be superfluous: The mini-USB connection and the spare thermometer integrated circuit.

The USB connector was expendable because all the data is going to be directly written in the SD-Card or transmitted via ANT. The only remaining usage of the USB connector was as a battery charger. In order to get rid of the USB connector an alternative way for charging had to be found. That is when the inductive charging came up. That is the perfect solution for two reasons, the whole circuit with the diode bridge, rectifier and capacitors is smaller than the mini-USB connector and the requisite of waterproofness is easier to fulfill without mechanical connections. The inductive charging solution is explained in detail in the next section.

About the spare thermometer, there were no need for it since the accelerometer, the altimeter and the battery gauge have built in thermometer that can be easily accessed.

2.1.12 Inductive battery charger

Inductive charging is based on the electromagnetic coupling of energy between two coils when the time varying flux of magnetic field generated by the first passes through the second one. [35]

External Charging Unit

There are many ways to generate this time varying electromagnetic field but the most common for this particular application is to use a primary coil in the charger whose current varies with time with a frequency high enough to propagate over the air but not excessively high so that electronic components in the bracelet can handle this signal (i.e. lower than their cut off frequency).

An example of an inductively chargeable embedded system can be found in an article [36] that has been taken as a reference for the design of the charger. In this article four MOSFET transistors are used to switch coil current forward and backward rapidly. This four transistor are controlled via two GPIOs of an ATMega microcontroller.

However, for reasons of simplicity and costs, this design will control the transistors via a clock signal generated by a 555 IC: *"The time machine"*. The output frequency is controlled using a RC network as shown in the schematic of the inductive charger on Figure 39. The values of the resistances and the capacitance can be easily determined using the data sheet. [37]

The best value for the frequency is to be determined empirically in the laboratory.

The transistors are also different from the ones used in the mentioned article. The TC4428A dual driver from Microchip is used because it has the four transistors in a single package. Moreover this IC has the transistors connected in pairs as needed for this application and one pair of transistors are driven trough an inverting gate reducing the number of input control signals to only one. [38]

Finally, the addition of an auto stand by circuit would be perfect to optimize the power consumed by the charger while not in use. A possible solution could be sampling the voltage level in the coil. When the second coil is coupled, the equivalent resistance of the primary coil would change and therefore the voltage seen on it's pins. A basic comparator circuit would do the job in addition to some kind of timer to test for the presence of the second coil from time to time. Although, in this case, the addition of a basic microcontroller should be a better solution. This last part has not been added to the design of the charger for this first prototype.

Internal Charging Unit

The coupled energy is rectified at the bracelet using a diode bridge followed by a capacitor to reduce the ripple and a linear regulator to ensure a good quality 5 volts input to the battery charger. This can be seen in the schematic on Figure 38.

Note that the linear regulator used in the bracelet allows a maximum current of 300 mA due to size restrictions. This limit has to be carefully considered. Consequently, the Vcc and the current limiting resistor of the inductive charger have to be chosen to avoid this limit.

2.1.13 Summary

Slave addresses of I2C devices

Device type	Device name	I2C Address
Accelerometer + gyroscope	MPU-6050	1101000
Battery monitor	LTC2942-1	1100100
Altimeter	BMP085	1110111

Table 11: Device addresses summary

Pin connections

Device	Device pins	Microcontroller pins
Power regulator	PMIC_CHRG	CN6/RB4
	PMIC_HPWR	CN7/RB5
Battery monitor	I2C bus	I2C1 peripheral
Accelerometer	I2C bus	I2C1 peripheral
Altimeter	I2C bus	I2C1 peripheral
	ALT_$\overline{\text{RST}}$	RE0
	ALT_EOC	RE1
ANT module	UART bus with RTS	UART2 peripheral
	ANT_SLEEP	RF0
	ANT_SUSPEND	RD1
	$\overline{\text{ANT\_RST}}$	RD0
RFID module	UART bus	UART1 peripheral
	RFID_RST	RE2
	RFID_INT	RE3
	$\overline{\text{RFID\_EN}}$	RB15

Device	Device pins	Microcontroller pins
SD Card	SPI bus	SPI2 peripheral
	SDCARD_CD (Card Detect)	RB2[3]
	SDCARD_CS (Chip Select)	RB3
Amplifier	VOICE_UP/DOWN	RE6
	VOICE_SLEEP	RE7
	VOICE_OUT	OC5
User Interface	LED1	RD7
	LED2	RD6
	SW	RD5/CN14

Table 12: Pin connections summary

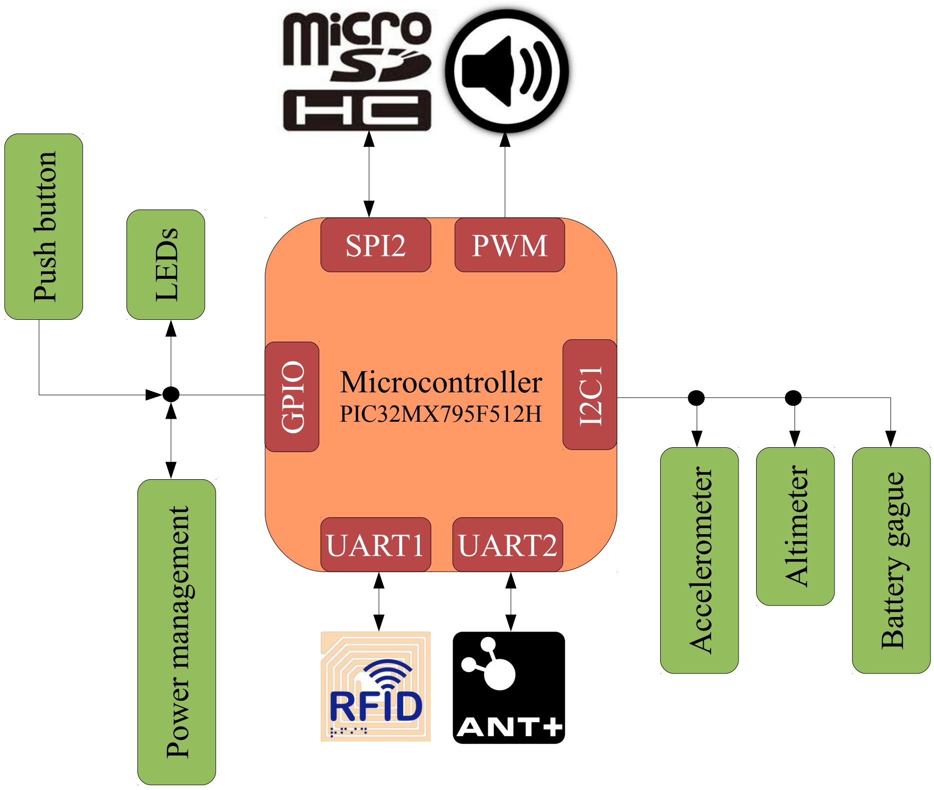


Figure 16: Information flow

3 Internal pull-up resistor has to be enabled.

2.2 Design of the printed circuit board

Once the schematic is completed, the design of the printed circuit board can start. This phase is divided in three tasks. Drawing the board dimensions, placing the components into it and finally routing all of them together. This three phases are explained in the following sub-chapters.

2.2.1 Dimensioning of the PCB

The dimensions of the board are drawn in the “dimensions” layer of the Eagle software. They are very important in this project since we want the PHDI to be comfortable and as small as an average bracelet. Sadly, the width of the PCB is determined by the RFID module chosen, the M1-Mini from Skietek which has a 26 mm diameter, hence the width of the bracelet. Many attempts to reduce the length of the bracelet have been done in the course of this thesis. However, the final length is longer than expected: 60 mm. The next figure shows the final board dimensions of the first prototype with a wider line.

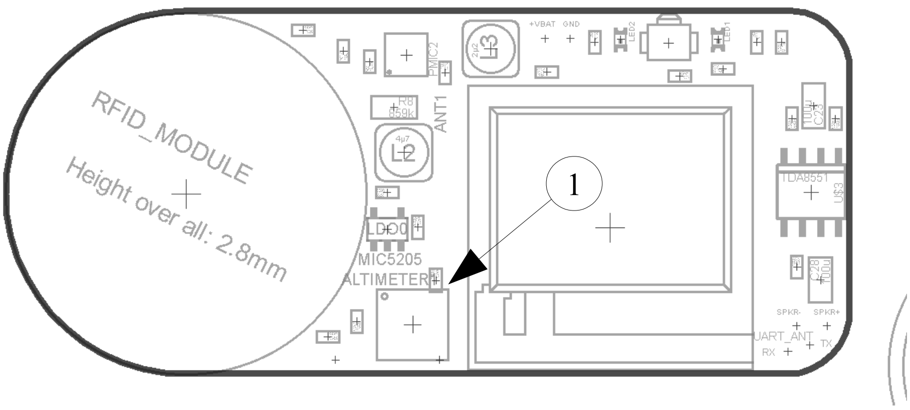


Figure 17: Board dimensions and components (top)

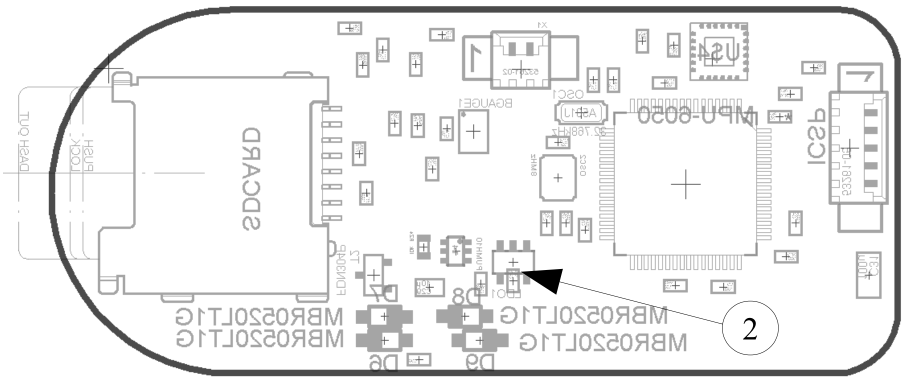


Figure 18: Board dimensions and components (back)

2.2.2 Component placing

There are many approaches on how to do the component placing. [23] The one that has been followed here states that the components should be placed in a way that makes the routing of signal lines easier. Therefore the first pin of every IC has a different orientation in the board.

Placing the RF modules and the SD card

The first components to be placed on the PCB were the RF modules and the SD-Card slot due to their bigger size. Many combinations have been tested and the optimum one appeared to be the one shown in Figure 17, where the RFID reader is placed on the left top layer and the SD-Card slot underneath it, on the back layer. The main reason to put the SD-Card slot under the RFID reader is because the RFID has many vias all along the perimeter making it difficult to fit components on it's other side of the board. With the proper rotation of the module, the SD-Card slot fits perfectly in between the RFID vias. Hence this was the chosen placement for this two components.

The ANT module was placed on the same side of the board as the RFID reader due to their similar height. The original thought was to place the higher components in the top layer of the board while keeping the back layer as thin as possible. Despite of this idea, the final result has reveal to be slightly different mainly because the SD-Card slot and the power and data connectors are higher than expected.

Placing the rest of the components

After placing the biggest modules, the next thing to place is the switching regulator. This is a very noisy device and it handles the highest amounts of power. Therefore route planning has to be done carefully and special attention has to be paid to route widths. The Figure 19 shows the final result. Remarkable details are highlighted with bright red. Among the important details one can find the massive via placed underneath the IC. It serves to two purposes: it provides a low impedance return path for the ground current and also works as a thermal via to cool the device down attaching the IC with the ground plane (more information about the ground plane in the section 2.2.3). Two very important components for the switching regulator are the inductors marked as L2 and L3. High speed switching current flows back and forth this two inductors and the switching regulator. Therefore, to reduce the radiated Electromagnetic Interference (EMI) the components have been placed as close as possible to the IC and in the same layer. [39]

The remaining components have been placed to optimize the space usage and the routing task. Basically filling the empty gaps in the board. The complete final design can be found in the figures 20 to 23 (one for each layer). And the final prototype in the Figure 26 y 27.

Some test vias can be found in the design for debugging purposes. Two of them are for the battery monitoring (+VAT and GND) and two more for the UART communication with the PC (named UART TX and RX).

Another set of vias are placed to connect external elements: the charging coil (unnamed vias) and the speaker (SPKR- and SPKR+).

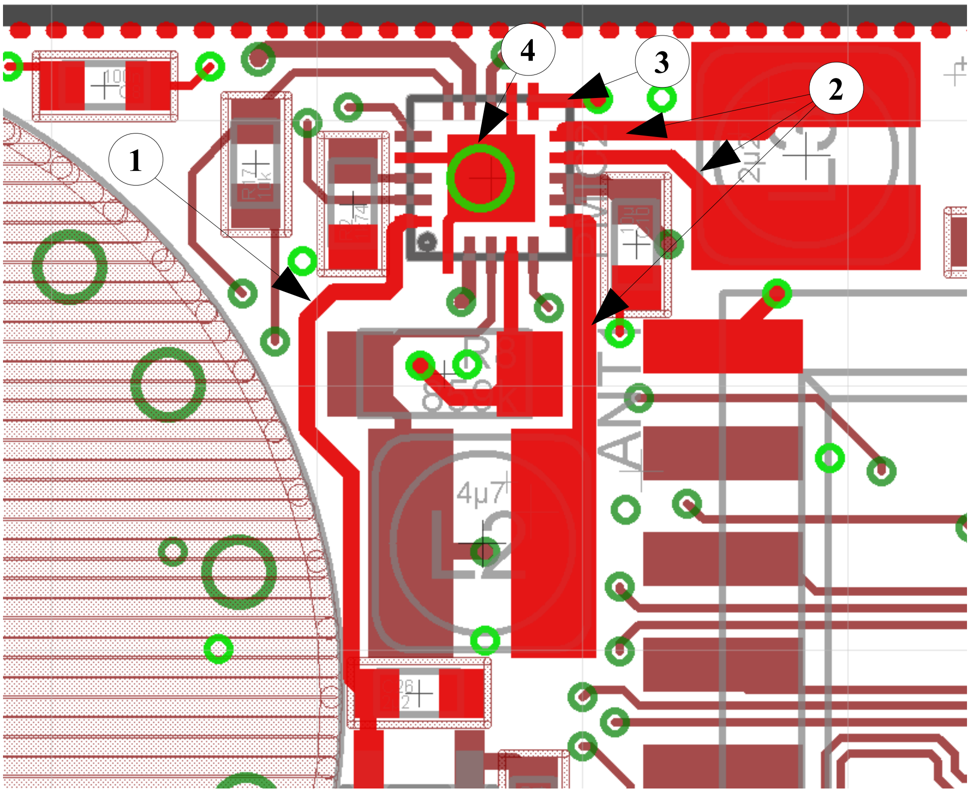


Figure 19: Switching regulator position and routing

2.2.3 Routing and number of layers

Routing all the signals through the board is a tedious work that requires patience and time. This process took place starting with the power lines and the clock lines in order to take special care to isolate them and dimension them correctly. Unfortunately this process could not be done properly with a cheaper two layer design that in theory “can achieve 95% of the effectiveness of a four-layer board by emulating what makes a four-layer board better” [39] mainly because of a lack of space to route wide enough power lines with the ground underneath them. Therefore a 4 layer design was made.

Upper layer

Power lines were designed first with a width of at least 10 mils[4]. As an example there is a long trace (1) that goes near the RFID module, from the lower part to the upper part of the Figure 19. This trace goes from the LDO (connected to the diode bridge also with a wide trace (3) and to the secondary coil) to the switching regulator and battery charger. It has to be wider since it drives the current to supply the whole circuit in addition to the current to charge the battery. This can represent up to 300 mA.

As told in the previous section, the inductances (2) exchange high-speed switching

4 One mil is a widely used unit in PCB design equal to a thousand of an inch.

current with the switching regulator. Therefore, their routes have been designed trying to comply with the 1:3 rule of width to length ratio where possible [39].

The next figure shows the final result for the upper layer, including ground fills where possible to shield routes as much as possible from cross talk.

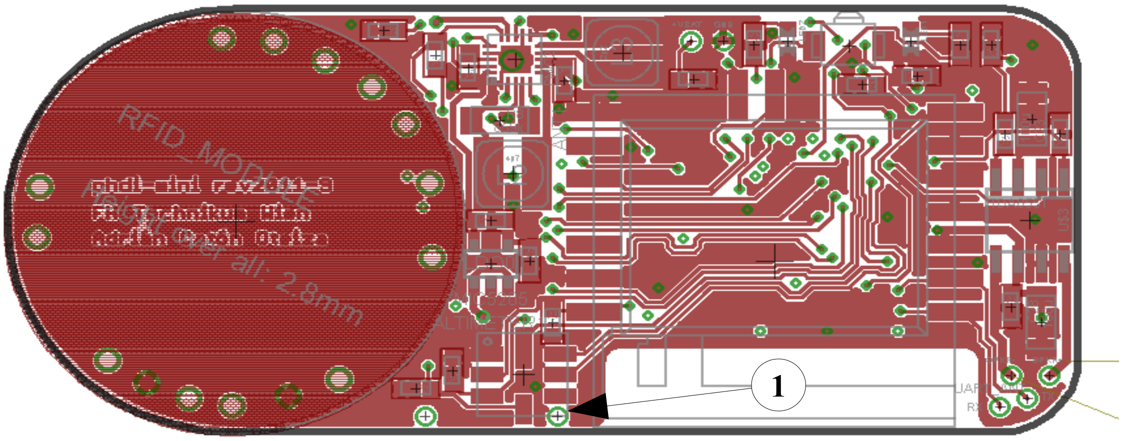


Figure 20: Top layer routing

It is remarkable the gap that has been performed where ANT's F antenna is located. This unfilled gap is present in all the layers as requested in the datasheet to optimize the radiated pattern of the ANT module.

Lower layer

The lower layer contains the microcontroller, the heart of the device along with two connectors for the battery and the debugging of the firmware, the accelerometer, the SD-Card slot, the battery gauge so it can measure the battery temperature and some auxiliary components. In both the upper and lower layers most of the empty spaces of the PCB are filled with ground planes to improve the isolation to crosstalk.

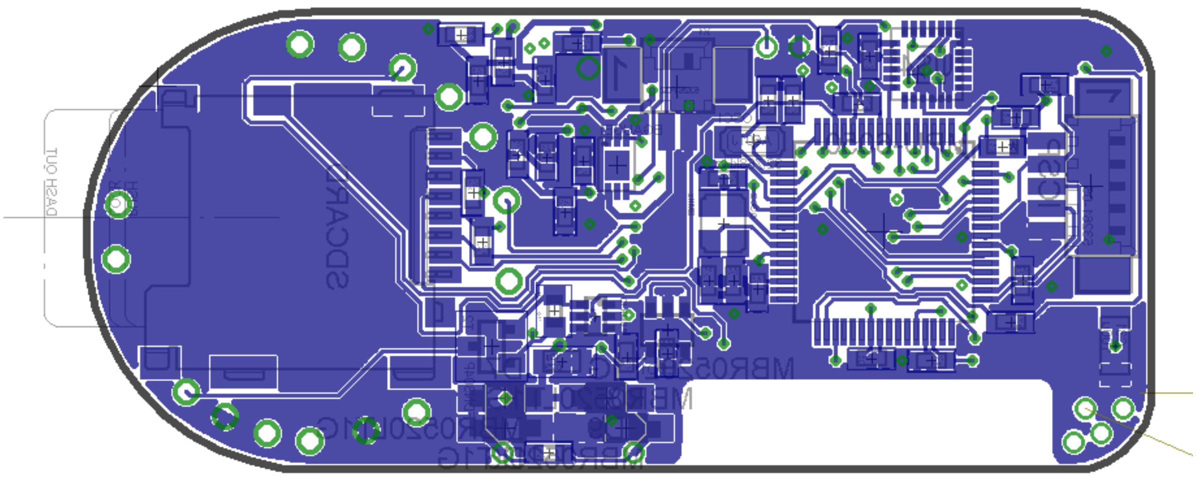


Figure 21: Back layer routing

Ground plane

A spare inner ground plane has been added to the design in order to provide a very low impedance return path for the current, thus avoid the unwanted EMI. As can be seen in Figure 22 no signals have been added to the ground plane to avoid big current loops. Also, when possible, vias have been carefully located to allow current to flow around them. [39]

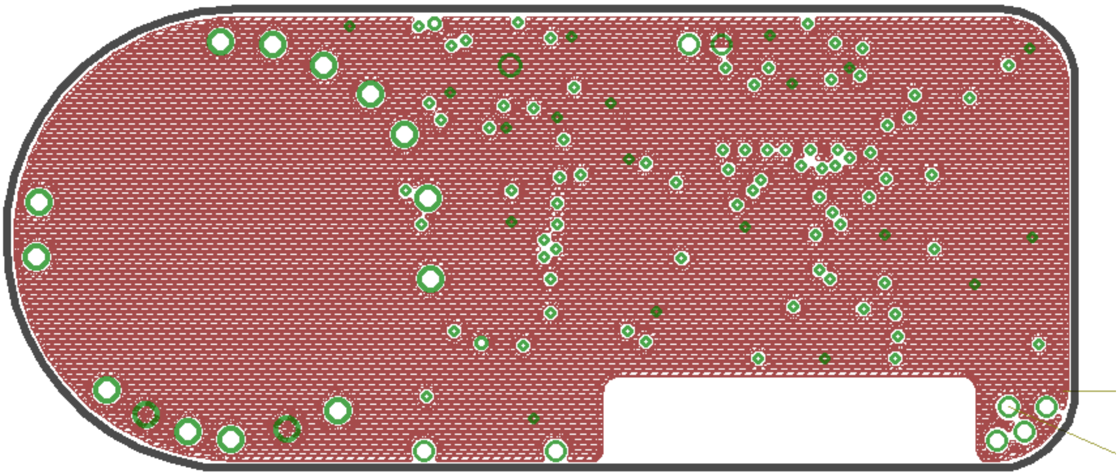


Figure 22: Ground plane layer

+3V3 plane

Another inner spare layer has been added for the 3.3 volts supply signal. This layer also contains some other signals when no other option was possible. Some effort has been done to route this signals by the borders of the board letting as much space as possible for the supplying currents to go as straight as possible.

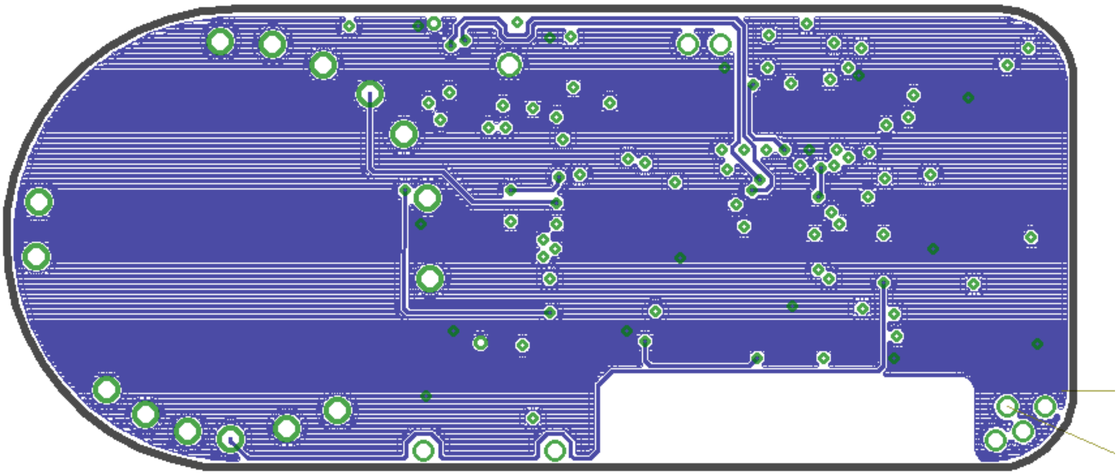


Figure 23: +3V3 layer

2.3 Estimated budget

The following chart is the list of the necessary hardware components required for building one device. The last column is the unitary price when ordering at least 1000 units.

One of the aims of this project is to have a production cost of less than $100 per unit (when ordering at least 1000), as can be seen in the following table, the estimated production cost is approximately a 50% higher than this objective. It can be adjusted taking out some of the unnecessary components. For example, the microcontroller can be switched with a cheaper one that meets the final necessities of this project.

Part	Value	Device	Package	Supplier	Reference	Unit price in € (ordering 1000)
ALTIMETER1	BMP085	BMP085	BMP085	digikey	828-1005-1-ND	3,470
ANT1	NTAP281MXII	ANTAP281MXIB	AP2	digikey	1094-1003-ND	11,892
BGAUGE1	LTC2942-1	LTC2942-1	FN6-2X3MN	farnell	1847652	4,400
C1	10nF	C0603	0603	rs	264-4595	0,004
C2	100n	C0603	0603	rs	624-2480	0,015
C3	100n	C0603	0603	rs	624-2480	0,015
C4	100n	C0603	0603	rs	624-2480	0,015
C5	100n	C0603	0603	rs	624-2480	0,015
C6	100n	C0603	0603	rs	624-2480	0,015
C7	0.1uF	C0603	0603	rs	624-2480	0,015
C8	100n	C0603	0603	rs	624-2480	0,015
C9	10µ	C0603	0603	rs	723-5540	0,092
C10	10µ	C0603	0603	rs	723-5540	0,092
C11	150p	C0603	0603	rs	191-973	0,164
C12	10p	C0603	0603	rs	698-3964	0,017
C13	10µ	C0603	0603	rs	723-5540	0,092
C14	10u	C0603	0603	rs	723-5540	0,092
C15	12p	C0603	0603	rs	147-437	0,010
C16	12p	C0603	0603	rs	147-437	0,010
C17	10µ	C0603	0603	rs	723-5540	0,092
C18	12p	C0603	0603	rs	147-437	0,010
C19	12p	C0603	0603	rs	147-437	0,010
C20	0.1u	C0603	0603	rs	624-2480	0,015
C21	10nF	C0603	0603	rs	264-4595	0,004
C22	0.1uF	C0603	0603	rs	624-2480	0,015
C23	100u	C1206	1206	rs	723-6515	0,458
C24	330n	C0603	0603	rs	698-3330	0,088
C25	100n	C0603	0603	rs	624-2480	0,015
C26	2µ2	C0603	0603	rs	723-5547	0,047
C27	100n	C0603	0603	rs	624-2480	0,015
C28	100u	C1206	1206	rs	723-6515	0,458
C29	10µ	C0805	0805	rs	723-5540	0,092

Part	Value	Device	Package	Supplier	Reference	Unit price in € (ordering 1000)
C30	C0603	603	#rcl	rs		
C31	100u	C1206	1206	rs	723-6515	0,458
C34	470p	C0603	0603	rs	264-4545	0,012
C37	10µ	C0603	0603	rs	723-5540	0,092
C43	100n	C0603	0603	rs	624-2480	0,015
C44	100n	C0603	0603	rs	624-2480	0,015
C45	10n	C0603	0603	rs	264-4595	0,004
C46	100n	C0603	0603	rs	624-2480	0,015
C47	100n	C0603	0603	rs	624-2480	0,015
D6	MBR0520LT1G	DIODE-SOD123	SOD123	rs	463-432	0,067
D7	MBR0520LT1G	DIODE-SOD123	SOD123	rs	463-432	0,067
D8	MBR0520LT1G	DIODE-SOD123	SOD123	rs	463-432	0,067
D9	MBR0520LT1G	DIODE-SOD123	SOD123	rs	463-432	0,067
ICSP	53261-05	53261-05	53261-05	rs	542-7270	0,804
	51021-05	1.25 mm fem. Conn.		rs	447-6580	0,190
		Connector adaptors		rs		
L2	4µ7	WE-TPC-S	WE-TPC-381	rs	573-785	0,592
L3	2µ2	WE-TPC-S	WE-TPC-381	rs	573-943	0,592
LDO0	P1802T-5002I/	MCP1802T-5002I/OT	SOT23-5L	rs	669-4957	0,494
LDO1	TPS731XX	SOT23-5L	#TI	rs	620-1445	1,100
LED1	LED	SMARTLED-TTW	led	rs	692-0975	0,041
LED2	LED	SMARTLED-TTW	led	rs	696-1314	0,038
OSC1	32.768kHz	FC-135	FC-135	rs	667-6108	0,732
OSC2	8MHz	C3E	C3E	rs	672-0646	1,280
PMIC2	LTC3558	LTC3558	FN20_3X3M	digikey	OUT OF STOCK	2,540
R2	1k74	R0603	0603	rs	708-7499	0,138
R3	4k7	R0603	0603	rs	619-7762	0,076
R4	100k	R0603	0603	rs	213-2531	0,006
R5	619k	R0603	0603	farnell	1171065RL	0,022
R6	200k	R0603	0603	rs	740-8849	0,013
R7	15k	R0603	0603	rs	213-2430	0,006
R8	859k	(866)!!	R1206	farnell	1501916RL	0,626
R9	649k	R0603	0603	farnell	1171067	0,022
R11	30	R0603	0603	rs	679-0204	0,006
R12	30	R0603	0603	rs	679-0204	0,006
R13	300	R0603	0603	rs	740-8852	0,013
R14	100k	R0603	0603	rs	213-2531	0,006
R15	100k	R0603	0603	rs	213-2531	0,006
R16	4k7	R0603	0603	rs	619-7762	0,076
R17	10k	R0603	0603	rs	740-8892	0,006

Part	Value	Device	Package	Supplier	Reference	Unit price in € (ordering 1000)
R24	10k	R-EU_R0603	0603	rs	740-8892	0,006
RFID_MODULE	M1MINI	M1MINI	M1MINI	digikey	753-1009-ND	35,000
SDCARD	SD_MICRO	SD_MICRO	MICRO_SD	Farnell	136-6700	2,130
SP1	AL201P	AL201P	AL201P	rs	619-8456	1,300
SW3	B3U-3000P	B3U-3000P	B3U	rs	419-861	0,526
T2	FDN304P	FDN304P	SOT23	rs	671-0403	0,232
T4	PUMH10	PUMH10	)T363_PHILI	rs	518-3835	0,036
U$2	;32MX795F512	PIC32MX795F512H-	FP64_10X10	rs	687-9344	6,030
U$3	TDA8551	TDA8551	SO08	rs	725-9085	0,636
U$4	MPU-6050	MPU-6050	QFP-N24(R	Spark Fun	SEN-10937	12,160
X1	53261-02	53261-02	53261-02	rs	542-7090	0,564
	51021-02	1.25 mm fem. Conn.		rs	279-9156	0,106
		Connector adaptors				
PCB						1,470
Battery						4,000
Speaker					716-6453	2,360
SD-Card						6,000
					TOTAL	**104,640**

Table 13: Estimated budget

2.4 Firmware

Some basic firmware has been developed to test the first prototype. Other functions have been implemented by Elsa Moriones as part of her thesis and have been also used to test this prototype. [40]

2.4.1 Hardware Profile

A header file called hardwareProfile.h is provided to abstract the source code from the specific hardware and make it more intuitive and human readable. Here most of the labels used in the schematics section are defined to substitute the specific port of the GPIO peripheral. In addition, with every definition, a macro is defined as well to set the direction of the GPIO peripheral. The whole file is presented in the following lines.

```
#ifndef __HARDWAREPROFILE_H
#define __HARDWAREPROFILE_H

#ifndef __PIC32MX__
#error "This example of fatfs is only for PIC32 at present!"
#endif

// define the Processor instruction frequency
#define FOSC   80000000L

// define battery capacity
#define Q_BAT   100 //100mAh battery

//SD-CARD/////////////////////////////////////////
// define the interface that the SD card is using
#define MEDIASD_IF_SPI2

// define the chip select pin function
#define _SetChipSelect(on)          {\
             mPORTBSetPinsDigitalOut(BIT_3);\
        if(on)\
             mPORTBClearBits(BIT_3);\
        else\
             mPORTBSetBits(BIT_3);\
                                    }

// Card detect - Read Only
#define SD_CD                 (mPORTBReadBits(BIT_2))
#define SDSetCDDirection()  (mPORTBSetPinsDigitalIn(BIT_2))
#define CD_PullUp_Enable() CNPUEbits.CNPUE4=1;
// Write enable - Read Only
#define SD_WE              0 //(mPORTGReadBits(BIT_1))
#define SDSetWEDirection()   //(mPORTGSetPinsDigitalIn(BIT_1))
```

```
// SPI configuration defines
#define FATFS_SPI_START_CFG_1      (PRI_PRESCAL_64_1 | SEC_PRESCAL_8_1 | MASTER_ENABLE_ON | SPI_CKE_ON | SPI_SMP_ON)
#define FATFS_SPI_START_CFG_2      (SPI_ENABLE)


//RFID//////////////////////////////////////////
#define RFID_RST LATEbits.LATE2  //
#define RFID_EN  LATBbits.LATB15 //RFID enable (activate low)
#define RFIDSetENDirection()  (mPORTBSetPinsDigitalOut(BIT_15))
#define RFIDSetRSTDirection() (mPORTESetPinsDigitalOut(BIT_2))

//ANT///////////////////////////////////////////
#define ANT_RTS            PORTBbits.RB8
#define ANT_RST          LATDbits.LATD0
#define ANT_SLEEP        LATFbits.LATF0
#define ANT_SUSPEND      LATDbits.LATD1
#define ANTSetRTSDirection() (mPORTBSetPinsDigitalIn(BIT_8))
#define ANTSetRSTDirection() (mPORTDSetPinsDigitalOut(BIT_0))
#define ANTSetSLEEPDirection() (mPORTFSetPinsDigitalOut(BIT_0))
#define ANTSetSUSPENDDirection() (mPORTDSetPinsDigitalOut(BIT_1))

//AUDIO AMPLIFIER////////////////////////////////
#define VOICE_SLEEP     LATEbits.LATE7
#define VOICE_VOLUME    LATEbits.LATE6
#define VOICESetSLEEPDirection() (mPORTESetPinsDigitalOut(BIT_7))
#define VOICESetVOLUMEDirectionOut() (mPORTESetPinsDigitalOut(BIT_6))
#define VOICESetVOLUMEDirectionIn() (mPORTESetPinsDigitalIn(BIT_6))

//BATTERY////////////////////////////////////////
#define PMIC_CHRG        PORTBbits.RB4
#define PMIC_HPWR        LATBbits.LATB5
#define PMICSetCHRGDirection() (mPORTBSetPinsDigitalIn(BIT_4))
#define PMICSetHPWRDirection() (mPORTBSetPinsDigitalOut(BIT_5))

//ALTIMETER//////////////////////////////////////
#define ALT_RST     LATEbits.LATE0
#define ALT_EOC     PORTEbits.RE1
#define ALTSetRSTDirection() (mPORTESetPinsDigitalOut(BIT_1))
```

```
#define ALTSetEOCDirection() (mPORTESetPinsDigitalIn(BIT_0))

//LEDS//////////////////////////////////////////////////////
#define LEDGreen      LATDbits.LATD7
#define LEDRed        LATDbits.LATD6
#define LEDGreenSetDirection() (mPORTDSetPinsDigitalOut(BIT_7))
#define LEDRedSetDirection() (mPORTDSetPinsDigitalOut(BIT_6))

#endif // __HARDWAREPROFILE_H
```

2.4.2 Battery

Battery charging status and gauge is monitored using two IC as seen in the section Battery protection, power regulation and monitorization. One of the ICs is a power regulator and a battery charger, it is responsible of charging the battery among other things. The charging status can be read from the microcontroller using the PMIC_CHRG data line defined in hardwareProfile.h. The second IC is a battery gauge. It is responsible for counting the battery charge, voltage and IC's temperature. The battery.c file communicates with these ICs and can be called periodically to read current battery charge and to advise via the red led and an ANT message of a low battery event. Also it is responsible of resetting the coulomb charge counter of the battery gauge IC when the battery is completely charged and to advise accordingly via the red led and an ANT message.

A structure is defined in battery.h to store all the battery status and configuration data. This structure is called **battery** and has the following fields:

```
struct battery
{
  int          capacity,
               voltage,
               temperature,
               prescaler,
               last_ant_sent=0;

  enum
  {
     CE_ENABLED = 0,
     CE_DISABLED = 1
  } charge_enable;

  enum
  {
     PS_LOW = 0,     // 100mA
     PS_HIGH = 1     // 500mA
  } power_select;

```

```
    enum
    {
        CHG_EN = 0,
        CHG_DIS = 1
    } charging;

    enum
    {
        CHG_COMPLETE = 0,
        CHG_RUNNING = 1
    } charge_complete;

    enum
    {
        CHG_FAULT = 0,
        CHG_NORMAL = 1
    } charge_fault;

    enum
    {
        USB_POWER_OK = 0,
        USB_DISCONNECTED = 1
    } usb_power;
};
```

Some other minor type declarations are made in hattery.h and some status definitions as well, to make the source code more human readable.

The first important piece of code found in battery.c is called upon initialization of the device and sets the configuration registers of the battery gauge. Depending on hardwareProfile.h Q_BAT constant this function sets the gauge counter prescaller. It also turns the battery gauge on in case it was in sleep mode. As can be seen, the function `I2C1LDWriteByte()` is used here to communicate with the IC. This function is inherited from the HDH project, and it is included in the i2c.c and i2c.h files.

```
void batteryInit(void) {
    //Inits the gauge counter
    unsigned char bat_data[10];
    char m;

    //Calculate the prescaler for the connected battery
    m = ceil((float)128*Q_BAT/65536 / 0.085);
    bat.prescaler=0;
    while (m > 0) {
        m = m >> 1;
        bat.prescaler++;
    }
    //Turn on the battery gauge and sets the prescaller
    I2C1LDWriteByte(BAT_ADDRESS, BAT_CONTROL, (0b00000000 |
```

```
    (bat.prescaler << 3)));
}
```

The second function to be called after the initialization of the battery sets the battery charger IC to provide low charging current to the battery (i. e. 100 mA). This is performed through the GPIO pin called PMIC_CHRG in the schematics and defined in the hardwareProfile.h. The code is provided below.

```
void batterySetLowPowerMode(void){
    //Sets the LTC3558 to supply low current to the battery
  (100mA)
    PMICSetHPWRDirection();
    PMIC_HPWR=0;
    bat.power_select=0;
}
```

Three functions are provided to access structure voltage, temperature and capacity fields. This three functions take care of reading this values from the battery gauge appropriately. They are called `batteryGetVoltage()`, `batteryGetTemperature()` and `batteryGetCapacity()`. The three of them return integer values.

The last and most important function is `batteryCheckStatus()`. It is called periodically to check the battery status and charger presence. In case of a low battery event, a charger plugged event or a charger unplugged event it reports to the user via ANT and the red LED. The source code can be found below.

```
void batteryCheckStatus(void){
    //This function checks the current charge and the charging
    //status and reports via red led and ANT messages
    int totalMeasurableCapacity =
                        0.085/128*(0xFFFF<<bat.prescaler);
    int foo = bat.charging;
    //ANT battery status message
    unsigned char ant_message[8] = {0x01,0,0,0,0,0,0,0};

    if (foo!=batteryGetChargerStatus()){
        if (bat.charging == NOT_CHARGING){
            //Not charging
            //Turn the charging red led off
            LEDRed=0;
            //Advise with an ANT message of charge ¿completed?
            if (bat.last_ant_sent!=BAT_ANT_CHARGE_COMPLETE){
                antInit();
                ant_message[1]=BAT_ANT_CHARGE_COMPLETE;//Low
                                              battery message
                if (!antSendAckData(ant_message))
                    bat.last_ant_sent=BAT_ANT_CHARGE_COMPLETE;
                    //If msg sent success. dont send it again
                antStop();
            }
```

```

        } else {
            //Charging
            //Advise with a still red led
            LEDRed=1;
            //Advise with an ANT message
            if (bat.last_ant_sent!=BAT_ANT_CHARGING){
                antInit();
                ant_message[1]=BAT_ANT_CHARGING;//Low battery message
                if (!antSendAckData(ant_message))
                    bat.last_ant_sent=BAT_ANT_CHARGING;//If msg sent success. dont send it again
                antEnd();
            }
        }
    }

    if (batteryGetCapacity()<totalMeasurableCapacity-Q_BAT+10) {
        //battery too low
        if (bat.charging == NOT_CHARGING) {
            //Advise with a blinking red led
            LEDRed^=1;
            //Advise with an ANT message
            if (bat.last_ant_sent!=BAT_ANT_LOW){
                antInit();
                ant_message[1]=BAT_ANT_LOW;//Low battery message
                if (!antSendAckData(ant_message))
                    bat.last_ant_sent=BAT_ANT_LOW;//If msg sent success. dont send it again
                antEnd();
            }
        }
    }
}
```

2.4.3 Power Modes

Due to the high importance of power consumption in this project, a special spare file in the source code has been implemented for dealing with it. This file is called **powerModes.c.** There, one can find five important functions that are accessible through the header file. This functions are listed below:

```
void powerSetSlowMode(void);
void powerSetUltraSlowMode(void);
void powerSetFastMode(void);
void powerSetSleepMode(void);
```

```
int powerGetSystemClock(void);
int powerGetPBClock(void);
```

The default power mode of operation is the **SlowMode**. This mode can be enabled using the first function: `powerSetSlowMode()`. This function sets the internal PLL dividers and multipliers to generate a 4 MHz system clock and a 4 MHz peripheral clock signals. This speed of operation reduces the power consumed for the entire bracelet to about 20 mA.

The process to modify the clock frequency is explained below with the source code.

```
1  //Slow mode is the default mode of operation
2  void powerSetSlowMode(void){
3      //Change the clock source meanwhile to the LPRC.
4      powerSetUltraSlowMode();
5      //Disable ints. code above is a critical section
6      unsigned int int_status=INTDisableInterrupts();
7      //Unlock the clock selection
8      SYSKEY = 0x0;
9      SYSKEY = 0xAA996655;
10     SYSKEY = 0x556699AA;
11     OSCCONbits.PLLMULT = 0b001;//MUL_16
12     OSCCONbits.PLLODIV = 0b100;//DIV_16
13     OSCCONbits.NOSC=0b011;//Return to the primary osc. w/ pll
14     OSCCONbits.OSWEN = 1;//Start the clock switch
15     SYSKEY = 0x0;//Lock the clock selection again
16     INTRestoreInterrupts(int_status); //Restore interrupts
17
18     sysclk=4000000;
19     pbclk=sysclk / (1 << mOSCGetPBDIV());
20
21     //Recalculate everything that depends on SYSCLK and PBCLK
22     // Set up Timer 1 to generate interrupts at 1kHz
23     // this is used to drive the underlying disk io system
24     OpenCoreTimer(sysclk / 2000);
25     // set up the timer interrupt with a priority of 1
26     mConfigIntCoreTimer(CT_INT_ON | CT_INT_PRIOR_1);
27 }
```

As seen in the line 4 above, the function `powerSetUltraSlowMode()` is being used as part of this process. This function switches the clock source from the external quartz crystal with PLL to a different clock source (the internal LPRC) as required in the data sheet. [41]

Also, the data sheet states that two special keys have to be written to the SYSKEY register prior to modify the PLL values, in order to unlock the OSCCON register as done in the lines 8 to 10.

The clock switching process is also explained in the source code below:

```
//Ultra slow mode is used when switching between modes
void powerSetUltraSlowMode(void){
```

```
    unsigned int int_status=INTDisableInterrupts();
    //Unlock the clock selection
    SYSKEY = 0x0;
    SYSKEY = 0xAA996655;
    SYSKEY = 0x556699AA;
    OSCCONbits.NOSC=0b101;//Set the new osc as LPRC
    OSCCONbits.OSWEN=1;
    SYSKEY = 0x0;//Lock the clock selection again
    INTRestoreInterrupts(int_status);

    sysclk=31250;
    pbclk=sysclk / (1 << mOSCGetPBDIV());

    //Recalculate everything that depends on SYSCLK and PBCLK
    // Set up Timer 1 to generate interrupts at 1kHz
    // this is used to drive the underlying disk io system
    OpenCoreTimer(sysclk / 2000);
    // set up the timer interrupt with a priority of 1
    mConfigIntCoreTimer(CT_INT_ON | CT_INT_PRIOR_1);
}
```

The third power mode is the **FastMode**. This fast mode sets the system clock to 80 MHz and the peripheral clock to 80 MHz as well. It is used only for playing audio files. This is necessary for audio playback since the usual sample rate in audio files ranges from some kilo samples per second up to 44 kSamples/s. Therefore, high speed processing is needed to fetch the data from the SD card to the buffers. The power consumption is very high in this mode of operation ranging from 80 mA to 120 mA when the audio amplifier is on. This mode can be enabled using the function `powerSetFastMode()`. This function is implemented the same way as the `powerSetSlowMode()` changing the PLLMULT and the PLLODIV values to 20 and 1 respectively.

The last power saving mode is the **SleepMode**. The function below sets the microcontroller to sleep, stopping the core. Only selected peripherals are allowed to stay awake among them, the Watch Dog Timer and the Change Notice peripherals. Prior to entering sleep mode, the WDT is enabled. If no other interruption is triggered while the bracelet is in sleep mode, the WDT will trigger one after the timeout set in the configuration words expires. Currently every 8.3 seconds the bracelet is waked-up from sleep mode.

```
//The device is supposed to be most of the time sleeping
void powerSetSleepMode(){
    //Set the WDT
    WDTCONbits.ON=1;
    SYSKEY = 0x0;
    SYSKEY = 0xAA996655;//Unlock the OSCCON
    SYSKEY = 0x556699AA;
    OSCCONSET=0x10;//Sleep mode
    SYSKEY = 0x0;//Lock the OSCCON back
```

```
    asm volatile ( "wait" );//Go to the specified mode
    WDTCONbits.ON=0;//After sleeping disable the WDT
}
```

Finally, the last two functions provide indirect, read only access to the variables while keeping basic abstraction from the data structures used in the powerModes.c file. `powerGetSystemClock()` returns the current system clock frequency and `powerGetPBClock()` returns the current peripheral clock frequency.

2.4.4 Altimeter

Most of the source code used for the altimeter comes from the predecessor project, the HDH, since both have been designed using the same altimeter model. Therefore, the code is considered to be known and no much comments about it are going to be done besides the found issues and solutions implemented.

The first theory about the wrong pressure and temperature readings was the size of the variables used for the calculations. This variables' sizes are different between the microcontroller families that have been used in both projects. In this thesis a 32 bits microcontroller was used and therefore the int type is 32 bits long and the short type is 16 bits long. The many calibrations that take place to obtain the pressure and temperature values use the shift operator and maybe some overflows were taking place. Therefore all the data types are reviewed and corrected to meet the specifications given in the datasheet [28]

After some time spent in different tests the second issue found was that the ALT_RST and ALT_EOC labels were wrongly defined one mixed with the other and vice-verse. Therefore, the End Of Conversion while was not taking place and that's the main reason why strange values were read.

Even though the read values were more consistent after fixing the last issue, they were far from correct. It was giving a 90.0 meters read where a 160.0 read should have been given. Thus a calibration of the device was done using the international barometric formula (3) with a known altitude and the read pressure. This is the pressure at sea level.

$$p_0 = \frac{p}{\left(\frac{altitude}{44330}\right)^{5.255}} \tag{3}$$

The last thing to be aware of is that the altitude is returned in decimeters and also that a variability of +/- 0.4 meters can be easily read without moving the device making it extremely difficult to use the altitude values for a reliable fall detection.

2.4.5 Real Time Clock and Calendar (RTCC)

The addition of a RTCC has been considered from both hardware and software points of view and some code has been implemented to make it work. However, due to time restrictions and lack of implementation of some vital parts on the AT the RTCC is not completely functional.

2.5 Test suite

To check if the prototype is functional and behaves as expected, test cases have been designed according to the standard guidelines. [42]

This test cases can be found in the section Test suite in the Appendixes. The results obtained from six different testers are also included in the Test results section of this document.

3 Results

A summary of the achievements is done in this section with images of the final prototype built in different stages of development and a description of the final functionality implemented.

3.1 Printed board

After three months of design, the PCB of the first prototype was manufactured and looked like the Figure 24 and 25. Unfortunately the final dimensions were bigger than desirable but still an improvement of more than 100% has been achieved in comparison with the former project, the HDH.

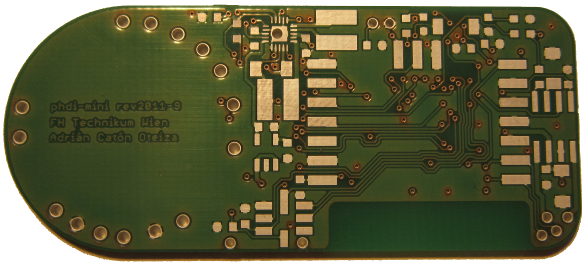


Figure 24: Front side of the first PCB

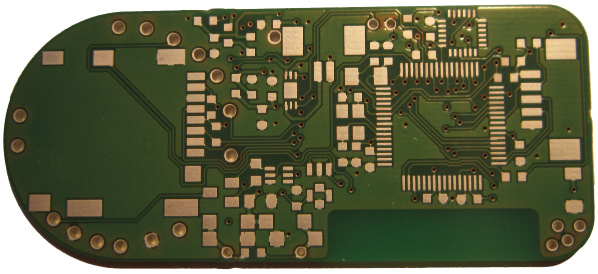


Figure 25: Back side of the first PCB

In addition, short before receiving the PCB the list of necessary components was written and ordered. A lot of care was taken to order the adequate components and the right amount of them.

3.2 First prototype

The population of the first prototype was primarily made by Mr. Michael Haller obtaining as a result the first working PHDI device. Two pictures of it are included below.

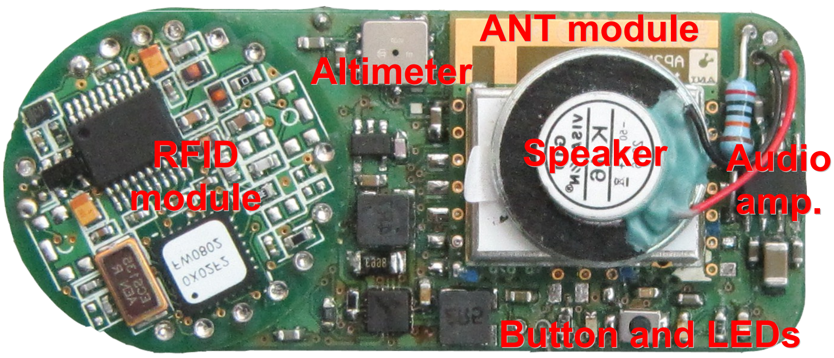


Figure 26: Populated front side

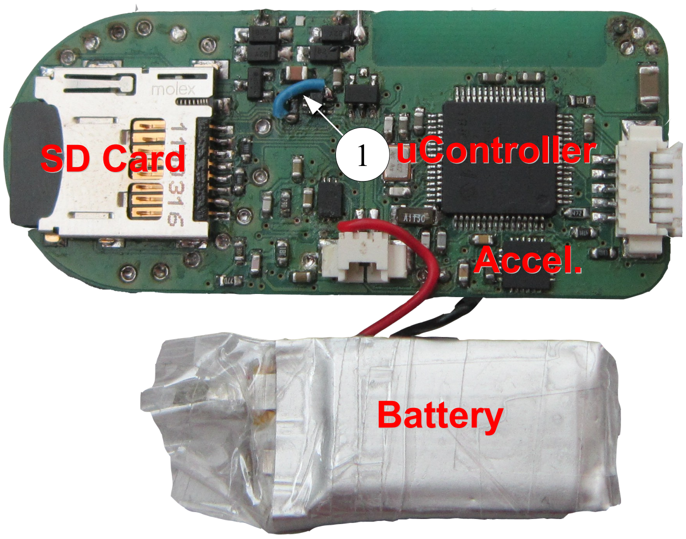


Figure 27: Populated back side with 100mAh battery

3.3 eHealth 2012 presented prototype

In the next picture, the fully functional prototype that was presented in the eHealth 2012 conference in Vienna is shown. The case has been designed by Mr. Gregor Wiktorin, member of the PHDI project, and built in the FH Technikum Wien.

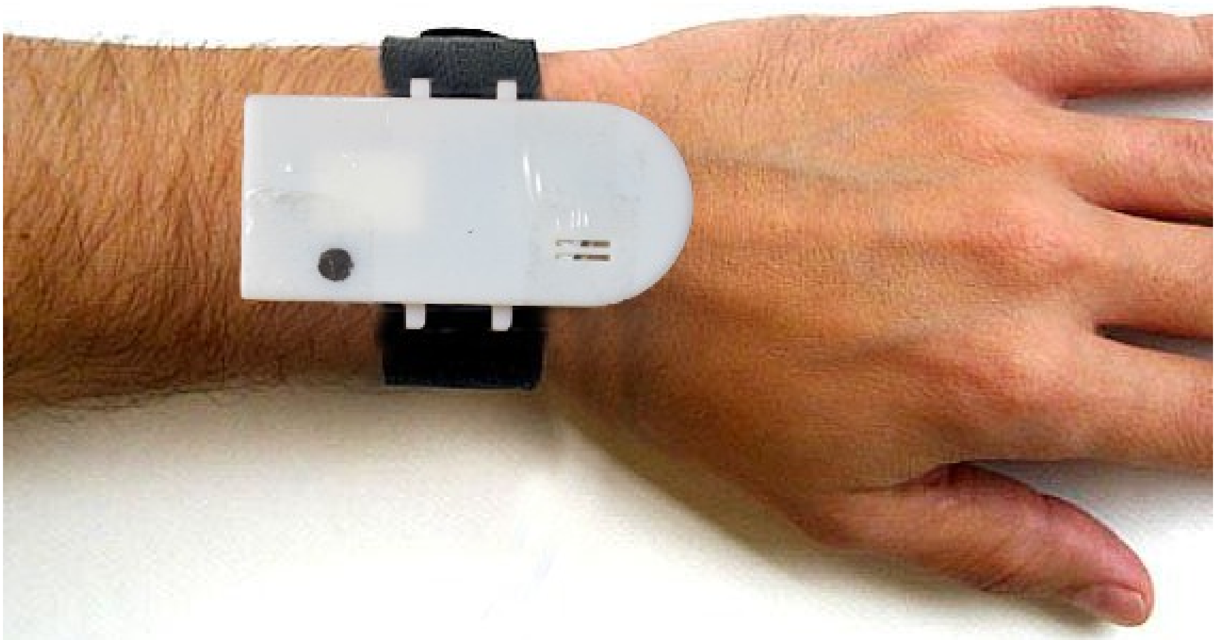


Figure 28: PHDI prototype presented on the eHealth 2012 Wien conference

A summary of the goals that have been achieved can be found in the section Achievements.

3.4 Battery

Recalling the previous section Battery protection, power regulation and monitorization, the key to achieve one week battery duration is to have a power consumption of less than 1.8 mA. Assuming that a 300 mAh battery is used.

The real current consumed by the PHDI is about 20 mA even though software has been carefully designed to reduce the power consumed. Actually, most of the time the device is held sleeping and the auxiliary modules such as the RFID or the ANT are also switched off.

This fact makes it impossible to achieve one week of battery life with a 300 mAh battery. In addition, the PHDI prototype built has a smaller 100 mAh battery because it was the one available in the biomedical laboratory.

To sum up, the prototype lasts for about 7 hours with the current settings.

3.5 Test results

The test suite presented in the previous chapter: Test suite, has been performed by two different people related to the project with the results shown in the next table.

Tester	**New bracelet test**	**Send tag id test**	**Read tag AT, not reachable test**	**Press the button but not read the tag test**
Adrián Catón	PASSED	PASSED	PASSED	PASSED
Elsa Moriones	PASSED	PASSED	PASSED	PASSED

Table 14: Test suite results summary.

As can be seen, most of the basic functions work correctly when tested by different people giving a sense of how good the goals have been achieved.

4 Discussion

In this section there is an explanation of the found issues and a road map for future versions of the device based on the experience acquired with this first prototype. This final part is supposed to be the starting point for any future development of the PHDI. However some of the found issues have already been fixed in this thesis as stated in sections above. Notice that only design issues are presented here, other issues such as manufacture problems are not commented.

4.1 Found issues

The first major problem was found when trying to use the RFID reader. When the microcontroller initiated communication with it using the standard UART interface no response was received. Using the oscilloscope was easy to find that the module was not receiving the necessary power in it's Vdd pin. After some hours of unsuccessful debugging the problem was found in one of the transistors used to power it up. In particular, the BJT transistor that takes part in the RFID powering circuit was wrongly designed and therefore one of it's pins was not connected properly. The origin of this issue dates back to the design of the Bluetooth thermometer from where the RFID schematic was taken. This issue has been solved in the first prototype using an aerial wire as can be seen in Figure 27, number 1. In addition, it has also been fixed in the 2012-0 revision of the schematics and board.

Once the problem described above was solved communication between the microcontroller and the RFID module worked. However, when a RFID read was requested to the module, nothing happened. Apparently the supply voltage dropped below the necessary 3 volts when the lecture was requested. This is due to the high power consumed by the RFID module and the inability to supply it of the ICD 3. Therefore, for this moment on a battery or an external power supply was needed to debug the source code.

An other major problem was found while populating the PCB. There were two capacitors overlapping with other components (see numbers 1 and 2 of Figure 17 and Figure 18). The Design Rules Checker did not complain about it because the components that the capacitors were overlapping with did not have any restricted zone drown, therefore they were transparent to eagle DRC. As seen in the Figure 26, one of the capacitors has been placed in a different spot maintaining it's functionality (1) and the other one has been omitted apparently without consequences (2). Both of them are now fixed in the revision 2012-0.

In addition, an other misplacement issue was found concerning the altimeter and one of the coil vias. This caused some problems during laboratory trials due to the obstruction of one of the sides of the via but should not be a problem in commercial prototypes where no wire unsoldering has to be done. Anyway, it has been easily fixed in the last revision of the board moving the altimeter some mm closer to the RFID. This design error can be seen in Figure 20, number 1.

During the firmware debugging process an other issue was found relating the RFID reading distance since the lectures were only possible when the RFID tag was really close to the module. After some research trough the datasheet and the design, the only reason for it's malfunction was the ground plane located underneath the module. As seen in the antennas related bibliography [25] any metal near an antenna changes considerably it's radiation pattern. In this particular case the antenna is almost short circuited and therefore low power is radiated. Even though nothing is said in the data sheet, the sales representative of Skyetek confirmed that this is most probably the cause of the short reach of the module. This issue has also been fixed in the 2012-0 revision of the schematics and board. For the first prototype a simple yet elegant solution was implemented consisting in moving the RFID module 2 mm away from the PCB as can be seen in the next picture.

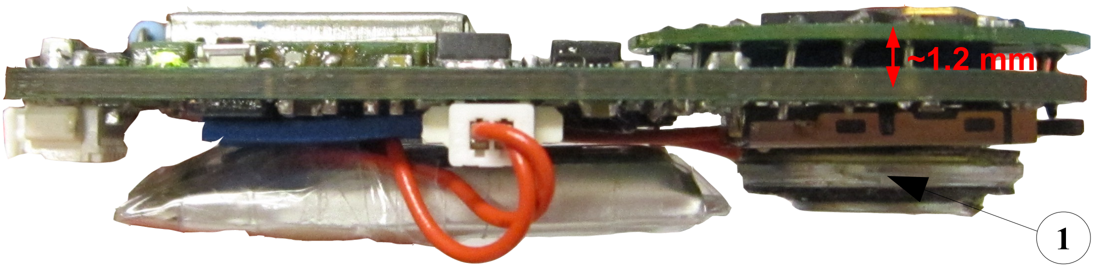


Figure 29: Prototype profile.

With this solution, the RFID tag can be read from up to about 1 cm and the maximum height of the bracelet remains almost unaltered.

Another problem came up when the fall detection application was being developed and no accelerometer interruption was available. Therefore, manual continuous polling had to be done in order to detect the various accelerometer interrupts. This problem has been solved in the “phdi-mini rev 2012-0.brd” and “.sch”, introducing a route between the interruption pin of the accelerometer and microcontroller CN11.

Even with the interruptions enabled, the chosen accelerometer doesn't meet all the necessities for this project, according to what Elsa Moriones said. The reason is the lack of a proper threshold register for motion detection. This threshold only admits values up to 250 mg independently of which scale is selected. It would be recommendable to switch back to the ADX345 even though it has no gyroscope.

The components shopping list is also an important thing to handle carefully because wrong components ordered are translated in prototype manufacturing delays and cost overruns. In this case, some components were wrongly ordered as can be seen in Table 13, highlighted in red. However, most of them were easily replaced with components available in the laboratory. The only exception was the speaker SP1, which was replaced temporally with a bigger one and finally with the speaker with RS reference number 716-6453 that is shown in Figure 29, number 1.

For comfort reasons, the side button has been changed with a up button with reference number 419-867.

A minor issue was found in the thermal via underneath the switching power regulator (Figure 19, 4). It is not in the middle of the exposed pad and therefore this component had to be manually placed, otherwise the superficial tension of the liquidized soldering paste in the via would have moved it to the left. This has been solved in the last version of the board.

Also in the exposed pads of the battery gauge and the accelerometer. The data sheet of the battery gauge says that some vias to a thermal plane should be placed underneath it or a bigger plane attached to it in order to dissipate heat and obtain better battery charge estimations. This was not done because as the data sheet specifies, it has to be isolated from any other signals and that ruled out the ground plane and made it very difficult to do in the given space.

Finally, there might be problems with the routes and vias located underneath the accelerometer with gyroscope. As the data sheet states, routes and vias underneath it can affect the proper function of the gyroscope if they are not shielded. Therefore, the inclusion of a ground plane underneath it is recommended for future versions as well as the modification of all routes underneath the gyroscope.

4.2 Future improvements

Most of the issues shown above are necessary improvements, most of which have been already added to the design. However, in this section other observations are made in order to improve the PHDI prototype and could be useful in a future.

The main aspect that has to be improved is the size. Therefore, here some ideas on how to do that are explained:

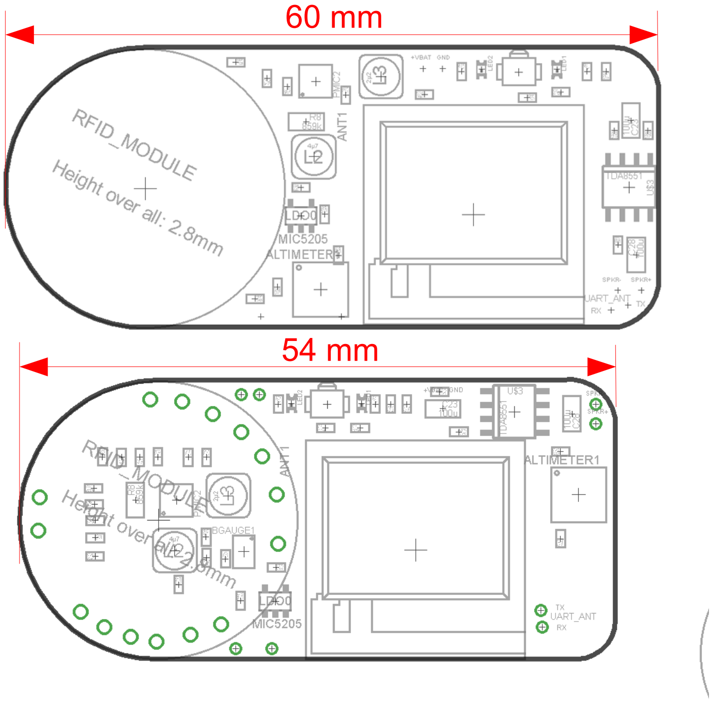


Figure 30: Comparison of the mini version and the hypothetical micro version of the PHDI

First, when the RFID module was separated from the PCB to increase it's read range a huge area of unused PCB appeared underneath it. This area could be used to place important components such as the power regulators and monitoring devices or even the microcontroller. In the figure above, a fast draft of how could the prototype be using this space is shown.

Other ways to improve the size would be changing the ANT module, for example with the PAN1327 shown in Figure 9 or any other smaller ANT device. Or getting rid of some modules like the altimeter which are not essential.

The prototype can also be thinner if the battery connector and the ICSP connector are omitted or replaced with something smaller.

In addition, the SD-Card slot is big and expensive. Therefore, it should be omitted in a theoretical commercial version of the device. The SD-Card could be soldered to the PCB saving one or two millimeters thick and a Euro or two.

The audio amplifier can also be switched with a smaller version like a TSSOP package. For example the LM4889 from National Semiconductors should fit. In addition it has less external components making it even better for this application.

Secondly, the battery duration has to be extended. The one week battery duration could be achieved using a 2400 mAh battery or finding out which device consumes that much power and replacing it with a lower consuming one.

The last thing that could be improved is the inductive charging. A step-up converter instead of the 5 volts linear regulator would probably make the coupling easier. Also, about the external inductive charger unit, an auto-stand-by system would be nice to reduce power consumption when no bracelet is being charged.

4.3 Achievements

In the following lines a summary of the achievements can be found.

- ✔ ANT and RFID communication protocols capable.
- ✔ No screen, only two LEDs and one button.
- ✔ Speech capabilities.
- ✔ Sensor for movement
- ✔ Small size.
- • Manufacturing cost per unit lower than $100. When producing 1000 devices.

  Final cost is about $140 per device, with the current design.
- ✗ Sensor for vital parameters.

  No pulse sensor, nor accurate temperature sensor is available.

Optional goals to be achieved are:

- • Waterproofness

  Electrically, the prototype is ready for inductive charging but the inductive charger has not been completely designed within the time frame of this thesis.
- • Fall detection algorithm

  Some work has been done in that field by Elsa Moriones [40]
- ✗ One week of battery duration.

  Currently a 100mAh battery is being used. It lasts about 6 hours of normal function. The power consumption has to be lowered from 15mA to 2mA and the battery capacity has to be increased to at least 400mAh
- ✗ Wake-up detection algorithm

  No wake up detection has been implemented.

Legend:

✔Fully achieved goal

• Partially achieved goal

✗ Not achieved goal

Bibliography

[1] European Commission, Eurostat, and Directorate-General for Employment, Social Affairs and Inclusion, *Demography Report 2010*. 2011.

[2] European Commission and Eurostat, *Active ageing and solidarity between generations*. 2011.

[3] R. Tarricone and A. D. Tsouros, *Home care in Europe: the solid facts*. WHO Regional Office Europe, 2009.

[4] “In Europe, care for the elderly is being transformed - The New York Times.” [Online]. Available: http://www.nytimes.com/2007/04/13/business/worldbusiness/13iht-wbelder.2.5273274.html?_r=1&pagewanted=all. [Accessed: 13-Apr-2012].

[5] K. Dracup, D. K. Moser, M. Eisenberg, H. Meischke, A. A. Alonzo, and A. Braslow, “Causes of delay in seeking treatment for heart attack symptoms,” *Social Science & Medicine*, vol. 40, no. 3, pp. 379–392, Feb. 1995.

[6] D. E. R. Warburton, C. W. Nicol, and S. S. D. Bredin, “Health Benefits of Physical Activity: The Evidence,” *CMAJ*, vol. 174, no. 6, pp. 801–809, Mar. 2006.

[7] G. J. Balady, B. Chaitman, D. Driscoll, C. Foster, E. Froelicher, N. Gordon, R. Pate, J. Rippe, and T. Bazzarre, “Recommendations for Cardiovascular Screening, Staffing, and Emergency Policies at Health/Fitness Facilities,” *Circulation*, vol. 97, no. 22, pp. 2283–2293, Jun. 1998.

[8] “This is ANT, the Wireless Sensor Network Solution.” [Online]. Available: http://www.thisisant.com/pages/technology/what-is-ant. [Accessed: 23-Apr-2012].

[9] M. Bitterman, “Development of a Data Model for a wrist wearable Health Data Hub,” FH Technikum Wien.

[10] “Nike+ FuelBand.” [Online]. Available: http://www.nike.com/fuelband/. [Accessed: 10-Apr-2012].

[11]“WakeMate–Wake up fresh; sleep smarter - WakeMate.” [Online]. Available: http://www.wakemate.com/. [Accessed: 10-Apr-2012].

[12] “aXbo sleephase alarmclock.” [Online]. Available: http://www.axbo.com/axbo/cms/leichter_aufwachen.aspx. [Accessed: 10-Apr-2012].

[13] “SleepTracker.” [Online]. Available: http://www.sleeptracker.com/. [Accessed: 10-Apr-2012].

[14] “The Wellness Monitor - Hammacher Schlemmer.” [Online]. Available: http://www.hammacher.com/Product/81511?source=EMZ00212&cm_ven=WC&cm_cat=20120103_EMZ002&cm_pla=BYR&cm_ite=LearnMore. [Accessed: 10-Apr-2012].

[15] “UP by Jawbone.” [Online]. Available: http://jawbone.com/up. [Accessed: 10-Apr-2012].

[16] “Fall Detection System | Devices, Automatic Fall Detection, Alerts, Systems, Pendants.” [Online]. Available: http://www.falldetectionsystems.com/. [Accessed: 10-Apr-2012].

[17] “BrickHouse Alert | Fall Detection Medical Alarm.” [Online]. Available: http://www.brickhousealert.com/. [Accessed: 10-Apr-2012].

[18] F. Sposaro and G. Tyson, “iFall: an Android application for fall monitoring and response,” *Conf Proc IEEE Eng Med Biol Soc*, vol. 2009, pp. 6119–6122, 2009.

[19] “Lifeline with AutoAlert automatically calls for help if a fall is detected.” [Online]. Available: http://www.lifelinesys.com/content/lifeline-products/auto-alert. [Accessed: 10-Apr-2012].

[20] R. S. Sangwan, R. G. Qiu, and D. Jessen, “Using RFID tags for tracking patients, charts and medical equipment within an integrated health delivery network,” in *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, 2005, pp. 1070 – 1074.

[21] “About Health Level Seven International.” [Online]. Available: http://www.hl7.org/about/index.cfm?ref=nav. [Accessed: 27-Apr-2012].

[22] “Hardware Design Guidelines.” [Online]. Available: http://www.doityourselfchristmas.com/wiki/index.php?title=Hardware_Design_Guidelines.

[23] J. Catsoulis, *Designing Embedded Hardware*, Second ed. O’Reilly Media, 2005.

[24] “PIC32MX3XX/4XX Family,” Microchip Technology Inc., U.S.A., Data Sheet.

[25] A. C. Aznar, *Antenas*. Alfaomega Grupo Editor, 2000.

[26] “LTC3558,” Linear Technology Corporation, U.S.A., Data Sheet.

[27] “MPU-6000/6050,” InvenSense, U.S.A., Data Sheet, Nov. 2011.

[28] “BMP085,” Bosch Sensortec GmbH, Germany, Data Sheet.

[29] “Electronic Components : RF Modules : Bluetooth : PAN1327/1317 from Panasonic.” [Online]. Available: http://www.panasonic.com/industrial/electronic-components/rf-modules/bluetooth/pan1327-1317.aspx. [Accessed: 10-Apr-2012].

[30] “This is ANT, the Wireless Sensor Network Solution.” [Online]. Available: http://www.thisisant.com/pages/products/ap2-module. [Accessed: 10-Apr-2012].

[31] “SkyeModule M1-mini,” SkyeTek Inc., Reference Guide.

[32] L. D. Jasio, *Programming 32-bit Microcontrollers in C: Exploring the PIC32*. Newnes, 2008.

[33] A. Palacherla, “Using PWM to Generate Analog Output,” Microchip Technology Inc., 1997.

[34] “Asymmetric digital subscriber line (ADSL) transceivers,” ITU, Recommendation G.992.1.

[35] R. K. Wangsness, *Electromagnetic Fields*, 2nd ed. Wiley, 1986.

[36] J. Wu, K. Kanukurthy, and D. Andersen, “Continuous, Inductively-Charged Power Supply for Portable Embedded Applications,” *Circuit Cellar Magazine*, Aug. 2006.

[37] “LM555 Timer,” Texas Instruments Incorporated, Data Sheet.

[38] “1.5A Dual High-Speed Power MOSFET Drivers,” Microchip Technology Inc., U.S.A., Data Sheet.

[39] “PCB Design Guidelines For Reduced EMI,” Texas Instruments Incorporated, Nov. 1999.

[40] E. Moriones, “Design and implementation of the operating software for the hardware prototype for Personal Health Data Identification (PHDI),” Fachhoschule Technikum Wien, Vienna.

[41] “PIC32 Family Reference Manual, Sect. 10 Power-Saving Modes,” Microchip Technology Inc., U.S.A., Reference Manual.

[42] J. D. Cem Kaner, “What Is a Good Test Case?,” *Proceedings of Florida Institute of Technology Department of Computer Sciences STAR East*, 2003.

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
ANSI	American National Standards Institute
CAD	Computer-Aided Design
CPU	Central Processing Unit
DRC	Design Rules Checker
DRC	Design Rules Check
EMI	Electromagnetic Interference
EOC	End Of Conversion (usually when converting from analog to digital)
ERC	Electrical Rules Checker
GDP	Gross Domestic Product
GP	General Practitioner
GPIO	General Purpose Input Output
HDH	Health Data Hub
HL7	Health Level Seven
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICD	In Circuit Debugger
ISO	International Organization for Standardization
LED	Light Emitting Diode
LPF	Low Pass Filter
LPRC	Low Power RC (oscillator)
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
OSI	Open Systems Interconnection
PAN	Personal Area Network
PCB	Printed Circuit Board
PHD	Personal Health Device
PHDI	Personal Health Data Identification
PWM	Pulse Width Modulation
QFN	Quad-Flat No-leads (package)
RISC	Reduced Instruction Set Computer
RTCC	Real Time Clock and Calendar
SPI	Serial Peripheral Interface
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver Transmitter
UI	User Interface
USB	Universal Serial Bus
WDT	Watch Dog Timer

Schematics

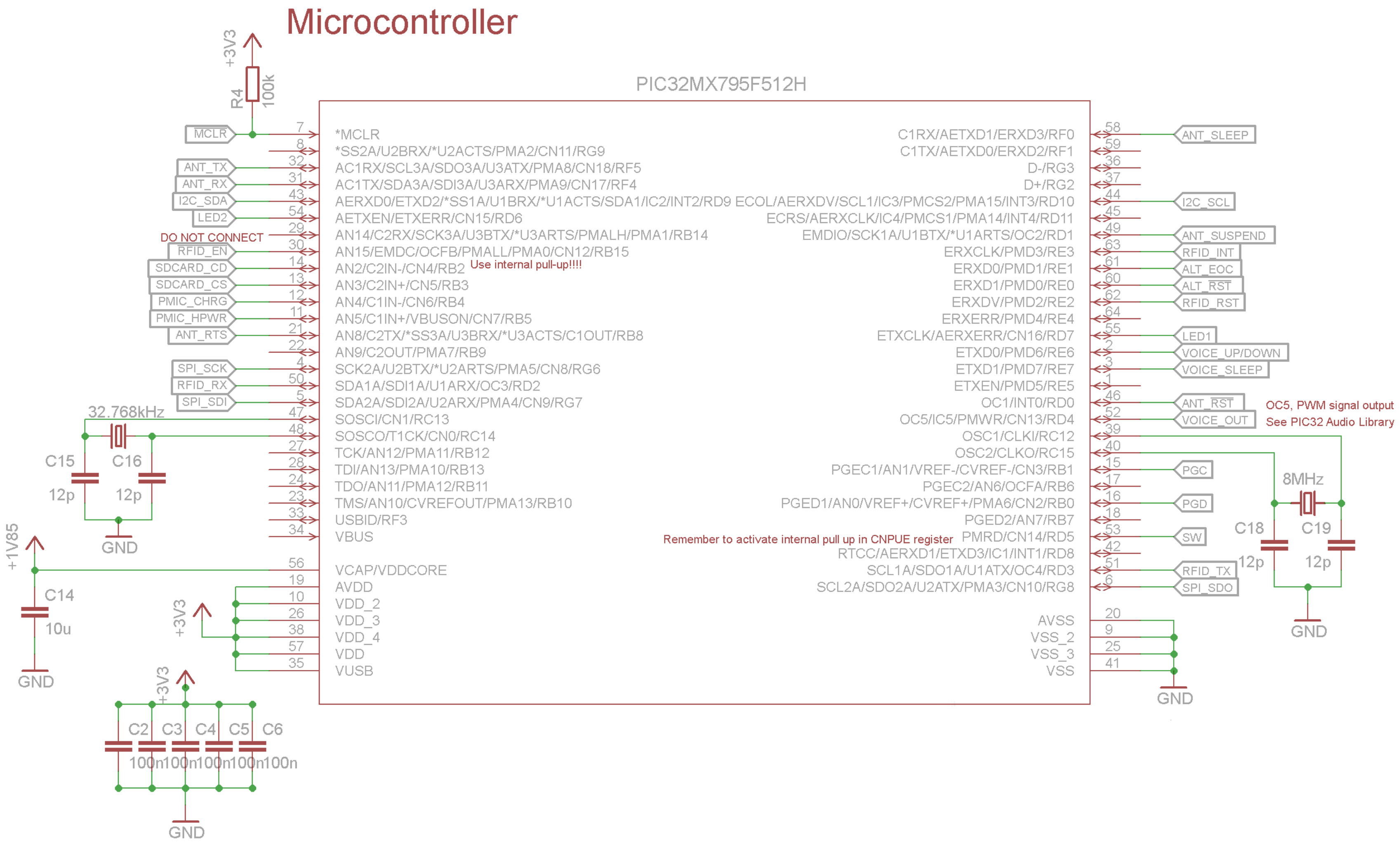


Figure 31: Microcontroller connections

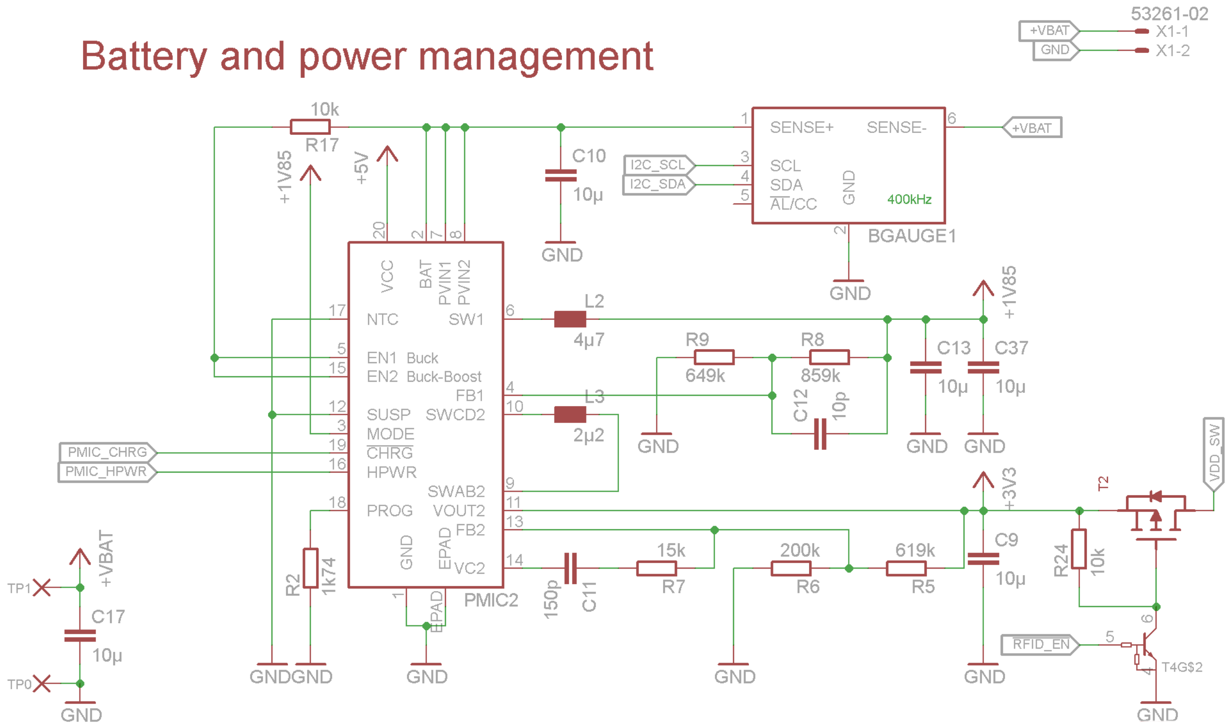


Figure 32: Battery protection-power management and monitoring schematics

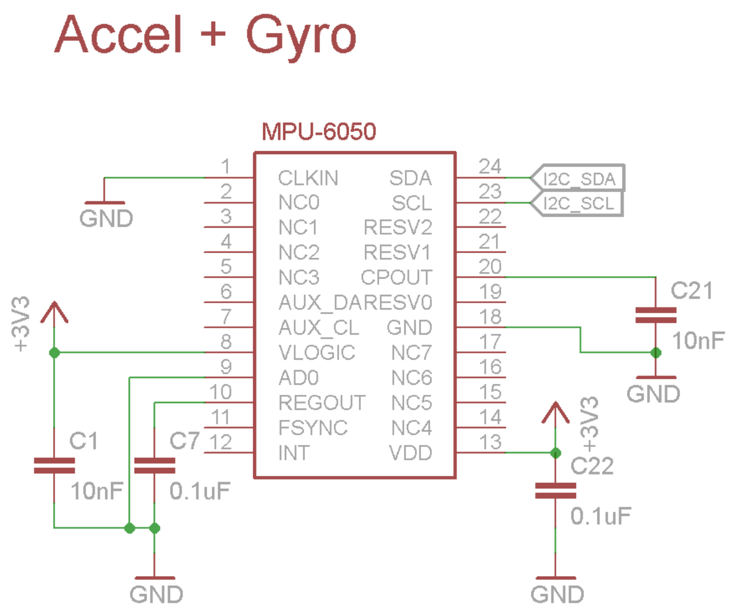


Figure 33: Accelerometer with gyroscope schematics

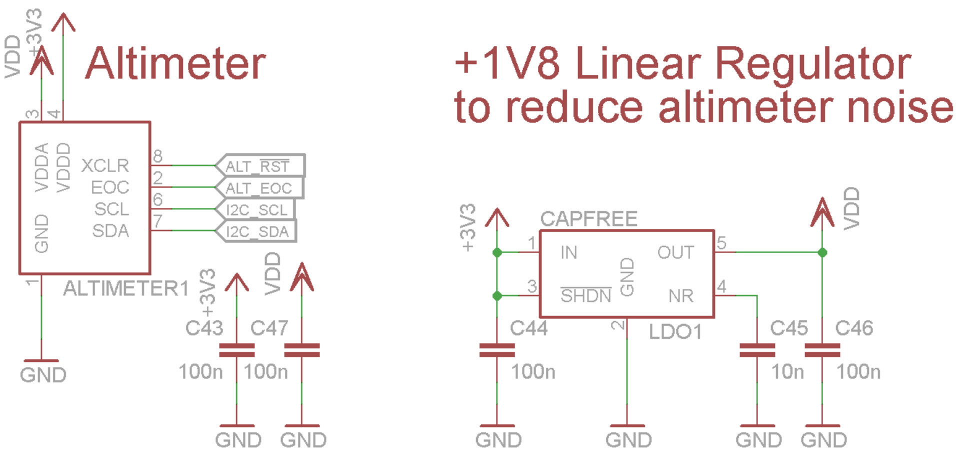


Figure 34: Altimeter schematics

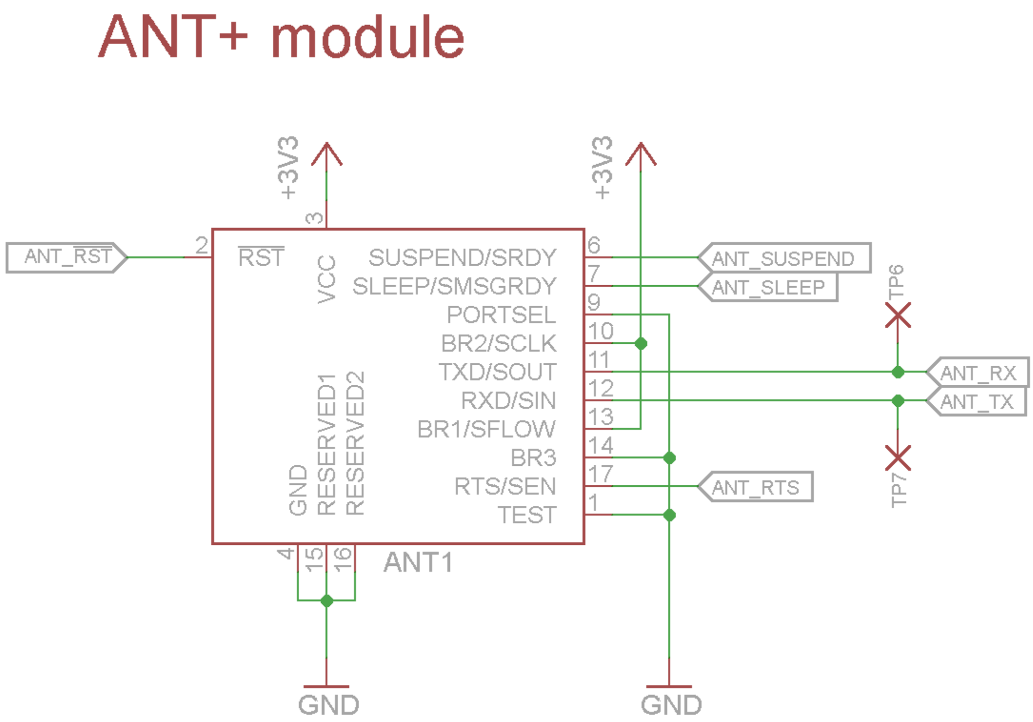


Figure 35: ANT module schematics

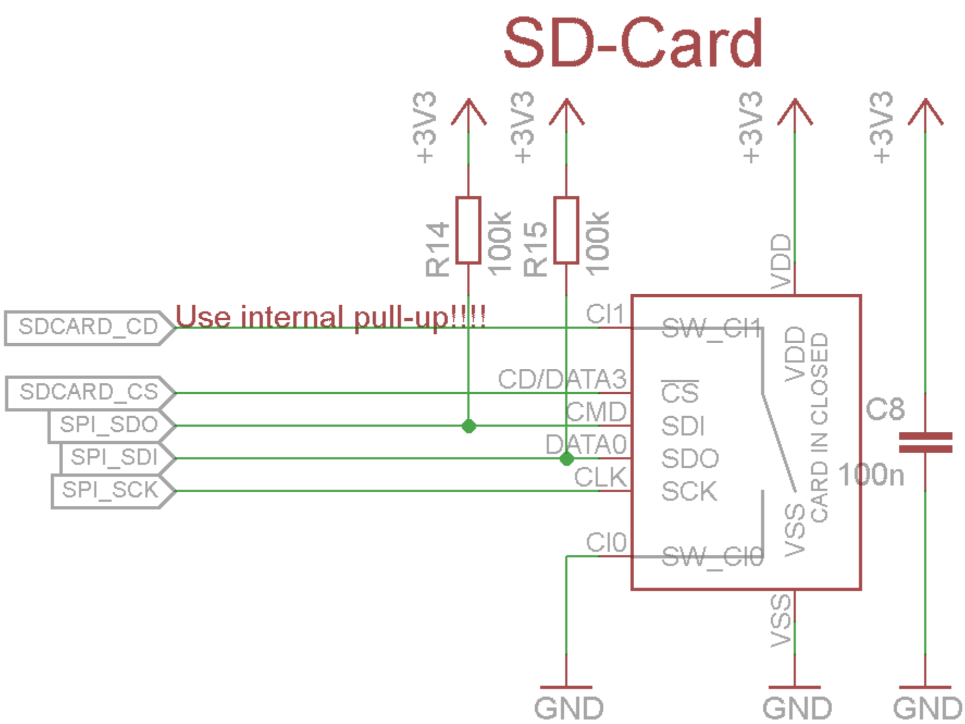


Figure 36: SD-card schematics

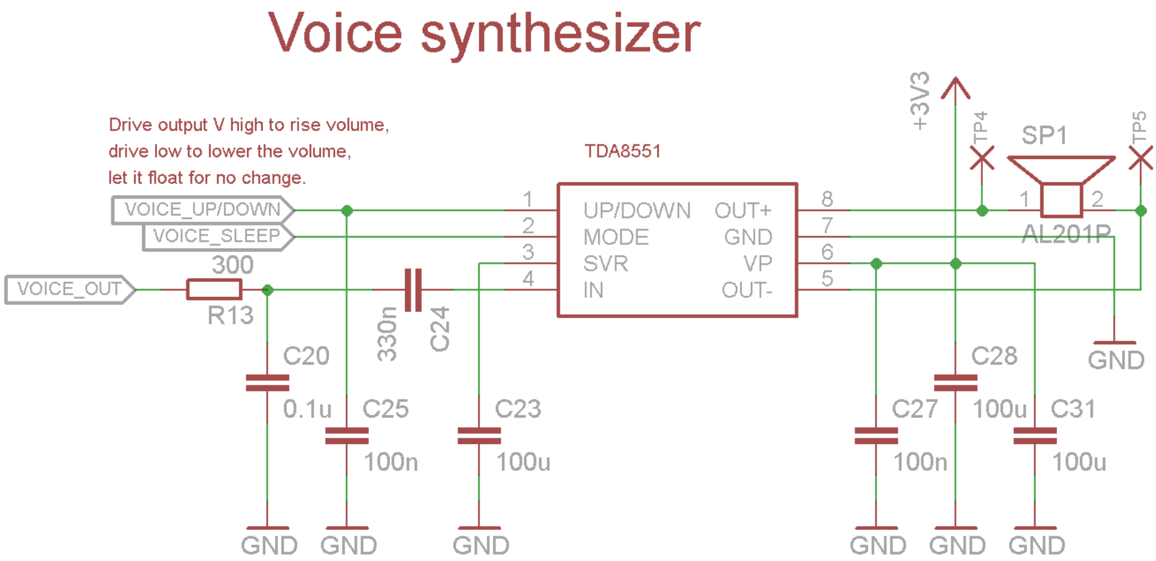


Figure 37: Voice synthesizer schematics

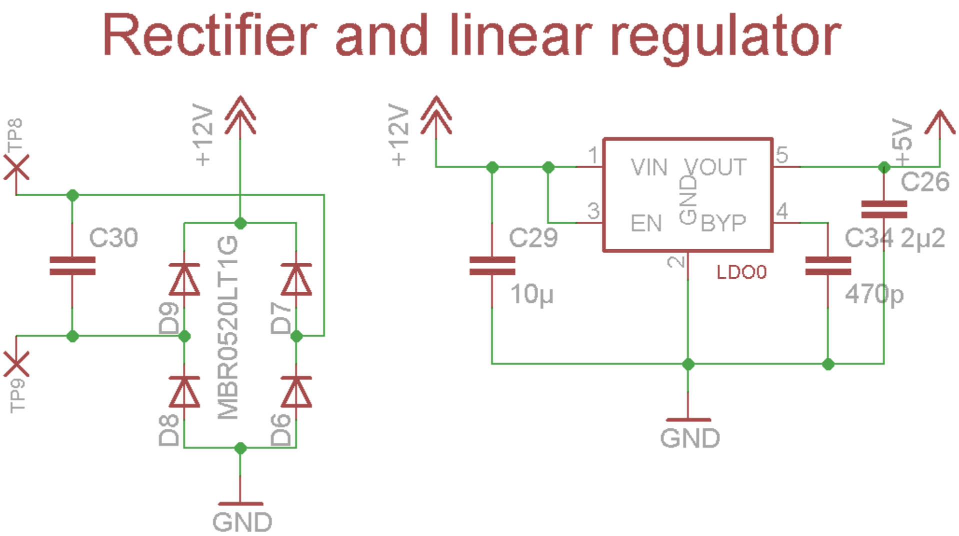


Figure 38: Rectifier and linear regulator schematics

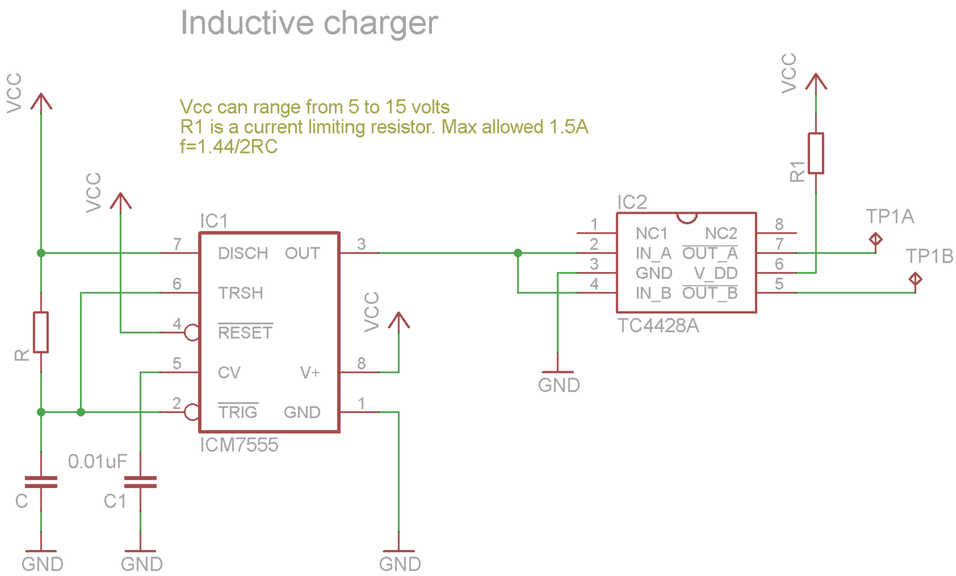


Figure 39: Inductive charger schematic

Test suite

This Test Cases are intended to evaluate the PHDI prototype **rev. 1**.

Personal information

Name	
Date	

Test Cases

Test case:	**Bracelet initialization**
Purpose:	This test checks if a fresh PHDI bracelet can be initialized successfully.
Prerequisites:	• Battery charged. • Fresh bracelet (not ever initialized)
Test Data:	Audio message 1: “Press the button 'add a client' in the Android tablet” Audio message 2: “Personal Health Data Identification”
Steps:	1. Press the button. 2. Verify that the audio message 1 is played. 3. Press the 'add a new client' button in the Android tablet 4. Verify that the audio message 2 is played.
Questions:	• Was the bracelet added successfully to the Android tablet ◯ Yes ◯ No
Comments	

Test case:	**RFID Tag Read and Sent to AT**
Purpose:	This test checks if an RFID tag can be read successfully and sent to the Android tablet.
Prerequisites:	• Battery charged. • Device in idle state (i.e. PHDI audio intro. already played.) • Android tablet ready to receive messages
Test Data:	Audio message 1: "Tag read successfully" Audio message 2: "Tag sent successfully"
Steps:	5. Press the button. 6. Closer the bracelet to the tag. 7. Verify that the "Tag read successfully" record is played. 8. Verify that the "Tag sent successfully" record is played.
Questions:	• Was the "Tag read successfully" record played? ◯ Yes ◯ No • Was the "Tag sent successfully" record played? ◯ Yes ◯ No
Comments	

Test case:	**RFID Tag not read.**
Purpose:	This test checks if after pushing the button, the bracelet can handle not to detect a tag.
Prerequisites:	• Battery charged. • Device in idle state (i.e. PHDI audio intro. already played.)
Test Data:	Audio message 1: "Tag not read, press the button to try again"
Steps:	1. Press the button. 2. Wait for 20 seconds. 3. Verify that the "Tag not read..." record is played.
Questions:	• Was the "Tag not read..." record played? ◯ Yes ◯ No
Comments	

Test case:	**RFID Tag Read. AT not available**
Purpose:	This test checks if an RFID tag can be read successfully but not sent to the Android tablet.
Prerequisites:	• Battery charged. • Device in idle state (i.e. PHDI audio intro. already played.) • Android tablet not ready to receive messages
Test Data:	Audio message 1: “Tag read successfully” Audio message 2: “Android tablet not reachable”
Steps:	1. Press the button. 2. Closer the bracelet to the tag. 3. Verify that the “Tag read successfully” record is played. 4. Verify that the “Android tablet not reachable” record is played.
Questions:	• Was the “Tag read successfully” record played? ◯ Yes ◯ No • Was the “Android tablet not reachable” record played? ◯ Yes ◯ No
Comments	