EASYID

A WIRELESS SENSOR PACKAGE TO MEASURE HUMAN INTERACTION FEATURES
EASYID

A WIRELESS SENSOR PACKAGE TO MEASURE HUMAN INTERACTION FEATURES

THESIS

Master of Science Thesis

For the Degree of Master of Science in Biomedical Engineering at Delft University of Technology

Mengdi Liu
Supervisor:

Dr. ir. Andre Bossche
Jeroen Bastemeijer
Laura Cabrera

Committee:

Prof. dr. Paddy French
Dr. ir. Andre Bossche
Prof.dr.ir. Wouter Serdijn
Laura Cabrera

Copyright © 2017 by M. Liu

An electronic version of this dissertation is available at http://repository.tudelft.nl/.
Simplicity is the ultimate sophistication.

Leonardo da Vinci
## CONTENTS

**Summary** ix  
**Acknowledgements** 1  

1 **Introduction** 3  
  1.1 Motivation 5  
  1.2 Contribution 5  
  1.3 Outline 6  

2 **Related Work** 7  
  2.1 Background 8  
  2.2 Related Work 8  
  2.3 Sensors 12  
    2.3.1 Motion Sensors 12  
    2.3.2 Proximity Sensor 13  

3 **EasyID: Hardware Design** 17  
  3.1 System Requirement 18  
  3.2 Single-board Computer 18  
  3.3 ID Detection Module 19  
    3.3.1 Receiver 19  
    3.3.2 Transmitter 20  
  3.4 Motion Detection Module 23  
  3.5 Miscellaneous 25  
  3.6 Hardware Architecture 26  
  3.7 PCB Layout 27  

4 **EasyID: Software Design** 29  
  4.1 System Architecture 30  
  4.2 ID Recognition 30  
    4.2.1 Encoding 31  
    4.2.2 Decoding 33  
    4.2.3 Data Link 34  
  4.3 Motion Sensor 35  
  4.4 Data Synchronization and Storage 36  

5 **EasyID: Evaluation** 37  
  5.1 ID Detection Module 38  
    5.1.1 Static scenario 38  
    5.1.2 Dynamic scenario 41
5.2  Motion Detection Module ........................................... 43
   5.2.1  Zero Level .................................................. 43
   5.2.2  Moving Objects ........................................... 44
   5.2.3  Reliability .................................................. 47
5.3  Senser Badge .................................................... 49

6  Conclusion ........................................................... 53
   References ......................................................... 56

Appendix ............................................................... 59
Extensive studies have been developed for the last few decades, for a better understanding of human behaviors’ roles during social events. Non-verbal behavioral cues as postures, body movement are of high importance to be captured during social interactions for those studies. Instead of using self-report methods, the fast, objective and long lasting automated sensing technique is used.

This thesis proposes a low-cost and small size customized data acquisition sensor badge, called EasyID, which is specially designed for the Vision Lab Group in TU Delft for their on body data capture during social events. EasyID is designed based on an open source platform, which provides enough space for future upgrade and extension. The important sensor functions are badge identification (ID) by infrared communication and motion capture through a 9 DoFs inertial measurement unit. The success of ID detection depends on the line-of-sight between the infrared transmitter-receiver pairs. Through pulse width modulation (PWM), any ID information can be sent through the infrared LEDs and can be recognized by other facing EasyIDs. The open source software that created to drive the circuits, is written in Python, which can be shared with other similar implementations. Different from C/C++ language, Python enhances the code readability for our clients and realizes a high-level programming method to drive the hardware. To overcome the delays while modulating the LEDs, an oscillator is used to generate the carrier frequency. The performance of EasyID’s ID detection is evaluated under both static and dynamic scenarios. With IR LEDs, the maximum range of detection is 3.5 meters distance and 120 degrees opening angle. For a fast and accurate motion capture, a 9 DoFs inertial sensor is selected and embedded on the add-on board. The sampling frequency of motion capture is typically 20Hz. The validation results have shown a 0.4% - 0.9% variation of the gyroscope, a 3.2% - 4.9% variation of the accelerometer and a 0.08% variation of the compass during zero motion test. Finally, a highly accurate global time stamp can be stored together with the above data from sensors as data logs. The EasyID is weighted 158g includes the Raspberry Pi and a 2200mAh rechargeable battery. The dimension is 85 × 56 × 25mm. The total price is 64 euros.
This thesis is submitted for the degree of Master of Science at the Delft University of Technology. This work is to the best of my knowledge original, except where references are made to the previous work.

Time flies. It has been roughly eleven months since I have started this thesis project. The completion of this project is approaching. At this moment, I would like to express my thanks for those who have helped and guided me through the progress of this project.

Please let me acknowledge at the start my appreciation of Dr.ir. Andre Bossche, my direct supervisor, who introduced me to this thesis work that activated the magnets of curiosity, knowledge, and wisdom in me. His patience, enthusiasm, and immense knowledge have pushed me forward. It is him, where I got the base of integrated sensors and this knowledge will benefit my future academic life, including the project I am undertaking now. And my special thanks to Prof. Dr.Wouter Serdijn and Prof. Dr.Paddy French, for their lectures in wearable medical sensors and devices. In their classes, they always present the material in a clear and logical sequence and demonstrate an expert knowledge in implantable devices as well as biomedical sensors. And the continuous unlimited sources provided from TU Delft is what keeps inspiring me during the project, especially when I am facing with problems that I couldn't solve according to what I had learned.

More specifically in the context of circuit design and manufacturing, I would like to thank Jeroen Bastemeijer who helped put forward the project by offering me with abundant information in hardware, software design and especially for soldering electronic components. As a biomedical engineering student who has little background in software, I would not be able to finish the codes with specialized theories and detailed data collection without him. My thanks are to his expertise, knowledge, and patience.

What I know through the whole project, is not only advances myself in electronics design for wearable sensors but provides me with collections of amazing views held by my colleague in EI Lab, Zu-yao Chang who helped me a lot with soldering the MEMS sensor, Douwe van Willigen who shared his experiences in PCB layout with me, and Laura Cabrera who kept concerning the process of the project and provided valuable feedbacks, which encouraged me the passionate about the project. The interesting and stimulation discussions I shared with them are precious. And I enjoyed the time working together with them.

At last, I would like to say thank you to my parents, my boyfriend Hongjia Wu and my best female friend Yaxi Peng. Their encourages, love and support have fueled me with confidence.
A spate of extensive studies involved in social interactions was developed in the past few years. Non-verbal behavioral cues as postures, body movement are of high importance to be captured during social interactions for those studies. Instead of using self-report methods, the fast, objective and long lasting automated sensing technique is used. In this chapter, first, a brief introduction of the thesis background is given, as well as the importance of automated sensing social behaviors. Then, the motivation of this thesis is followed by comparing state of the art sensor packages. The contribution of the thesis is listed in a general description. Improvements in more details will be shown in the following chapters. Finally, this chapter outlines the structure of the following chapters.
1. **Introduction**

Social interaction can be defined as a simple or complex event in which a certain agent performs some social actions addressed at another agent that is actually or virtually present [1], which can be understood as human behaviors to express certain self-regulated human attitude, as beliefs, evaluations, social emotions etc. Even currently, people heavily rely on email, telephone, and virtual communication platforms as Facebook; human sciences (psychology, anthropology, sociology, etc.) have shown how social interactions dominate our perception of the world and shape our daily behavior [2]. Therefore, a spate of extensive studies involved in social interactions was developed in the past few years.

**But how can we measure social interactions?**

There is a spate of social signals that can be used for the study of human-human interaction. The social signal is the communicative or informative signal that provides information about social interactions. Instead of detecting verbal social signals using a microphone, non-verbal behavioral cues as posture should be taken as a more important aspect of social aspect [2]. Because several decades of human sciences have shown that we are surprisingly effective at understanding social signals underlying the rich variety of nonverbal behaviors displayed by people around us [3]. And this makes it possible to use non-verbal social signals to define human interactions.

![Figure 1.1: Nonverbal Behavioral Cues and Social Signals during Social Interactions (Lepri B, Staiano J, Rigato G, et al., 2012)](image)

Instead of using self-reports and recall surveys which have difficulties in recalling activities in the past; are subjective; and are impossible for long-term measurement, we use automated sensing techniques. The rapid development of sensors and signal processing technology has made it possible to capture social interaction features on human through sensor badge during face-to-face conversations [4, 5]. As **wearables** become more and more integrated into our daily life, it provides the opportunity to use appropriate sensors to capture information about how groups of people interact over periods of weeks, months and even years [6]. They can precisely detect a range of events and can be precise, responsive and permanently operational [7]. Especially, the small size and lightweight Microelectromechanical systems (MEMS) sensor can enable reliable data captured from humans in a fast and unobtrusive way. This makes it accessible to use the wearable sensor packages, that are uniquely designed to collect motion and identification data from people wearing them and answer epistemological questions on human behavior in a data-driven manner, which provide prescriptive guidelines for persuading humans to undertake certain actions in real-world social scenarios.
1.1. Motivation

This thesis is motivated by the research focusing on devising models to automatically interpret face-to-face human social behavior using wearable sensors in the Vision Lab Group TU Delft. The wearable sensors can automatically capture human behavior during social events as meetings, the data will be used to synchronize with video cameras observation of people wearing the devices, which helps the researchers gain better understanding user roles in those meetings. Most of the state-of-the-art wearable sensor packages are able to measure human social behaviors in a fast and unobtrusive way, however, they are specially designed for a certain project, to measure certain types of social signals, which makes it inconvenient to use the packages during other studies or even an extension of the certain project. This may cause missing some major variables as well as unnecessary data collection that leads to extra power consumption. For example, the Sociometer, has a MEMS microphone on board, which is not necessary for nonverbal behaviors detection. At the same time, sociometer does not have any gyroscope, that makes it impossible to define the orientation or angular speed of users. Meanwhile, open source platform based sensor package can be adaptable during the future development and research work, which is more flexible than a Printed circuit board (PCB). Considering the fast speed development of sensors and signal processing technology, open source platform can be upgraded and manufactured easily in the long term.

The study goal of the thesis is to design a low-cost and small size customized data acquisition sensor badge, which is specially designed for a certain client and meet all the requirements, but also provides enough spaces for future upgrade and extension.

The sensor badge should have motion sensors to capture human behavior related to their body movement; should detect and recognize any facing device with the same configuration; and should store all data in a full synchronized fashion, with the global time stamp as a reference. The device must be small and light and be able to run for at least a day (6-8 hours) on the same battery while storing all sensor data as it records. The devices will be operated as part of a network with many nodes.

1.2. Contribution

This thesis presents a low-cost and small size customized sensor package, EasyID, based on an open source platform. The wearable sensor package can collect high accurate real-time human body movement data and identification (ID) data of anyone wearing them. The whole system, when scaled to a large dense crowded scenario, up to 256 devices can be able to be configured.

The main contributions of the thesis are listed:

1. Customized PCB EasyID consists of an open source platform and an add-on Printed circuit board (PCB) with infrared communication module (transmitter, receiver)

---

1 Computer Vision Lab is a part of the Pattern Recognition & Bioinformatics Group in TU Delft, which develop new techniques for the automatic analysis of images and videos.
for ID detection, a MEMS 9 DoFs inertial sensor for human motion sensing, a real
time clock for accurate time stamp, two indicators, and one push button for user
interfaces to establish a real-time data collection sensor package at a high level of sensitivity.

2. **Open Source Platform** EasyID is designed on an open source platform Raspberry
Pi 3 Model B and the add-on board is accessible by Raspberry Pi 2 Model B, Rasp-
berry Pi Model B+, and all Raspberry Pi 1 Model. It can be easily upgraded, manu-
factured, which enables the possible extension of the sensor package.

3. **Python Programming** To control the EasyID, a high-level programming language,
Python is used, which enhances code readability. Compared to C/C++ language,
python is slower, which causes the problem while modulating infrared transmitters. A hardware solution is designed to solve the problem through a 555 oscillator.
At the same time, the open source software developed to drive the circuits in this
thesis can be shared with other similar implementations.

4. **Low Cost and Small Size** The overall cost of the manufactured EasyID is 64 euros.
The dimension of EasyID is $85 \times 56 \times 25\text{mm}$.

1.3. **Outline**

The rest of this thesis is organized as follows: related works and theoretical back-
ground are provided in Chapter 2. The hardware design of EasyID is explained in
Chapter 3. Details of the software design and system architecture are given in Chapter
4, followed by the experimental evaluation of EasyID presented in Chapter 5. Finally,
conclusions and future work are summarized in Chapter 6.
EasyID is such a wearable sensor package that automatically measures human interaction features. It is a wearable data acquisition board, and functions as data recording during human face-to-face conversations. Data collection through wearables can be lower in price and more reliable than using human-delivered questionnaires, because a sensor-based approach is free from recall failures and personal interpretation bias of surveys, and does not involve coextensive human coders/observers. In general, sensor package that measures social behaviors is fast becoming ubiquitous with their implementation in wearable sensors. This chapter will explain what is sensor badge, and what makes a fully functional sensor badge for collecting human face-to-face interaction data. Related works will be described according to the published time of them. Through the comparison and analysis of previous work and EasyID, the improvement of EasyID will be listed. Consequently, the related theoretical background of motion sensing and proximity sensing will be described after the related works.
2.1. BACKGROUND

Sensor badges capable of detecting face-to-face interactions, body movement, and proximity to others, through non-verbal social signals, have been developed recently to capture individual and collective patterns of human behaviors. This can be correlated with those individual (group) performances as well as the effects of the face-to-face communication. The ability to discover face-to-face communication networks automatically allows scientists and researchers to gather interaction data from larger groups of people.

The design of the social signals measurement badges was motivated by the fact that a large number of organizations already require employees to wear RFID name tags that identify them and grant them access to several locations and resources. These traditional RFID name tags are usually worn around the users’ neck or clipped to their clothing. With the development of microelectronics, it is now possible to augment RFID badges with more sensors and computational power that allow measuring human behavior without requiring any additional effort on the user’s side. By capturing individual and collective patterns of human social interaction with sensor package, it is possible to identify users’ personality as well as their particular interesting during social events.

In the following subsections, several major measurements and their theoretical background will be explained individually, i.e. motion capture, proximity sensing including infrared communication method.

2.2. RELATED WORK

In this section, available wearable sensor systems that are used as social interaction data acquisition will be described, and their basic hardware and software configurations will be given as well.

Choudhury and Pentland [6] initiated behavior analysis using wearable sensors by developing the Sociometer in 2003, which were worn by participants to record various aspects of their behavior. The board has an Infrared (IR) transceiver, a microphone, two accelerometers, on-board storage, and its own power supply. The wearable stores the data locally on a 256MB compact flash card and is powered by four AAA batteries. The
range of detection is approximately six feet, which is adequate for picking up face-to-face communication [6]. One of the limitations is that it can only detect two people when they are involved in social interaction.

The sociometer (Figure 2.1) records the following information for each individual:

1. Information about people nearby (sampling rate 17Hz – sensor IR)

2. Speech information (8KHz - microphone)

3. Motion information (50Hz - accelerometer)

Similarly, one early sensor platform was developed in 2002 at Massachusetts Institute of Technology (MIT) (Figure 2.2, application see Figure 2.3, which is the data acquisition system for user-interface implementation, including exploring heart rate, breathing, temperature, and skin conductance changes in different situations [8]. The whole system includes a data-collection system consisted of a PIC 16C711 microcontroller, an FM radio-transmitter module, and a power supply circuitry with a 9V battery. The microcontroller provided four 8-bit analog-to-digital converter inputs, additional output pins to control optional LEDs or other components, and processing capabilities to assemble and send data packets over the RF link using a serial protocol [8]. The receiver module was connected to the serial port of a PC or other device, and the data were broadcast continuously in packets consisting of a predefined header byte, four samples from each analog input, and a check-sum byte.

Another badge platform developed at MIT for facilitating interaction in large groups of people was the UbER-badge [9]. (Figure 2.4) An UbER-badge measures 8.25 x 10.5cm and weighs 0.1kg with all four AAA batteries installed. At an average current of about 100mA, badges last for roughly 15 hours of continuous use. The badge is equipped both with an IR communication channel to support face-to-face communication, and an RF communication channel to support larger distance communication. The IR system consists of a composite IR LED with a 17-degree spread, and an IR receiver with integrated demodulator, photodiode, and a Cygnal C8051F301 processor, which acts as a dedicated IR communication controller to buffer incoming and outgoing IR messages. A slightly
quicker version of the Sony-IR protocol is used on the badges, with the IR modulated at 40kHz. The badge’s IR communication is sensitive up to 3 meters.

An improvement of the UBeR-Badge has been claimed in 2010 [10] (Figure 2.5). The improvement is basically the display. The badge is equipped with an IR channel (875 nm modulated at 38 kHz) to support face-to-face and local communication. Badges can notice each other via the IR channel up to 3 meters and across large angles (e.g., 60°). And the badge also supports an RF section to support higher bandwidth, non-line-of-sight communication across larger distances with a range of 100 meters.

Sociometric badge 2.0 and 3.0 [11] are two more developed ones compared to [6]. The front view of the sociometric badge 3.0 is shown in 2.6 2.7. The badge 3.0 has a three-axis MEMS accelerometer (Analog Devices, ADXL330) is used to detect when a person is moving. An IR transceiver module (Vishay, TFDU4300) is used to detect when two people are facing each other. A bridged-output audio power amplifier (Analog Devices, SSM2211) drives an electromagnetic speaker on the badge to play back messages and reminders. The main processing unit is an ARM microcontroller (Atmel, AT91SAM7S256). A 2.4-GHz wireless transceiver (Chipcon, CC2500) and a class 2.0 Bluetooth module (BlueRadios, BR-46AR) have been incorporated for enabling wireless communications with fixed base stations and other Bluetooth-enabled devices. The badge is powered by a 950-mAh lithium-polymer battery that is rechargeable through USB. In ad-
2.2. RELATED WORK

dition, data can also be transferred through the USB port. The dimensions of the badge inside the plastic enclosure are $4.5 \times 10 \times 2$ cm, and the total weight including the battery is 110 g.

More recent works of wearable sensor package towards the ultra portable badge, are designed to monitor body movement and the social interactions. The arrival on the market of major players like Apple, Google and Microsoft popularized smartwatches and smartglasses and facilitated the development and widespread adoption of sensing applications (e.g. with Android Wear), opening doors in many areas including sport and personal monitoring.

Compared to the above-proposed work, this thesis has following improvement:

1. **Open source platform** EasyID is designed on an open source platform Raspberry Pi module B+ vision 3 and can also be accessed by all Raspberry module A and B. And this can be easily upgraded, manufactured and enable the possible extension of the sensor package.

2. **Python Programming** To control the EasyID, a high-level programming language, Python is used, which enhances code readability. The open source software developed in this thesis to drive hardware can be shared with other similar implementations.

3. **Faster Signal Processing** EasyID has a 1.2GHz Broadcom BCM2837 64bit CPU with a 1GB RAM compared to a 400 MHz previous work in 2011. This enables a faster signal processing during data collection.

4. **Larger Data Storage** EasyID has a micro SD card socket, which can insert up to 128GB SD card for loading the operating system and storing the data.
5. **Larger Detection Range** The ID detection can sense facing devices with the same configuration up to 3 meters away. The maximal viewing angle of the device is 120 degrees.

6. **Low cost and small size** The overall cost of the manufactured EasyID is 64 euros. The dimension of EasyID is $85 \times 56 \times 25$mm.

7. **Larger Dense Scale** EasyID can scale up to 256 devices with 8-bits data transition. There are sufficient spaces for future upgrades, which can easily be scaled up to 2084 devices with the current hardware configuration.

### 2.3. SENSORS

#### 2.3.1. MOTION SENSORS

It is possible to recognize meaningful expressions of emotions as talking, gesturing and laughing through the monitoring and analyzing of body motions. [12–15] Motion capture (mocap) is the process of recording the body motion of objects or people, which can be obtained by the inertial navigation system (INS).\(^1\) INSs were first developed for rockets in the Second World War. In the decade that has followed, a wider range of application is realized, from airplanes or the automotive industry to smart phones or even running shoes with the development of new technology, with the benefits from MEMS technology. MEMS technology (Microelectromechanical systems) offers rugged, low cost, small and lightweight inertial sensors relative to the other available technologies.

![Figure 2.8: Accelerometer System: a) electromechanical schematic; b) equivalent system model; and c) simplified sensor interface schematic. (Shaeffer D K., 2013)](image)

---

\(^1\)INS is defined as a navigation aid that uses motion sensors (Accelerometers) and rotation sensors (Gyrosopes) to continuously calculate the position, orientation, and velocity of a moving object without the need for external references.
MEMS relies on miniaturized electromechanical elements for their operations. Three main types of input sources used in inertial sensing, are accelerometer, gyroscope and electromagnetic. They can be used individually to capture human motion, however, the need of the combination of them is growing. How each of them functions individually and in a system will be given. Accelerometers use a comparatively simple system to drive the MEMS, whereas gyroscopes require significant complexity to maintain self-oscillation and regulate the mechanical amplitude of the drive motion.

One of the main advantages of accelerometers is their relatively lower error rates as compared to other motion sensors: their bias errors can be estimated more easily than gyroscopes. However, its measurements are sensitive to dynamics which are not related to the motion that aims to detect, for example, the involuntary vibrations of a human body. Therefore, such dynamics need to be reconciled or smoothed during the features extraction. The effect of the sampling rate of an accelerometer on recognition accuracy was evaluated by Maurer et al [16] and it was found that there is no significant gain for having sampling frequency higher than 20 Hz.

Rotation measurements from gyroscopes are necessary to discriminate between motion modes whose acceleration measurements may seem almost identical, e.g., between a stationary platform at zero velocity and a platform moving at constant velocity. The main drawback of gyroscopes is that they tend to have high error rates: they usually have higher drift errors and higher run-to-run bias errors than accelerometers. Such errors can be estimated and partially removed by combining it with a magnetometer [17].

2.3.2. Proximity Sensor

Proximity sensing can be defined as the detection of nearby objects presence without any physical contact. Proximity sensing can be achieved either directly by sensing proximity, or indirectly by inferring their absolute positions [18]. The Table 2.1 lists the most common methods to detect proximity. The electromagnetic gradient is usually subdivided into different spectrums according to their energy, namely radio waves, microwaves, infrared, (visible) light, ultraviolet, X-, and gamma radiation. Among all interfaces to measure distance (approaching distance) between two nodes, Radio Frequency (Bluetooth, Wifi, etc) and infrared radiant are low cost and most used to measure distances. At the same time, infrared can be used to measure face-to-face conversation features with the line-of-sight.

Infrared band of the electromagnet corresponds to 430THz to 300GHz and a wavelength of 980nm. The propagation of light waves in this band can be used for a communication system (for transmission and reception) of data. IR communications are based on technology which is similar to the remote control devices such as TV and entertainment remote controls used in most homes today. IR offers a convenient, inexpensive and reliable connection between two portable devices or between a portable device and a fixed device. Unlike radio-frequency (RF) wireless links, IR wireless cannot pass through walls, line-of-sight mode. Therefore, IR communications or control is generally not possible between different rooms in a house, or between different houses in a neighborhood. This might seem like a disadvantage, but IR wireless is more private than RF wireless.
### Table 2.1. List of proximity sensing mechanisms

<table>
<thead>
<tr>
<th>Type</th>
<th>Theorem</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical sensors</td>
<td>Change of the electromagnetic field generated by coil, target converted into analogue output</td>
<td>Requires metal coil, Small measurement range, Range depends on sensor, Low conductivity</td>
</tr>
<tr>
<td>Capacitive</td>
<td>Two conductive plates with minimal supporting electronics to detect the obstacles</td>
<td>Ambiguous data, quickly decaying resolution and exposure to external influences</td>
</tr>
<tr>
<td>Optical sensors</td>
<td>The reflections of the light sent out can be detected by photodiode</td>
<td>Unstable with environmental changes, Dark color, non-matte surfaces not be sensed</td>
</tr>
<tr>
<td>Acoustic sensors</td>
<td>Ultrasonic energy can be transmitted, then received</td>
<td>Vulnerable to noise, Expensive to maintain, Flat surfaces hard to detect</td>
</tr>
<tr>
<td>Microwaves</td>
<td>Transmit a short pulse detect the reflection via microwaves</td>
<td>Complex and expensive, Doppler effect, Noisy output</td>
</tr>
<tr>
<td></td>
<td>Transmitted and reflected microwaves</td>
<td>Doppler effect, Large power supply</td>
</tr>
<tr>
<td>Radio Frequency</td>
<td>Signal strength can be used to measure distance</td>
<td>Biased measurement around RF cellular towers</td>
</tr>
</tbody>
</table>

As early as 1992 [24], the Active Badge 2.10 was developed as the earliest active infrared-based system, consisting of a cellular proximity system. Each person wears a small infrared badge which emits a globally unique identifier one-tenth a second every 15 seconds to the ceiling central nodes for estimation of badge’s location. PWM infrared signals are used for communication between badge and sensor. Since then, the infrared communication has been used for the proximity and location sensing between nodes.

The transmitter of an IR LED inside its circuit, which emits infrared light for every electric pulse given to it. This pulse is generated as a button on the remote is pressed, thus completing the circuit, providing bias to the LED. The LED on being biased emits light of the wavelength of 940nm as a series of pulses, corresponding to the button pressed. However since along with the IR LED many other sources of infrared light such as us human beings, light bulbs, the sun, etc, the transmitted information can interfere. A

![Infrared Communication](image-url)
solution to this problem is by modulation.

Pulse Width Modulation (PWM) \cite{25} is a method for generating an analog signal using a digital source. A PWM signal consists of two main components that define its behavior: a duty cycle and a frequency. The duty cycle describes the amount of time the signal is in a high (on) state as a percentage of the total time of it takes to complete one cycle. The frequency determines how fast the PWM completes a cycle and therefore how fast it switches between high and low states. The transmitted signal is typically modulated with a carrier frequency of 38 kHz (or any other frequency between 36 to 46 kHz). By cycling a digital signal off and on at a fast enough rate, and with a certain duty cycle, the output will appear to behave like a constant voltage analog signal when providing power to devices, which can be used for ID information transaction during social features measurements. It is the most common method because it is easy to control with low error rate. Besides from PWM, there are several Amplitude-shift keying (ASK) modulation encoding methods, as Pulse Position Encoding, Pulse Distance Encoding, and Manchester Encoding.

The receiver consists of a photodetector which develops an output electrical signal as light is incident on it. The output of the detector is filtered using a narrow band filter that discards all the frequencies below or above the carrier frequency (38 kHz in this case). The filtered output is then given to the suitable device like a Microcontroller or a Microprocessor which controls devices like a PC or a Robot. The output from the filters can also be connected to the Oscilloscope to read the pulses.
While a spate of studies on the design of sensor packages has shown fast and accurate human features capture, the up-to-date wearable sensors together with single board computers can offer faster and more reliable human data collection methods. In this chapter, the customized sensor package – EasyID will be introduced from the system requirements to the hardware configurations. Circuit design and selection of components will be described individually. This chapter contains the contents of the selection of open source platform, which performs as the power driver and microcontroller for the whole device. At the same time, the choice of major components as motion sensors and infrared transceiver and their designed circuits are shown together with the reasons for selection. The block diagram of hardware design will be given. Finally, the sensor package is manufactured as a PCB. PCB layout is given at the end of this chapter.
3.1. SYSTEM REQUIREMENT

According to the previous chapter, several design instructions were already given based on the literature review. In this section, a general description of the functional and non-functional requirements will be given. For more detailed information, the list of full requirements is attached in the appendix 6.

Overall, the sensor package should be able to measure body motion data (acceleration, orientation, and velocity), detect any facing devices with the same configurations and recognize their IDs. In the meantime, it should be able to store all the data in a fully synchronized fashion, using a global time stamp as a reference. For the motion capture module, the acceleration of the users should be measured with a range of ±2g at a resolution of 0.1g and a maximum deviation of ±0.1g; the linear velocity of the users should be measured with a range of ±10m/s at a resolution of 0.1m/s and a maximum deviation of ±0.1m/s; and the angular velocity of the users should be measured with a range of ±90°/s at a resolution of ±1°/s and a maximum deviation of ±5°/s. At the same time, the sampling frequency of the motion data should be at least 20Hz. For the face-to-face detection modules, the detection range should be 3 meters, and the data rate should be at least once per second. Sufficient I/O capabilities (>10) should be left open for hardware extension. And the total weight of the device (including the battery and the enclosure) should not be too large. A small size of the device is necessary for the user’s comfort. The whole system, when scaled to a large dense crowded scenario, at least 32 devices should be able to configure.

3.2. SINGLE-BOARD COMPUTER

Sensor badges are small in size and light in weight, which requires high-performance sensors as well as a small and cheap microprocessor. The operation of a single-board computer (SBC) is such a flexible platform that offers a fast multitasking capacity with a high level of automation [26]. Among all SBC, ARM is the leading in microprocessor technology aiming at designing a wide range of multi-functional, power-efficient, low-cost, convenient size and high-performance microprocessors. Because of its unique features and interfacing options, ARM technology is currently used in a large number of development board as well as wearable devices and handheld gadgets. Currently, there are 121 ARM based single-board computers with different ARM architectures. Based on the requirements of the wearable sensor packages, we selected some common Linux oper-
ating boards with low cost and small size characteristics and gave a comparison among 
them. Linux operation system is a collaboration of free and open-source software. With 
Linux, free and open-source software development and distribution are possible, which 
benefits the extension and upgrade of EasyID. Among those selected Linux based ARM 
SBC, Raspberry Pi 3 Model B [27] (Figure 3.1) is selected, as the microcontroller and de-
development platform for EasyID. The reasons are following: (See full comparison in the 
Appendix 6)

1. Fast processor and large storage capacity
2. Large user community and extensive software available.
3. Relatively small size, low cost with low power consumption.
4. On board 802.11n wireless and Bluetooth 4.1 modules enable proximity sensing 
   with a range of 5 meters.

3.3. ID Detection Module

The ID recognition module of EasyID will be described in this section. According to 
the previous literature review, for the ID detection module, we use infrared commu-
nication, which is composed of two parts: infrared receiver and infrared transmitter. We 
start with the selection of the components, followed by their circuits and their layout 
when embedded on the PCB.

3.3.1. Receiver

Infrared receivers are selected based on the carrier frequency, application, maximum 
and minimum irradiance, and opening angle. The target receivers are among those re-
 mote control ones, because they are low in cost, small in size, and can easily detect 
modulated infrared signal from the environment with a low supply current. The desired 
receiver should consume the lowest power but can receive a large range of signals. So 
the receiver with lowest minimum irradiance (defined as power consumed per unit area 
[28]) but the highest angle of half transmission (opening angle) is the best solution, for 
the sensing distance is defined as the square root of the division between receiver mini-
 mum irradiance and transmitter radiant intensity (Figure 3.2). According to the require-
ments listed in the previous section, the receiver should see signals from a wide (180°)

![Figure 3.2: Relevant Values for IR Transmission Distance](image-url)
range, however, it is not possible for one receiver to receive signals in a range of 180°. So multiple receivers will be needed to meet this requirement. The frequency should be typically 38kHz, which should be the same frequency as the transmitter’s carrier, and ideally, all other noise is blocked by the receiver’s bandpass filter.

TSOP34438 is such an infrared receiver that has the lowest minimum irradiance among all 45 degrees angle of half transmission 38kHz modules. With 45 degrees angle of half transmission, two receivers should be sufficient. Meanwhile, two receivers can provide two different sensing zones when receiving ID information. This can help us separately store ID information from different devices that laying on different orientations towards the receiver device. The electrical connection is shown in Figure 3.3. The three pins are data out; ground; and power respectively. A decoupling capacitor C is used to stabilize the power. And a pull-up resistor is 4.7kΩ as usual.

Since the largest possible half transmission angle is 45 degrees, two infrared receivers are sufficient to cover the full range of 180 degrees. When they are embedded in the PCB, they should be bent at 45 degrees to the board, and then rest on the chest of the user. With two infrared receivers connected to two different GPIO pins (4, 18 GPIO.BCM), their received data can be stored separately which helps better to derive facing angles of the ‘talking devices’.

3.3.2. Transmitter
For the infrared transmitter, we need to select the IR LED and a stable oscillator for 38kHz carrier frequency. For 38kHz carrier frequency, we head for a 555 timer, because of its low price, ease of use, and stability in generating pulses. We select TLC555 over NE555 because:

1. It is functionally interchangeable with NE555.
2. It has a higher input impedance, which allows a smaller timing capacitors to be used. As a result, more accurate time delays and oscillations are possible.
3. The power consumption of TCL555 is low across the full range of power-supply voltage.
4. The package of TCL555 is SOIC8, which is one-fourth of the NE555 package.

1GPIO.BCM is referring to the pins by the “Broadcom SOC channel” number
3.3. ID Detection Module

The TLC555 [29] application circuit can be seen in Figure 3.6. It is capable of being used in astable and monostable circuits. In an astable circuit, the output voltage alternates between VCC and 0 volts, with the introduction of a resistor RB. By selecting values for $R_A$, $R_B$, and C we can determine the period/frequency and the duty cycle. To establish a 38kHz carrier frequency, theoretically, a $10nF$ capacitor, a $1.4k\Omega$ resistor ($R_A$), and a $1.2k\Omega$ resistor ($R_B$) will do a duty cycle of 60%. However, in reality, the circuit tends to show deviations in oscillation frequency, with a capacitor tolerance (1%). By putting a $8.1 – 9.1k\Omega$ in parallel with the $1.2k\Omega$ resistor, which causes a $0.12k\Omega$ decreased value in $R_B$ the carrier frequency can be set back to 38kHz.

When selecting the infrared transmitter, it is important to understand what makes a large infrared transmission distance. The maximum possible transmission distance of an IR remote control system (receiver and transmitter) depends on various parameters but is mainly dependent on the radiant intensity (the radiant flux emitted per unit solid angle) of the emitter (Ie) and the sensitivity of the receiver (Ee min.). Figure 3.2 explains how to calculate transmission ranges in the simplest case assuming a quadratic relationship between the distance d and the irradiance of the receiver Ee. Given emitter intensity Ie, the maximum distance is calculated as 3.1:

$$d_{\text{max}} = \sqrt{\frac{I_e}{E_{\text{emin}}}}$$  \hspace{1cm} (3.1)

A larger emitter intensity is what makes a better IR LED for infrared communication. Meanwhile, a wide transmitter angle can enable a wide sensing range during infrared detection. The typical intensity values of selected through hole emitters are listed in Table 4.2. We can see from the table that, the radiant intensity increases with decreasing of emission angle.

### Table 3.1. Emitters for TSOP Receiver Modules

<table>
<thead>
<tr>
<th>Emitter</th>
<th>Package Diameter</th>
<th>Wavelength (nm)</th>
<th>Radiant Flux $I_f = 100mA/mW$ typ.</th>
<th>Radiant Intensity $I_f = 100mA/mW$ typ.</th>
<th>Emission Angle</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSLB3940</td>
<td>3 mm</td>
<td>940</td>
<td>40</td>
<td>65</td>
<td>± 22°</td>
<td>T-1</td>
</tr>
<tr>
<td>TSAL6100</td>
<td>5 mm</td>
<td>940</td>
<td>35</td>
<td>60</td>
<td>± 10°</td>
<td>T-1/4</td>
</tr>
<tr>
<td>TSAL6200</td>
<td>5 mm</td>
<td>940</td>
<td>35</td>
<td>60</td>
<td>± 17°</td>
<td>T-1/4</td>
</tr>
<tr>
<td>TSAL6400</td>
<td>5 mm</td>
<td>940</td>
<td>35</td>
<td>60</td>
<td>± 17°</td>
<td>T-1/4</td>
</tr>
<tr>
<td>VSLB3940</td>
<td>3 mm</td>
<td>940</td>
<td>40</td>
<td>65</td>
<td>± 22°</td>
<td>T-1</td>
</tr>
</tbody>
</table>

*All IR emitting diodes shown in Table 1 are suitable for use with the Vishay IR receivers for standard remote control applications (38kHz).*

The selection of infrared LEDs 3.7 is through the comparison of the calculated radiant intensity at the same sensing point, same sensing distance and sensing viewing angle. The viewing angle is 30° and the sensing distance is 1 meter. We calculate the relative
radiant intensity according to the relationship between the angular displacement and radiant intensity for a certain fix angle (30°). The radiant intensity based on the distance in between the transmitter and the receiver. The total value is calculated relative radiant intensity, which is the product of the radiant intensity and relative ratio. Through the calculation, we select three best IR LEDs, with the highest total values of relative radiant intensity.

**Figure 3.7: LED Selection Criteria and Selected IR LEDs**

<table>
<thead>
<tr>
<th>Emitter</th>
<th>Radiant Intensity</th>
<th>30° Relative Ratio</th>
<th>Total Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSAL6200</td>
<td>72</td>
<td>0.15</td>
<td>10.8</td>
</tr>
<tr>
<td>TSAL6600</td>
<td>50</td>
<td>0.3</td>
<td>15</td>
</tr>
<tr>
<td>V1583940</td>
<td>65</td>
<td>0.2</td>
<td>13</td>
</tr>
</tbody>
</table>

**Figure 3.8: Infrared Transmitter Schematic**
In practice, the relationship between irradiance and transmission distance does not exactly follow a quadratic curve. In most cases, the actual distance and angle are both smaller than calculated as the expression. At the same time, when an opaque window or light pipe is used between the receiver and the emitter, the actual transmission distance will be reduced.

The electrical connection of the infrared transmitter is shown in Figure 3.8. There are three IR LEDs in parallel, together covering a full angle of approximately 180 degrees. Each of them connected to a resistor. The current flow through each of them is set to 10mA, which is sufficient for a detection range of 3 meters for the receivers. According to the datasheet [30, 31] of the IR LEDs, we calculate the voltage difference between the forward current and forward voltage relationship. The Vbe voltage gives us R1, R2, R3 value, which is 180Ω. The switches MOSFET Q1 and Q2 together control the pulses and spaces of the IR transmission through GPIO control by the Raspberry Pi. When the oscillator gives no signal to the circuit, the Q2 is switched off, which allows GPIO controls the LEDs. The 555 keeps giving modulated signals with 38kHz, this makes sure LED blinks with a constant 38kHz. If GPIO is High, the Q1 is on, the current directly flows to the ground, so that the LEDs are ON.

3.4. Motion Detection Module

MEMS (Microelectromechanical systems) rely on miniaturized electromechanical elements for their operations. Three main types of electronic positional sensors, used for capturing human body movements, are accelerometer, gyroscope and electromagnetic. They can be used individually to capture human motion, however, the needs of the combination of them are growing. They together can contribute to the 9 DoFs wearable inertial motion sensors. After looking into several MEMS inertial motion sensors from different providers as NXP, Analogue Device, InvenSense, Bosch, and STMicroelectronics, we selected several sensors with overall good performances but the low price and small size. With excellent sensing performance and full integration, LSM9DS1 [32] has been selected as the most common used IMU (Inertial Measurement Unit) for human motion capture. The performance of LSM9DS1 is approximately the same with LSMLDS1, but has an extra 3 DoFs compass measurement. One advantage of LSM9DS1 is that it is integrated on the Sensehat 2, which has been used to measure human motion on Raspberry Pi. A comparison of LSM9DS1 among selected sensors can be found in table 3.2.

The chip (Figure 3.9) and the example PCB (3.10) are shown. LSM9DS1 is a handful integrated circuit (IC) that measures three key properties of movement - acceleration, angular velocity, and heading. The LSM9DS1 measures body’s acceleration in gees. The minimum range of LSM9DS1 accelerometer capacity is ±2g, and can be scaled up to ±16g. The sensitivity of the accelerometer is ranging from 0.06mg - 0.7mg based on their measurement range. The gyroscope can measure angular velocity in DPS (degree per second) with a maximal range up to 2000dps. The sensitivity is 8.75 - 70 mdps. And finally the magnetometer has a range of ±400 - ±1600, measures the power and the direction of magnetic fields.

---

2Sensehat: An add-on board for Raspberry Pi, made especially for the Astro Pi mission.
The IMU electrical connection circuit is described in 3.11. As we can see in the Figure, there are four decoupling capacitors in between the power and the ground. According to the datasheet [32] Power supply decoupling capacitors (C6, C12, C8 = 100 nF ceramic, C11 = 10 µF Al) should be placed as near as possible to the supply pin of the device (common design practice). The reason for placing a 10 µF capacitor is to compromise the possible tolerance. Capacitor C7 (100 nF) should be a capacitor with low ESR (Equivalent Series Resistance) value and should be placed as near as possible to the C1 pin [32]. Pin 21 must connect to GND with the ceramic capacitor with 10 nF (±10%), 16 V. A pull-up resistor R9 is connected to the accelerometer and gyroscope data enable.

The LSM9DS1 is connected to two I2C pins of Raspberry Pi. The I2C embedded inside the LSM9DS1 behaves like a slave device and the I2C protocol must be adhered to. In the I2C of the accelerometer and gyroscope sensor, after the start condition (ST) a slave address is sent, once a slave acknowledge (SAK) has been returned, an 8-bit sub-address (SUB) is transmitted.
3.5. MISCELLANEOUS

Without the real time clock, data synchronization will be challenging and additional “global” base stations (badges placed in fixed locations and turned on at a precise date and time) will be needed. Each EasyID will keep track of its own global time in micro-controller clock-ticks that can be later on converted into the total amount of elapsed time since the device was turned on. However, each time the EasyID is turned off the clock-tick counter will be reset to zero and therefore several algorithms have to be implemented in order to synchronize each badge’s data and obtain real time-stamps. Therefore, a real-time clock is essential for large data collection studies.

The DS3231M is a low-cost, extremely accurate, 8 pins I2C real-time clock (RTC). The device incorporates a battery input and maintains accurate timekeeping when main power to the device is interrupted. The integration of the MEMS resonator enhances the long-term accuracy of the device and reduces the piece-part count in a manufacturing line. The DS3231M is available in the same footprint as the popular DS3231 RTC. The battery is a general small package type CR1220, which can provide power for RTC for at least 10 years.
Figure 3.14: Hardware Architecture Block Diagram

One push button is connected to a GPIO to control the power on and off of the add-on board. Two indicators can be used to show any information for setting up of the device. A 3 pins connector is useful when another expansion board is sacked on, to provide the 3.3V source; 5 V source; and the ground. All of them can be easily controlled by Raspberry Pi.

3.6. HARDWARE ARCHITECTURE

The overall hardware architecture block diagram is shown in Figure 3.14.

1. CPU Quad Core 1.2GHz Broadcom BCM2837 64bit CPU.

2. Memory 1GB RAM; Removable microSD memory card (up to 128 GB).

3. Sensors MEMS 9-axis IMU (to capture body movement); IR transceiver/receiver (face-to-face interaction detection); BCM43438 wireless LAN and Bluetooth Low Energy (BLE) (wireless data transfer and proximity detection).

4. Interface 4×USB 2.0; 10/100mb Ethernet; 26 GPIO pins.

5. Miscellaneous 2 LEDs (charging status indicator); 1 push button; Real time clock with a 12 mm coin cell baterry; Rechargable power bank.
3.7. PCB Layout

With PCB, the environmental effects are minimized, which helps to provide stable and fast modulated infrared signal for ID detection. The all-in-one design can also provide small sized board for users easily and comfortably carry.

The PCB Design of EasyID sensor board was designed with Altium. We take the advantages of both sides space, which can minimize the board size as much as possible. For the location of the components, we first consider the position of the receivers and the transmitters. Then we fix them and try to put the rest according to their schematics. After laying out all components, we start routing. To avoid the collision of routing, we first design the longest one, then a shorter one and leave the shortest and ground to the end. Testing the short circuit is important before finishing the routing. The schematic of the add-on board is attached in the appendix. The PCB layout can be seen in 3.15 3.16.

Figure 3.15: Add-on Board Front View

Figure 3.16: Add-on Board Back View
Overall, the dimension of the add-on board is $26.50 \times 56.75$ mm with 2 layers. Together with the open source platform, EasyID is $85 \times 56 \times 25$ mm without a plastic enclosure. There are 26 GPIO pins available on board. The prototype is weighted 158g. The total price of the EasyID is 64 euros. (Figure 3.17)
Based on the hardware circuit design, the software configuration is given. According to the selection on SBC as well as the motion sensors and their circuit design, the operating system and its software development can be determined. Basic software set up is implemented in this project, for the ID recognition, inertial sensor readout, and data storage. The chapter starts with the system architecture and followed by each hardware control, especially the algorithm involved in infrared communication for identification detection. Finally, full synchronized data storage will be described at the end.
4. **EasyID: Software Design**

4.1. **System Architecture**

The architecture overview of the EasyID is shown in the figure 4.1. The EasyID is mainly composed of three main functional components, namely IR transmitter, receiver; IMU; and RTC. Each of them is driven by the Raspberry Pi through the GPIO control. When the infrared receiver receives any infrared signals with the same software configuration, the decoding process is triggered on. Together with the highly accurate global time stamp provided by the RTC, the IDs and human movement information can be stored into a micro SD card in the device. If the receiver doesn’t receive anything for one second, the infrared transmitter starts to send its own ID to others. Data captured through the 9 DoFs IMU is forwarded before storing into the file, according to their data source (accelerometer, gyroscope, and compass).

4.2. **ID Recognition**

Figure 4.2 describes the ID detection software architecture. This section will start with the modulation and encoding of the infrared transmitter and then the demodulation and decoding of the infrared receiver. How signals travel through different chan-
4.2.1. Encoding

According to previous chapters, one effective way to eliminate the noise and interference of ambient infrared sources is signal modulation. The transmitted signal is modulated using a carrier frequency of 38 kHz, because it is a common carrier frequency and suitable for our goal. The IR LED is made to oscillate at this frequency for the duration of the pulse. The light signals are pulse width modulated and contained in the 38 kHz frequency. There are two ways to modulate infrared LEDs at 38 kHz: one is using the oscilloscope to drive PWM (Pulse Width Modulation), and the other is an oscillator. For 38 kHz modulation, we use 555 oscillators instead of the software package in Linux called LIRC (Linux Infrared Remote Control), because:

The LIRC package doesn’t provide enough software development space for users. The 38 kHz oscillation is out of the range. We can observe that the light signals get distorted from square waveform to triangle waveform, with LIRC package. The reason is even if Raspberry Pi has a fast processor, using Python to drive hardware always have a limitation on processing time (and it is the same with all other signal board computers). Another reason is that the Linux is not a real-time system, which may interrupt the user program. It is not accurate enough to make sure an accurate pulse length use the software only. The pulse length is what distinguishes the logical 1 from logical 0, so we can’t sacrifice the accurate pulse for the privilege of using software packages. Meanwhile, the only disadvantage of using hardware would be power dissipation and take up of extra physical space. The TCL555 oscillator can already provide low power solution (< 750 mW) with small physical size (5×6 mm).

The 38 kHz carrier signal is pulsed on and off to generate the bit stream for the IR communication. There are already some existing protocols that can be used to realize infrared communication between nodes, which are compared in chapter 2. We design a new protocol (see Figure 4.3), for identification detection among devices with the same configuration. Using existing protocols can cause noises such as commands from infrared remote controllers. The leading pulse is a 3.6 ms pulse with a 1 ms space, the log-
ical "1" is 1.5ms with a 1 ms space and finally, the logical "0" is 0.6ms with a 1ms space. The characteristics of the transmission are listed as below.

1. Pulse width modulation
2. 38kHz carrier frequency
3. Bit time is 2.5ms or 1.6ms
4. 8-bit address

The design of the IR protocol is inspired by the NEC protocol, which bursts at a carrier frequency of 38 kHz. Among all common Vishy infrared receiver support protocol as RC5/RC6 code, Sony code, Sharp code and NEC code, NEC is selected because it is simple, the pulse width modulation can be stable. It starts the transmission using a leader code, a burst with a length of 9 ms, followed by a pause of 4.5 ms and then the data word. Our protocol is a slightly different from NEC code. The total length of the binary string is 8 bits; allowing a scale up to 256 devices. The logical 1 was designed as three times larger than the logical 0 (1.5ms to 0.5 ms), which makes the differences between 1s and 0s large and the total time small. Space is twice larger than pulses in logical 0. But the 0.5ms pulse length was not stable during the experiments because of the inaccurate Python package time.sleep. As a result, the lowest possible value is 0.6ms according to my test results. So we keep the rest as it should be and make the pulse length in logical 0 a bit longer. The coding scheme of the encoding part is simple, we use a Python package RPi.GPIO control on a Raspberry Pi. The GPIO.HIGH generates pulses and GPIO.LOW provides spaces in between two pulses. With designed pulse length, the IR LEDs can send signals that contain ID information to others through
4.2.2. DECODING

With the same configuration, the decoder of the infrared receiver reads out the pulse length and translates them into binary strings.

The whole program can be divided into parts: i). read out raw data; ii). transfer into binary strings; and iii). calculate the ID number. It is better to separate the three parts during coding because it is faster and can reduce a lot of errors. After reading out the raw data, the program starts with the decoding part. In this way, a faster (1.67 times) ID detection can be established because abundant data won't need any processing to recognize a certain ID information.

Reading out the raw data: pin 18 and pin 4 (BCM) are used for capturing signal change during the reception. The pulse, space and their time in between can be captured through RPi.GPIO package and datetime package. All raw data can be stored into a list called "command" 4 times per second, for later translations and calculation of the signals. During the raw data reading out, a simple low pass filter is implemented to filter out high-frequency noises signal. Meanwhile, the sampling frequency is defined as a variable, which can be changed according to scale scenario.

Translation into binary strings: The most difficult part for decoding is to translate the raw data into binary strings. Figure 4.4 explains how the program evolves. The program starts with a decision if a pulse is received, if true, then it is decided if this pulse is the leading pulse (3.6 ms)? If yes, read-out of the binary string is started and the next bits are read-out and decoded as 1’s or 0’s. When we reach up to 8 bits data, the program produces a binary string, which marks the end of the program. The pulse length difference can be used to separate the leading pulse, logical 1 and logical 0. To make sure that exactly eight bits address is decoded, a counter variable is defined. The preamble variable is defined to capture the beginning of the infrared signals.
Calculation of ID number: the calculation from binary string into ID number requires a function. The last bit is calculated as the ID information. If the binary string is $[1,0,1,0,1,0,1,0]$, the ID equals to 85 4.2.2.

$$ID = 1+0 \times 2 + 1 \times 4 + 0 \times 8 + 1 \times 16 + 0 \times 32 + 1 \times 64 \ (4.1)$$

4.2.3. **Data Link**

While the encoding and decoding of the infrared nodes can make a full functioning physical layer configuration, several problems can happen during infrared communication between two nodes or among three nodes. The first one is that the device keeps receiving its own ID. The second one occurs when two nodes or more transmit their ID simultaneously, collisions might happen. Collisions cause a higher error rate. An algorithm is designed to solve the above problems.

The algorithm is shown in Figure 4.5, which aims at separating the transmitting and receiving channels. If the receiver doesn’t receive anything from other devices, the timer starts. If one second passes, then it starts to transmit. Within this second, the receiver keeps listening. This protocol ensures that the transmission and reception of infrared signals are two channels, which eliminates the chance of seeing itself as well as of interfering with transmissions from other devices.

Is_received is a function to decide if the decoding part should be activated. Instead of reading out a series of infrared pulses and spaces, this function only checks if there
is a transition from space to pulse (or the other way around). If yes, the decoder can be activated. With this function, decoding and encoding can speed up to 3 times than without.

Because the designed PCB board has two infrared receivers to cover a full angle of 180 degrees, it is important to control two receivers respectively. The timer is used to check the time passed if the receiver doesn’t receive anything. Only if neither receiver A nor receiver B receives anything, the timer can check if a second transmission delay has passed. The software configuration of both receivers is the same, but they can’t be on the same channel. They should run in parallel, without interrupting each other.

4.3. MOTION SENSOR

An overview of the connection with software and hardware for the motion capture part can be seen in Figure 4.6. The RTIMULib is a C++ and Python library that makes it easy to use 9-dof and 10-dof IMUs with embedded Linux systems (especially for Raspberry Pi). By default, RTIMULib will try to discover 9DoFs sensor on the I2C bus. If the I2C bus is occupied, the sensor will be detected. Especially, this software package provides magnetometer calibration, which is in the file RTIMULibCal. Change of measurement range can be realized through the changing of software setup in the RTIMULib installed in Raspberry Pi. The sensors measurement ranges can be easily scaled up and down through changing the set up in RTIMULib.h.

Python file sense_hat.py in Python 2.7 can be adjusted and adapted to the user space for MEMS sensor of the EasyID. The amended the codes are available in the Appendix. In the library, it provides the access to the IMU sensor (LSM9DS1), with functions to initialization the IMU, and to return the raw data from the IMU (accelerometer, gyroscope, and compass respectively).

Python file mems.py is a program made in this project to read out data from GPIO and print out the required data with the function given by sense_hat.py in Python 2.7. The sampling frequency is set up to 20Hz currently and can be changed according to different scales. The raw data from gyroscope, accelerometer and the compass in the IMU is read out separately but recorded as a signal sequence and then stored together with the time stamp as data log.

4.4. DATA SYNCHRONIZATION AND STORAGE

Data synchronization is realized through the on board real time clock. The real time clock (RTC) use the I2C bus, and setting up the module requires the support for RTC through adding a device tree overlay. The command that we used to capture and print out the current time is from the python package time in Python 2.7.

All ID information and motion sensor raw data together with their time frame can be stored into files. With a certain function to store any printed out data into the file, all printed data with the global time stamp can be stored into a CSV (Comma-Separated Values) format. The motion sensor data and ID information data are stored separately because the sampling time of them are different. Every a certain hour/minute, the file will be closed and a new file will be created to allow data write in it. The name of the file will be the time when the file is opened. All data will be stored into the micro SD Card
and can be directly downloaded from the Raspberry Pi after the measurements with the SCP (secure copy) command.
5

**EasyID: Evaluation**

It is essential to validate all the functional components as well as the whole device of EasyID after finishing the hardware and software design. This chapter presents the performance evaluation of the EasyID badge. Its performance will be evaluated first with each module separately. Then the whole sensor package will be validated. For each module, it starts with the evaluation of the ID detection module, followed by the motion sensing module. The battery life of the EasyID is tested with two different types of rechargeable power bank. Communication among three devices is recorded also. Finally, it will show how the EasyID badge is going to be implemented on the human body.
The system performance of EasyID is validated first with respect to the ID detection, and subsequently the motion sensor is tested to see if the data collection and data storage are working properly. Finally, the practical EasyID implementation is described at the end.

5.1. ID Detection Module

The experimental setup for evaluation the ID detection is shown in Figure 5.1. In this section, the evaluation is designed to see the performance of ID detection between nodes in different cases. Two EasyID devices are facing each other, with different distances in between, or different opening angles. Both of them can be controlled by a personal computer (PC). All of them are connected to the same local network, via a router (D-link DIR-615). We only discuss the infrared communication between two devices in this section. Both of the devices can be receiver node and transmitter node.

5.1.1. Static Scenario

The performance of EasyID’s ID detection is first evaluated under static scenarios, i.e., fixed environmental factors, as shown in Figure 5.2. The default environmental setup is located at the room HB 15.160, with ceiling lights kept off and windows shuttered. We use IR LEDs TSAL6200 for the static scenario.

Data quality versus distance: Data quality is defined as the possibility of the correct ID detection during a certain period of time. Since there are two symmetrical devices, the average values of the two devices' data quality are calculated. The data quality of EasyID is tested by varying the distance between the two devices. The distance is defined as the perpendicular distance between them. The two devices are directly facing to each other, free from the impact of incidence and irradiation angles. The experiment is designed as this, we put one device fixed at one point, and move the other one with a constant speed (100mm/s). For each plot, the experiment lasts for 5 seconds. In this way, the distance in between the two devices is increasing with time. EasyID reports the instantaneous ID detection every 0.05 second (20Hz). The data is recorded in files called distance1.csv, distance2.csv and distance3.csv stored on the SD card and then copied to be analyzed. Through Figure 5.3, we can see that EasyID maintains its peak ID detection.
rate at up to 2.5 m with the default setup. After this distance, it starts to drop and then drops dramatically because the received signal strength is not sufficient for the receiver to decode the signal.

**Data quality versus angle:** The performance of EasyID is also measured while varying the incidence angle from LED to the photodiode. The dependent variable is the same with the previous experiment, i.e. data quality, but independent variable is irradiation angles. During the experiments, the perpendicular distance (1 m) from LED to the photodiode is constant, with the variation of incidence angle. The incidence angle is defined as the acute angle between the perpendicular distance and relative distance (after one of the device moves around the 1 m circle respect to the other). The data recorded as one EasyID is fixed at a certain point, the other one is moving around the 1m circle with a constant speed of 5 degrees per second till it is facing to the side of the fixed one. The
sampling rate of the test is 20Hz. From the results, we can observe that the data quality starts to drop from 40 degrees (Figure 5.4). The reason why the data quality drops not so dramatically as the distance relationship one could be that the three LEDs complement each other. The overlap angles of three LEDs can reduce the speed of data quality decreasing. If one of the LEDs reaches to its lowest radiance at a certain angle, the other two still can send effective information.

**Comparison with other carrier frequency:** The maximum sensing distance of the 38kHz carrier frequency and the 36kHz carrier frequency is different. Figure 5.5 is the screenshot of the oscilloscope with 38kHz modulated transmitted signal. With 36kHz carrier frequency, the maximum possible range is only 0.6 meters. With 38kHz frequency, the maximum possible distance can reach up to 3.3 meters. With a 38kHz infrared receiver, a 38kHz modulated transmitter ensures the receiver receives maximal signals. The responsivity of a 38kHz infrared receiver with a 36kHz modulated infrared signal sending drops by 40%.

**Comparison with different LEDs:** Three LED modules are selected to be quantified and compared according to their behaviors during the experiments, namely VLSB3940, TSAL6200 and TSAL6400. VLSB3940 and TSAL6400 are selected to make improvements for the prototype. VLSB3940 is a 3mm IR LED module, which has the similar calculated sensing range as the TSAL6200. TSAL6400 is proven to have a wider opening angle and a longer sensing distance according to chapter 3. The procedure for the experiments is the same as the data quality versus distance and angle ones. The maximum sensing range is measured for 95% data quality during the signal communication. The experimental results do not perfectly match the Datasheets. There is a trade-off between the sensing distance and the opening angles. The selection of LEDs should be based on the implementation. The test results have shown biases. VLSB3940 has the advantages of the largest opening angle, with a small physical size. Therefore it can be selected for the crowded dense scenario. But the 5 mm IR LEDs can enable a longer sensing distance, suitable for large scale scenario. TSAL6400 is overall the optimal solution. It has a long

![Figure 5.5: The Snapshot of the Transmitted Signals without Encoding](image)
sensing distance as well as a wide opening angle.

<table>
<thead>
<tr>
<th></th>
<th>VLSB3940 3mm</th>
<th>TSAL6200 5mm</th>
<th>TSAL6400 5mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum distance (meters)</td>
<td>2.1</td>
<td>3.3</td>
<td>3.5</td>
</tr>
<tr>
<td>Maximum angle (degrees)</td>
<td>160</td>
<td>100</td>
<td>120</td>
</tr>
</tbody>
</table>

### 5.1.2. Dynamic Scenario

This section presents the system performance of EasyID under the dynamic scenario. Important factors could be the change of ambient light, remote controller interfering and the change of room layout.

**Ambient light:** The change of ambient light may be a potential factor, that influences the infrared communication, and causes problems during ID detection. The change of ambient light during research could be a shuttering or unshuttering the window of the room, or turning the room’s ceiling lights on or off, within a certain room.

**External remote controller:** An external remote controller can affect the possibility of sensing correct ID if the receiver senses the signals from the remote controller. The remote controller selected in this experiment is the controller for the projector in the meeting room (EWI 1 HB 15.040). The remote controller blinks once every five seconds towards the receiving photodiode. Results of the data quality are analyzed to see the effect of a remote controller.

**Different rooms:** Change of room layout may cause the reflection of infrared and change the infrared radiance during the measurements. Since the most two common real venues are restaurants or theaters, instead of placing the devices only at EWI HB 15.160, where the most of the experiments are done, different rooms are selected to mimic different scenarios that may occur during the real life implementation of the devices. One is a meeting room HB 15.040, which mimics the situation in theaters (Figure 5.7) and the other one is at the cafeteria at EWI (Figure 5.8), for the situation in the restaurant.

Scenario 1 is the control group, with a shuttered window, an off ceiling light, and no extra interfering sources. Scenario 2 is the case when the window is shuttered but ceiling light on. Scenario 3 is when the window is unshuttered with an off ceiling light. Scenario 4 is a blinking remote controller scenario. During Scenario 5, we move the devices to the meeting room. Scenario 6 is when the devices are in the restaurant at EWI. With a constant distance of 0.5 meters between two devices. the devices are directly facing each other. The data quality is calculated by the number of correct ID detection divided by the total numbers of detection.

From Table 5.2, we can see ceiling light is not a major factor that causes errors in ID detection (2%). However, an unshuttered window can introduce a spate of noise, which causes the difficulty for the infrared receiver (photodiode) to distinguish the infrared signal from noise. One way is to increase the ID information signal by amplification, the other way is to reduce the noise from outside, to increase the Signal-to-Noise Ratio (SNR). However, the 38kHz carrier frequency and 940nm wavelength have already filtered out most of the noise, there is little space for decreasing the noise. At the same time,

---

1 EWI: Building 36, Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) of TU Delft
adding an amplifier will cause the extra power consumption and reduce the bandwidth. Even if there is an amplifier, the noise from sunlight is still too large for the LEDs. As a result, for future research, it is strongly recommended to do the measurements without the facing directly to the sunlight. A remote controller can cause errors of ID detection also. Even if the receiver is designed for 38kHz IR signal, the tolerance allows it to receive some 36kHz or 40kHz modulated IR signals. We observe a relatively high data quality during the meeting room scenario. The reason why meeting room (theater scenario) has a relatively high data transmission accuracy is it is relatively dark compared to the restaurant and do not have that many reflections according to the layout.

<table>
<thead>
<tr>
<th>Senario</th>
<th>Description</th>
<th>1 min</th>
<th>5 min</th>
<th>10 min</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Senario 1</td>
<td>Control group</td>
<td>98.6%</td>
<td>95.5%</td>
<td>96.7%</td>
<td>96.9%</td>
</tr>
<tr>
<td>Senario 2</td>
<td>Ceiling light on</td>
<td>96.2%</td>
<td>92.1%</td>
<td>96.0%</td>
<td>94.8%</td>
</tr>
<tr>
<td>Senario 3</td>
<td>Window opened</td>
<td>56.3%</td>
<td>58.9%</td>
<td>58.0%</td>
<td>57.7%</td>
</tr>
<tr>
<td>Senario 4</td>
<td>Remote controller blinked</td>
<td>65.5%</td>
<td>60.1%</td>
<td>60.2%</td>
<td>61.9%</td>
</tr>
<tr>
<td>Senario 5</td>
<td>Meeting room Fig 5.7</td>
<td>88.2%</td>
<td>81.1%</td>
<td>81.7%</td>
<td>83.6%</td>
</tr>
<tr>
<td>Senario 6</td>
<td>Restaurant Fig 5.8</td>
<td>76.0%</td>
<td>75.2%</td>
<td>60.9%</td>
<td>70.7%</td>
</tr>
</tbody>
</table>
5.2. MOTION DETECTION MODULE

The main challenge to using motion sensors in activity recognition is the bias and noises their measurements or signals suffer from. They can mislead an activity recognition module from recognizing to which activity the sensor signals correspond. In this section, we will validate the motion sensor with zero level detection as well as moving objects response experiments. All experiments are repeated three times to eliminate the measurement errors.

5.2.1. ZERO LEVEL

Zero level motion detection quantifies the variation of the sensor when there is no input. If the device is motionless it feels about 1g (0.98) of acceleration at the z axis. For most low-cost inertial navigation applications, the only aiding source that can be used
to limit the velocity errors (and thus reduce the position error divergence rate) is the zero-velocity updates. The box plots are plotted to better indicate the variability of the measured variables. Figure 5.9 5.10 and 5.11 describe the variation with zero motion on the accelerometers. In the x axes list three repeated experiments. The box shows the 25 and 75 percentile of the dataset. The recording time is 1 min for each experiment. The sampling rate is 20 Hz, so the size of the dataset is 1200 for 1 minute. Y axes of the box plot represent the measured values. The red lines in the box plot are medians of the recorded data. According to the figures, recorded acceleration on x, z axes are only showing the red lines instead of boxes (except y axis Figure 5.10). Since the acceleration is not a continuous variable, so it is easier to see some extreme outlier 2.

For the gyroscope, we can refer to Figures 5.12 5.13 and 5.14, which show the offsets from the gyroscope measurements. The upper line represents the maximum values of the total datasets. Any values that are more than three times the height of the boxes will be considered as extreme values. We can see some extreme values during the zero level detection on gyroscope and compass Figure 5.15. But they happen randomly, not in every experiment, which could be measurement errors or occasional errors. The offsets for three axes depend on the measurement range. For a ±2000dps measurement range, the zero gauss level is claimed ±30dps (degrees per second). With zero movement, the variation is ranging from 0.05dps - 0.15dps for a 245dps range.

5.2.2. MOVING OBJECTS
To evaluate the motion sensor sensitivity to the physical change, a constant velocity (20 dps) was applied to the device to see if the motion sensor can capture the changes. The measured angular velocity is 21 dps (Figure 5.16). Figure 5.17 shows how the device responds at a known acceleration (0.2g). The blue lines are measured raw data, the red line is the plotted desired data. From Figure 5.16 and Figure 5.17 we can see the mea-

---

2 Extreme outlier: Any values that are more than three times the height of the boxes.
sured data don’t perfectly match the expected data, because of the following reasons:

1. The speed of the moving objects is not constant. It is hard to move the device with a constant speed only by hands. The sensor is of high sensitivity, a little shaking during the movement can cause unexpected values. The limitation of the experimental instruments is lacking an automatic moving machine.

2. Noise. The sensitive IMU easily captures any noise from the environment and occasion during experiments. Small physical changes from the table or the device may cause the measured data not desired. Proper smoothing algorithms should be implemented after capturing the raw data.

Figure 5.18 and 5.19 indicate the measured orientations compared to iPhone S6.
Both of the measurements last for 5 seconds. The sampling frequency is 20Hz. Compared to iPhone 6s, EasyID can measure a higher accurate data, the bias between them are small (0.87%).

Before real life implementation of the EasyID, good calibration with the more advanced experimental setup is desired. Calibration on the human body is necessary, to understand the factors that determine the variations.
5.2. **Motion Detection Module**

The reliability test is important to define how stable the IMU and motion capture can be with the change of time and the operations on the devices. It is designed as the com-

---

**Figure 5.16: Gyroscope Measurements vs Standard**

**Figure 5.17: Accelerometer Measurements vs Standard**

### 5.2.3. **Reliability**

The reliability test is important to define how stable the IMU and motion capture can be with the change of time and the operations on the devices. It is designed as the com-
Comparison of acceleration, angular velocity, the orientation and navigation values between some environmental changes. Three environmental changes are i). change of room temperature; ii). change of location; and iii). shaking of the device. The variation is calculated as the difference between average values divided by the average values under the first condition. The data recordings last 1 minute. Details about the calculation steps are listed in the appendix. For the room temperature, two values are 25°C and 30°C, and no significant difference between the change in temperature is observed. With the change of location, from EWI HB 15.160 to EWI cafeteria the, we also don’t observe any significant differences between measurements. After series of sudden shake, however, a 7.9% variation changes can be observed once at z axis accelerometer data. This gives us the guidance of using the device without sudden shakes.
5.3. Sensor Badge

After testing the main modules one by one, experiments are done to validate the sensor badge as a device and prototype and indicate the behaviors during the real life experiments.

Battery Life: The experiment is aimed at testing the battery life of the EasyID. Two available power banks are randomly selected. Figure 5.20. The larger one is the Lepow Moonstone with a capacity of 6000 mAh, the smaller one is Lab31 with 2200 mAh capacity. For the small one, it has a dimension of $25 \times 27 \times 88$ mm and weights only 110 gram. With Lab31 power bank, the EasyID can be powered for approximately 7 hours. Lepow is $82 \times 80 \times 9$ mm and weights about 163g. With Lepow (the EasyID can be powered up to 20 hours with all functional components working.

Three Devices: Communication among three devices is possible with the designed software and algorithms. By changing the waiting times in between the time where the receivers don't receive anything and time where IR signals send, three EasyID sensor packages can receive each others’ ID without any collisions within one local network. Figure 5.21 indicates the experimental setup. All of them are connected to the same local network, and each of them can be recognized by the console (my PC) based on their address. The distance is no longer defined as the perpendicular distance anymore. But the central point distance. The average distance in between the devices is 0.5 meters and the viewing angle is approximately 60 degrees. The experiments are operated in the restaurant because it has a wider space to arrange three devices at the same table. The data quality for three devices is 57.2% (EasyID 1 address 129), 61.8% (EasyID 2 address 139) and 69.3% (EasyID 3 address 149). The wait time of the three devices is set as 0.5s (EasyID 3), 0.6s (EasyID 2), 0.7s (EasyID 1). This experiment has proven that three devices can detect each others’ IDs without collision and don’t sense its own ID. Infrared communication protocols for a larger group can be an interesting thesis topic for embedded software or telecommunication students.

EasyID on the body: Figure 5.22 shows the EasyID as a badge, with a rechargeable battery and a strip. On-body tests are aimed at testing if the EasyID can work properly on the human body. The EasyIDs are hanging on the chests of the two users, with the infrared LEDs and receivers facing to the environment. The ID detection data and
motion sensing data can be printed and stored as data logs. Figure 5.23 is the screenshot of the demo video during on-body tests. The video is linked at the end of the appendix. The two participants are facing to each other. Then one of the participants is turning towards the wall during the recording, which causes ID information disappears on the other EasyID’s user interface. Then, he turns back to the participant, and the signal comes back. Then both participants walk towards to each other and do a handshake. We notice some changes on the IMU logs during the movements of participants, as well as during one of the participants is turning around. The recorded data includes handshake, detection of ID, re-detection of ID and moving body with a global time stamp. To see if this device is user-friendly, I invite 20 volunteers (10 male and 10 female) between 22 to 55 years old to perform the experiment of users’ comfort of the device. 85% of the participants have claimed a satisfying experience wearing EasyID. One of them complains about the rechargeable battery being separate with the device, two are confused about the hang a device on their chest.
Figure 5.23: Wearable EasyID on body
6

CONCLUSION
The design of an open source platform based sensor badge is important for future research on the role of social behaviors during social events. This thesis proposed a low-cost and small size customized data acquisition and storage sensor badge, called EasyID that provides enough space for future upgrade and extension. Its key modules include i) motion sensor; ii) infrared transmitter; and iii) infrared receiver. EasyID is based on the low cost and small sized open source platform Raspberry Pi, with a Quad Core 1.2GHz 64bit CPU, 1GB RAM and the on-board Bluetooth 4.1. The motion sensor is 9 DoFs inertial sensor LSM9DS1 from STMicroelectronics, to capture body movement. The infrared receivers and transmitters are PWM modulated pairs with a 38kHz carrier frequency. The 38kHz carrier frequency is generated through a low-power CMOS integrated circuit TCL555. The real time clock is a highly accurate DS3231M from Maxim Integrated, for the global time stamp. Experimental results have validated the stable and real-time data collection of EasyID. The sensing angle of the EasyID is 120 degrees and the sensing distance 3.5 meters with a 95% of the accuracy of successful ID detection. The motion sensor can be scaled up to 16g, and the sampling rate is set typically to 20Hz. The maximum sampling frequency is 20kHz. The EasyID is weighted 158g includes the raspberry pi a 2200mAh rechargeable battery and the dimension is 85 × 56 × 25mm. The total price includes the raspberry pi is 64 euros.

This thesis has following characteristics, that make it more advanced than the previous sensor packages.

1. **Customized PCB** EasyID consists of an open source platform and a add-on Printed circuit board (PCB) with infrared communication module (transmitter, receiver) for ID detection, a MEMS 9 DoFs inertial sensor for human motion sensing, a real time clock for accurate time stamp, two indicators, and one push button for user interfaces to establish a real-time data collection sensor package at a high level of sensitivity.

2. **Open Source Platform** EasyID is designed on an open source platform Raspberry Pi 3 Model B and the add-on board is accessible by RASPBERRY PI 2 MODEL B, RASPBERRY PI MODEL B+, and all RASPBERRY PI 1 MODEL. It can be easily upgraded, manufactured, which enables the possible extension of the sensor package.

3. **Python Programming** To control the EasyID, a high-level programming language, Python is used, which enhances code readability. Compared to C/C++ language, python is slower, which causes the problem while modulating infrared transmitters. A hardware solution is designed to solve the problem through a 555 oscillator. At the same time, the open source software developed to drive the circuits in this thesis can be shared with other similar implementations.

4. **Low Cost and Small Size** The overall cost of the manufactured EasyID is 64 euros. The dimension of EasyID is 85 × 56 × 25mm.

The proof-of-concept of wearable sensor badge has been presented and validated in this thesis. For the future work, I plan to extend its functionalities from two directions:
i) Bluetooth. Current EasyID can detect any facing device with the same hardware and software configuration. An improvement of the EasyID is to sense nearby devices using radio-frequency based techniques, such as the BLE, which is already on the Raspberry Pi 3. And with Bluetooth, it is possible to enable the communication between devices; ii) Enclosure. To protect the EasyID from damage during the experiments in the future, it is recommended to design a suitable enclosure for the EasyID, for the wearability of users.
REFERENCES


Full Requirements

- **Functional requirement:**
  1. The device shall measure body motion data (acceleration, orientation, and velocity).
  2. The device shall detect any facing or nearby devices with the same configurations.
  3. The device shall recognize the IDs detected devices.
  4. The device shall be able to store all the data in a fully synchronized fashion, using a global time-stamp as the reference.

- **Non-functional requirement:**
  1. The device must have the capacity to scale to dense crowded scenarios.
  2. The device should be easy for users to carry. (Maximum: 200g in weight and 100*100*50mm for dimension)
  3. The device should be initially low cost. (Each<100 USD) 9. The device should require minimal maintenance.
  4. The start up time of the device when powering should be less than 30 seconds.
  5. The accelerometers of the users should be able to be measured. (Minimum: a range of ±2g at a resolution of 0.1g and a maximum deviation of ±0.1g).
  6. The linear velocity of the users should be able to be measured. (Minimum: a range of ±10m/s at a resolution of 0.1m/s and a maximum deviation of ±0.1m/s).
  7. The angle velocity of the users should be able to be measured. (Minimum: a range of ±90°/s at a resolution of ±1°/s and a maximum deviation of ±5°/s).
  8. The configuration capabilities of measurement would be provided using a software.
  9. The sampling rate of body motion data should be higher than 20 Hz.
  10. The device should be able to detect another device within a range of 3 meters.
  11. The device should recognize the IDs of nearby (50-300cm) devices.
12. The device should be able to store all the data in a fully synchronized fashion, using a global time-stamp as the reference.

13. The global time-stamp should be accessed at a resolution of 0.1s and the maximum deviation of ±0.1s.

14. The storage capacity of the device must be large enough to store all the sensing data in a working day. (2GB)

15. The data must be able to be downloaded from the devices at the end of a recording.

16. The wireless device must be able to run for 6-8 hours on the same battery.

17. The device should have low power consumptions with the maximum 1W.

18. The device should be able to stay functioning at the temperature range from 0-40 °C.

19. The lifetime of the device should be longer than one year.

20. Sufficient I/O capabilities (>10) should be left open for hardware extension.

21. Serial peripheral interfaces as I2C and SPI should be left open on boards for the potential measurement of bio signals.

22. The device should provide means for programming and debug the software.

23. The device needs to be evaluated in a system configuration of at least 32 devices.

24. The device should be able to robust to mechanical shock. (2g)

25. The device must not cause any severe damage to itself.

26. The device must not cause any health issues to users around it.

27. A third-party device should not read the device.
# The Comparison of Open Source Platform

Comparison of Raspberry Pi, BeagleBone, OLinuXino, Orange Pi and ODROID

<table>
<thead>
<tr>
<th>Name</th>
<th>RASPBERRY PI 3 MODEL B</th>
<th>RASPBERRY PI 2 MODEL B</th>
<th>BeagleBone Black</th>
<th>BeagleBone Green</th>
<th>BeagleBone Green Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>86 x 56 x 17mm</td>
<td>86 x 53 x 4.5 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>1.6 oz</td>
<td>1.4 oz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture</td>
<td>64/32-bit ARMv8</td>
<td>32-bit ARMv7</td>
<td>32-bit ARMv7-A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoC</td>
<td>Broadcom BCM2836</td>
<td>Texas Instruments AM3359 (AM335x)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>1.2GHz 64-bit quad-core ARM® Cortex®-A7</td>
<td>900 MHz 32-bit quad-core ARM® Cortex®-A7</td>
<td>1 GHz 32-Bit Sitara™ ARM® Cortex®-A8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processor Speed</td>
<td>1.2GHz</td>
<td>900 MHz</td>
<td>1GHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock speed (Max.)</td>
<td>400 MHz</td>
<td>250 MHz</td>
<td>606MHz</td>
<td>800MHz</td>
<td></td>
</tr>
<tr>
<td>Cache</td>
<td>16KB of L1 and 512KB shared L2</td>
<td>16KB of L1 and 256 KB shared L2</td>
<td>32KB of L1 Instruction &amp; 32KB of Data Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDRAM</td>
<td>1GB LPDDR2</td>
<td>512MB DDR3</td>
<td>512MB DDR3</td>
<td>512MB DDR3-1866</td>
<td></td>
</tr>
<tr>
<td>EEPROM</td>
<td>N/A</td>
<td>4KB</td>
<td></td>
<td>4KB</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>Broadcom VideoCore IV</td>
<td>PowerVR SGX530</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power consumption*</td>
<td>800 mA @ 5V</td>
<td>210 – 460 mA @ 5V</td>
<td>210 – 460 mA @ 5V</td>
<td>Low power mode (ELP)</td>
<td></td>
</tr>
<tr>
<td>I/O</td>
<td>40 GPIO</td>
<td>65 (digital) + 7 (analogue)</td>
<td>65 (digital) + 7 (analogue) + 2x Grove connectors*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB 2.0</td>
<td>4</td>
<td>1+1 (Standard A host port + mini B device port)</td>
<td>4 +1 (Micro USB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video out</td>
<td>HDMI, composite video</td>
<td>MicroHDMI, cape add-ons</td>
<td>Via HDMI Cape only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio out</td>
<td>Multi-Channel HD Audio over HDMI, Analog Stereo from 3.5mm Headphone Jack</td>
<td>MicroHDMI, cape add-ons</td>
<td>Via A2DP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low-level peripherals</td>
<td>17× GPIO, UART, I2C, 2x SPI</td>
<td>4x UART, 8x PWM, LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion</td>
<td>8x PWM, I2C and UART, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On Board Internet</td>
<td>10/100mb Ethernet 802.11n wireless, Bluetooth 4.1</td>
<td>10/100mb Ethernet</td>
<td>10/100mb Ethernet</td>
<td>10/100mb Ethernet</td>
<td>802.11n wireless, Bluetooth 4.1</td>
</tr>
<tr>
<td>Real time Clock</td>
<td>RTC modules possible</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-board storage</td>
<td>Micro SD card slot</td>
<td>8-bit eMMC (Rev B: 2 GB, Rev C: 4 GB); microSD card slot</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage capacity</td>
<td>64GB</td>
<td>32GB</td>
<td>4GB (provided)</td>
<td>64GB</td>
<td></td>
</tr>
<tr>
<td>Power source</td>
<td>2 sources MicroUSB or GPIO header 5V1, 2.5A</td>
<td>4 sources USB port on a PC; A 5VDC 1A power supply; USB connector; Expansion connectors</td>
<td>3 sources A USB port on a PC; A power supply with a USB connector; Expansion connectors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>$35</td>
<td>$32</td>
<td>$55</td>
<td>$39</td>
<td>$45</td>
</tr>
<tr>
<td>Name</td>
<td>A13-OLinXino-WIFI</td>
<td>A10-OLinXino-LIME-4GB</td>
<td>Orange Pi Plus2</td>
<td>ODROID-C2</td>
<td>ODROID-XU4</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Size</td>
<td>100 x 85 x 17 mm</td>
<td>84x 60 x 17 mm</td>
<td>108 x 60 x 17 mm</td>
<td>85x 56 x 17 mm</td>
<td>98 x 74 x 29 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>1.4 oz</td>
<td>1.4 oz</td>
<td>1.4 oz</td>
<td>1.4 oz</td>
<td>1.4 oz</td>
</tr>
<tr>
<td>Architecture</td>
<td>32-bit ARMv7-A</td>
<td>32-bit ARMv7-A</td>
<td>32-bit ARMv7-A</td>
<td>64-bit ARMv8-A</td>
<td>ARMv7</td>
</tr>
<tr>
<td>SoC</td>
<td>Allwinner A13</td>
<td>Allwinner A10</td>
<td>Allwinner H3</td>
<td>Armlogic S905</td>
<td>Samsung Exynos5 Octa</td>
</tr>
<tr>
<td>CPU</td>
<td>1GHz 32-bit ARM®</td>
<td>1GHz 32-bit ARM®</td>
<td>1.2GHz 64-bit</td>
<td>2GHz 64-bit Quad-core ARM® - A53</td>
<td>2GHz Quad - ARM® Cortex® - A15 and 1.3GHz Quad - Cortex® - A7</td>
</tr>
<tr>
<td></td>
<td>Cortex® - A8</td>
<td>Cortex® - A8</td>
<td>Quad-core ARM®</td>
<td>- A7</td>
<td></td>
</tr>
<tr>
<td>Processor Speed</td>
<td>1GHz</td>
<td>1GHz</td>
<td>2GHz</td>
<td>2GHz</td>
<td></td>
</tr>
<tr>
<td>Clock speed (Up to)</td>
<td>553MHz</td>
<td>553MHz</td>
<td>696MHz</td>
<td>333MHz</td>
<td>933MHz</td>
</tr>
<tr>
<td>SDRAM</td>
<td>512MB DDR3</td>
<td>1GB DDR3</td>
<td>2 GB DDR3</td>
<td>2GB LPDDR3</td>
<td></td>
</tr>
<tr>
<td>EEPROM</td>
<td>2 KB</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>GPU</td>
<td>ARM Mali400</td>
<td>ARM Mali400</td>
<td>3 x ARM Mali450</td>
<td>M1i-628 MP6</td>
<td></td>
</tr>
<tr>
<td>Power consumption</td>
<td>Stand by: 380 mA @ 6V [23]</td>
<td>Stand by: 210 mA @ 5V</td>
<td>1 A @ 5V</td>
<td>350 ~ 880 mA @ 5V</td>
<td>600mA @ 5V</td>
</tr>
<tr>
<td>I/O</td>
<td>74 GPIO UEXT connector*</td>
<td>40 GPIO (Ground)</td>
<td>40 (digital) + 7</td>
<td>60 GPIO</td>
<td></td>
</tr>
<tr>
<td>USB 2.0</td>
<td>3 + 1 USB hosts, (3 available for users 1 for WIFI)</td>
<td>2 + 1 (USB-OTG)</td>
<td>4 + 1</td>
<td>4 + 1</td>
<td></td>
</tr>
<tr>
<td>Video out</td>
<td>VGA (800 x 600 resolution) LCD interface</td>
<td>HDMI, Integrated CVBS</td>
<td>HDMI 2.0 4K-60 Hz</td>
<td>HDMI</td>
<td></td>
</tr>
<tr>
<td>Audio out</td>
<td>3.5 mm Jack, HDMI</td>
<td>3.5 mm Jack, HDMI</td>
<td>3.5 mm Jack, HDMI</td>
<td>HDMI/ I2S</td>
<td>I2S</td>
</tr>
<tr>
<td>Low-level</td>
<td>3x 12C; 2x UARTs;</td>
<td>3x 12C; 2x UARTs;</td>
<td>UART, compatible with Raspberry Pi B+</td>
<td>UART; 12C; ADC; I2S</td>
<td></td>
</tr>
<tr>
<td>peripherals</td>
<td>LCD</td>
<td>LCD</td>
<td></td>
<td>I2S</td>
<td></td>
</tr>
<tr>
<td>On Board Internet</td>
<td>WIFI RTL8188CU</td>
<td>10/100/1000M Ethernet,</td>
<td>10/100/1000M Ethernet</td>
<td>USB IEEE 802.11b/g/n 1T1R WLAN with Antenna</td>
<td></td>
</tr>
<tr>
<td></td>
<td>802.11n 150Mbit Ethernet, WIFI RTL8189ETV, IEEE 802.11 b/g/n</td>
<td>10/100/1000M Ethernet</td>
<td>802.11 b/g/n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real time clock</td>
<td>On board</td>
<td>RTC modules</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>On-board</td>
<td>4GB NAND flash, microSD card slot</td>
<td>4GB NAND flash, (Not included) microSD card slot</td>
<td>8GB eMMC Flash TF card (Max. 64GB) / MMC card slot, up to 2T on 2.5 SATA disk</td>
<td>8GB eMMC 5.0 HS4000 Flash Storage; MicroSD Card Slot</td>
<td></td>
</tr>
<tr>
<td>storage</td>
<td>4GB</td>
<td>8GB</td>
<td>64GB</td>
<td>32GB</td>
<td>32GB</td>
</tr>
<tr>
<td>Power source</td>
<td>3 sources: 6-16VDC input power supply (Noise immune design); 1 USB OTG; Battery option and connector</td>
<td>4 sources 5VDC input power supply (Noise immune design) 1 USB OTG; Battery option and connector SATA Connector</td>
<td>2 Sources DC input USB</td>
<td>2 Sources DC input power supply; 1 USB OTG; 2 Sources DC input; USB</td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Linux</td>
<td>Linux</td>
<td>Kali Linux</td>
<td>Linux (might not stable)</td>
<td>Linux</td>
</tr>
<tr>
<td>Price</td>
<td>$49</td>
<td>$39</td>
<td>$39</td>
<td>$40</td>
<td>$74</td>
</tr>
</tbody>
</table>

*UEXT is a board to board connector connector, which supports three serial communication interfaces - I2C, SPI and RS232. There is also 3.3V line and GND. It is a great way to expand the features of the development boards you already have. The customer can choose which new feature he wants to expand.

*Grove connector: UART and I2C connectors for SeedStudio Grove modules

*Power consumption depends on the tasks on board. Most of the values in this table are at idle, which make the minimal limits. And the maximal power consumption still needs experiments.
The Schematic of the Add-on Board

Figure 6.1: Schematic of EasyID
Calculation of Reliability

1. Calculate the average of measurement (Accelerometer, Gyro, Compass) under case 1 before 1 minute T1
2. Calculate the average of measurement (Accelerometer, Gyro, Compass) under case 1 after 1 minute T2
3. Calculate the difference between the two averages D1
4. The variation V1 is difference D divided by T1
5. Calculate the average of measurement (Accelerometer, Gyro, Compass) under case 2 before 1 minute T3
6. Calculate the average of measurement (Accelerometer, Gyro, Compass) under case 2 after 1 minute T4
7. Calculate the difference between the two averages D2
8. The variation V2 is difference D2 divided by T3
9. Compare V1 and V2. If the variation of V1 V2 is greater than 0.05, we consider this as a significant change.
EasyID Demo

The demo video of this project is available here: https://www.dropbox.com/sh/3qus0i5qm6lqw7/AAC0X7y28MnIIH1_nu_VP7P-a?dl=0.

For a better viewing experience, please download it and then watch it offline.