

DESIGN AND DEVELOPMENT OF A FALL DETECTION DEVICE WITH INFRARED RECEIVING CAPABILITIES

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering

By

RAMZI AMMARI

B.S. Hashemite University, 2003

M.S. Egr., Wright State University, 2006

2011

Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

August 15, 2011

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Ramzi Ammari ENTITLED Design and Development of A Fall Detection Device With Infrared Receiving Capabilities. BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Computer Engineering.

Jack Jean, Ph.D.

Thesis Director

Mateen Rizki, Ph.D.

Chair
Department of Computer
Engineering
College of Engineering and
Computer Science

Committee on
Final Examination

Jack Jean, Ph.D.

Yong Pei, Ph.D.

Meilin Liu, Ph.D.

Andrew Hsu, Ph.D.
Dean, School of Graduate Studies

Abstract

Ramzi, Ammari. M.S.C.E, Department of Computer Science and Engineering, Wright State University, 2011. *Design and Development of a Fall detection device with Infrared Receiving capabilities.*

Fall related injuries are the leading cause of death and hospitalization among the elderly. Falls among older people become a major problem facing hospitals and nursing homes. In this study we put an effort to design a wireless device capable of detecting falls with the hope that this study will provide a path towards better healthcare monitoring and better independent living for the elderly.

In this project I showed how the fall detection device can be interfaced with different systems to achieve functionality without adding extra cost. For seniors who prefer to stay at their homes and live independently, the device can communicate with their smart phone to request help if needed. For hospitals and nursing homes, an infrared receiver and infrared signals decoding algorithms were implemented to interface with FastFind software to keep track of the location of the residents who fall or request help. There is also an option of having a live video feed from the specific room where the fall was detected.

Contents

1	Introduction	1
1.1	Fall Detection	1
1.1.1	Causes and Prevention of Falling In the Elderly	2
1.2	Thesis Motivation	5
1.3	Thesis Objectives	6
2	Hardware Development	7
2.1	Hardware Architecture	7
2.2	Phase 1:Arduino Uno Development Board	7
2.2.1	Phase One Main Components	9
2.3	Phase 2: Printed circuit board	12
2.3.1	Hardware Components	13
2.3.2	Phase 2 Design Considerations	14
2.3.3	Hardware Design	15
2.3.4	Infrared Receiver Hardware Connection	16
2.3.5	Panic Button Hardware Connection	17
2.3.6	Bluetooth and Accelerometer Hardware Connections	17
3	Processing sensor's data	20
3.1	Decoding the infrared signal	20
3.1.1	Sony Protocol Example	21
3.2	Infrared Receiver Code Implementation	21
3.3	Converting Accelerometer values into G units	24
3.3.1	Fall Detection Algorithm	26
3.4	The Choice of The Wireless Sensor	26
3.4.1	Comparison Between Zigbee and Bluetooth Low Energy	28

4	Code Development and System Integration	30
4.1	Uploading The Code to The Microcontroller	32
4.2	Interfacing with HyperTerminal	33
4.3	Interfacing with Matlab (Graphical Representation)	34
4.4	Independent living for the elderly	36
4.4.1	Interfacing with a smart phone	36
4.4.2	Interfacing With The Android Phone	38
4.4.3	Changing The Baud Rate of the Bluetooth Modem	38
4.4.4	Android Application	40
4.5	Interfacing With FastFind	45
4.5.1	The Tracking Process	46
4.5.2	FastFind Fall Detection Application	48
5	Conclusion	52
5.1	Summary	52
5.2	Recommendations And Future Work	53
6	Attachments: Schematic	55
7	Bibliography	57

List of Figures

2.1	The shield that contains the components of hardware phase 1	8
2.2	The device after mounting the shield on top of the Arduino board . . .	8
2.3	Arduino Uno development board	9
2.4	Voltage distribution for main components in phase 1 hardware	10
2.5	ADXL335 pin connections to the Arduino board	11
2.6	Second phase hardware building block	12
2.7	PCB components with Bluetooth module	13
2.8	PCB components without the Bluetooth module	14
2.9	Pin connections between the microcontroller and the IR	16
2.10	Panic button connection	17
2.11	Accelerometer/Bluetooth hardware connections	18
2.12	Accelerometer /Bluetooth modules placement on PCB board	19
3.1	Pulse width encoding for Sony protocol	21
3.2	Infrared signal Decoding operation	23
3.3	The key characteristics of Zigbee, Wi-Fi and Bluetooth [9]	27
4.1	Fall detection device interfaces	31
4.2	HyperTerminal interfaces	33
4.3	Matlab code flowchart	35
4.4	Matlab graphical representation for the accelerometer and IR values .	36
4.5	Pin connections between the Arduino Uno and the Bluetooth modem	39
4.6	Bluetooth module Baudrate default settings	40
4.7	Interfacing with smart phone to request help	41
4.8	pairing the android app with the fall detection device	42
4.9	Fall detection on the Android smart phone	43
4.10	Request help through panic button on the Android smart phone . . .	44

4.11	Tracking process for the FastFind software	46
4.12	Tracking process for the FastFind software	49
4.13	FastFind fall detection application with camera view	51
5.1	Gyroscope/accelerometer combo board [13]	54

Acknowledgement

I would like to thank everyone who supported me and were involved in one way or another in the preparation of this thesis.

First and foremost, I would like to express my sincerest gratitude to my advisor Dr. Jack Jean for giving me the opportunity to work on the project and for his support, enthusiasm, and immense knowledge which helped me during the course of this project.

I'd like to thank the committee members, Dr. Yong Pei and Dr. Meilin Liu, for giving me valuable suggestions and advice on this thesis. My sincere gratitude to Dr. Blair A. Rowley who gave me important guidance during my first steps in this project and for his insightful comments.

I would like to thank all my friends who have helped me out through this journey, I would also like to thank Jianlong Chai for his invaluable assistance.

Last but not the least, I am ever-thankful to my family for their understanding, encouragement and endless love.

1 Introduction

1.1 Fall Detection

For elderly people unexpected falls can cause significant injuries. Fall related injuries are the leading cause of injury-related deaths among adults 65 years old and older; these injuries are also the number one reason for emergency room visits [1]. An average nursing home with 100 beds will experience anywhere from 100 to 200 falls annually, with typical nursing home patient experiencing a fall 2.6 times per year [2].

Elderly people aged 65 and older experience the majority of these falls. Falls among older people become one of the most common and often critical problems facing hospitals and health care providers. For that reason previous research has approached this problem to come up with a mechanism to detect falls.

Common fall detection methods include a detection feature and a trigger to issue an emergency call to provide medical care. Fall detection is based on algorithms that use sensor values such as accelerometers, gyroscopes, and air pressure sensors. These sensors cannot fully differentiate between falls and ADL (Activity of Daily living); This leads to a high percentage of false alarms which is the main reason for the failure of these systems, thus they are mainly rejected for commercial use by monitoring services [3].

In our design, we tried to tackle the problem and find solutions for assistant living for the elderly by adding the room location capability to identify the location of the fall, send this information to a software that triggers a wireless camera and stream a live video feed of that room. By doing so we can identify the fall first and send medical care to the patient which is an optimal solution for an assisted living situa-

tion where you have a large number of elderly people as well as large number of rooms.

Independent living for the elderly is also an important aspect of this project. The high cost of these systems for personal use made it difficult to be commercially successful. In this project different wireless technologies are introduced and the reason of choosing one particular technology over others is discussed.

1.1.1 Causes and Prevention of Falling In the Elderly

Understanding the causes of falling in the elderly help us find different ways to prevent falls. The fall detection device's main function is to detect a fall but it would be useful if we can prevent falls from happening in the first place, or developing a mechanism to protect the elderly if a fall happens. For example deploying a cushion which can serve as protection in case of a fall, although this mechanism is not implemented in this project, it is recommended for future work.

Younger people falls more than elderly people. However, the injuries they suffer are less sever usually a small bruise or discomfort, unlike the elderly who often ends up with broken bones and suffer more severe injuries, so it's important to avoid falls.

The causes of falls are known as risk factors, usually more falls occur when there are more risk factors involved. Being aware of these factors can help understand the risks involved and can help you find ways to prevent them or make the changes necessary to provide protection. I will briefly discuss the main risk factors of falls among older adults:

1. Osteoporosis: it is the condition in which bones become less resistant to stress because of calcium loss which causes the bone structure to deteriorate; osteoporosis is the main cause of fractures in older adults, especially among women.

- Prevention tips :

- Taking a bone density test to find out if you have osteoporosis.
- Regular exercise can reduce the likelihood of bone fractures in people with osteoporosis
- Eat or drink sufficient calcium.

2. Vision changes: Our vision changes as we get older; things become hard to see in dark areas, increased sensitivity to glare and the field of vision narrows down. Cataracts and glaucoma alter older people's depth perception and visual acuity. All these changes increase the chances of falls and limit the ability to move in the environment you living in.

- Prevention tips:

- Make regular visits to the eye doctor to make necessary changes to your eye glasses.
- Keep the entrances and hallways clear from objects and obstacles that you may not see while walking.
- Do not ignore any eye conditions you have such as Cataracts and glaucoma
- Clean eye glasses often to improve visibility.

3. Balance problems and medications: reduced muscle strength and changes in posture leads to balance problems. The human balance uses the feedback from

the interaction between multiple sensory, motor and integrative systems to keep the balance. When these sensors and systems don't send the right feedback balance issues occur. Medications contribute also to falls by reducing mental alertness and impacting postural stability.

- Preventing Tips:

- Be aware of the side effects of the medications you are taking, and consult with your physician about the effects of combining multiple medications.
- Don't drink alcohol while taking medications; alcohol is known to negatively interact with medications.
- Exercise more to strengthen your posture.

4. Environmental risks and hazards: There are many risks and hazards in the indoor and outdoor environment, such as loose rugs, unstable furniture, slippery surfaces, steep stairs, uneven pathways and the list goes on. There are many ways to prevent falls and make the surrounding environments safer.

- Preventing tips :

- Fall proof your environment.
- Always be aware of your surroundings.
- Keep walk areas clear of clutter, rocks and tools.
- Install grab bars and Add non-skid mats to baths.

1.2 Thesis Motivation

In many situations healthcare requires constant measurements of patient vitals, but due to either large number of patients or small staff sizes, these measurements are only taken over long intervals or not monitored at all. In assisted living situations, a nurse may visit the residents only a few times, but could provide better patient diagnostics or even emergency care in critical situations if they could provide more frequent monitoring.

To improve the diagnostic ability and emergence response, the need for wireless sensor nodes for healthcare monitoring is extremely important. There are many vital parameters such as temperature and situational events like falls that can aid in better healthcare monitoring. Being able to implement this device and have a better idea about its issues and constraints, gives a baseline set of data that can be applied to other vital parameters that can be measured as technology grows and the number of sensor types increase.

Another important issue in healthcare monitoring is the location of patients. In an assisted living situation or a hospital where there are hundreds of rooms, its important to know the location of the fall in order to send immediate help to that location. Infrared (IR) transmitters or beacons placed in strategic locations can be programmed to broadcast their location ID; with the use of IR receivers these signals can be decoded and the position of the patient can be determined with minimum processor resources without adding any extra cost.

1.3 Thesis Objectives

In chapter 1, the importance of continuous healthcare monitoring as well as causes of fall and ways to prevent these fall are introduced. In chapter 2, Hardware architecture for two different prototypes is discussed, the components are presented and the hardware design is implemented. Chapter 3 describes the method used to convert the accelerometer values into G units, the implementation of infrared signal decoding and the reason for choosing Bluetooth as the wireless sensor of choice for this project, also in this chapter different wireless technologies are compared and discussed.

Chapter 4 describes how the device can be interfaced with different hardware/software to achieve the goals we needed to accomplish in this project, which includes interfacing with HyperTerminal, Matlab, Android smart phone, and FastFind software. Finally, Chapter 5 provides the conclusion for this project as well as recommendations for future work and ways to add more functionalities and improvements to the current device.

2 Hardware Development

2.1 Hardware Architecture

In this project, I worked on developing a fall detection device that can work and interface with different systems to have a solution for problems that other devices have. It was important to prioritise what needed to be done in the limited. Therefore so I decided to use an Arduino development board as an initial design to save time on designing a circuit board from scratch, and go through the process of printing the circuit board and soldering, without knowing if the device will function properly or the challenges associated with this project.

The development board has all pins exposed and the sensors can be easily attached and tested, This initial prototype is phase 1 of this project. The second phase include the desig of a compact printed circuit board, soldering all the components, adding a panic and a switch button, and designing the board to fit an enclosure with a compartment for the 9V battery.

2.2 Phase 1:Arduino Uno Development Board

The initial goal for this device was to design a prototype with sensors and wireless capability. The device needed to be portable, have a long battery life, light in weight, and small in size, so it is easy and comfortable for the elderly to wear it or carry it around.

The components were made as a shield that can be easily mounted on top of the Arduino development board to keep all the components on the inside of the device, to provide better protection against damage that may occur to these components in case of a fall.

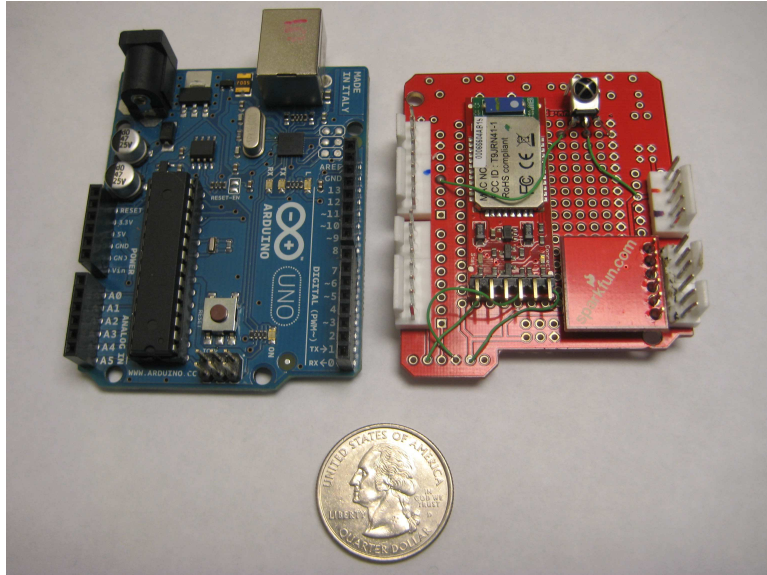


Figure 2.1: The shield that contains the components of hardware phase 1

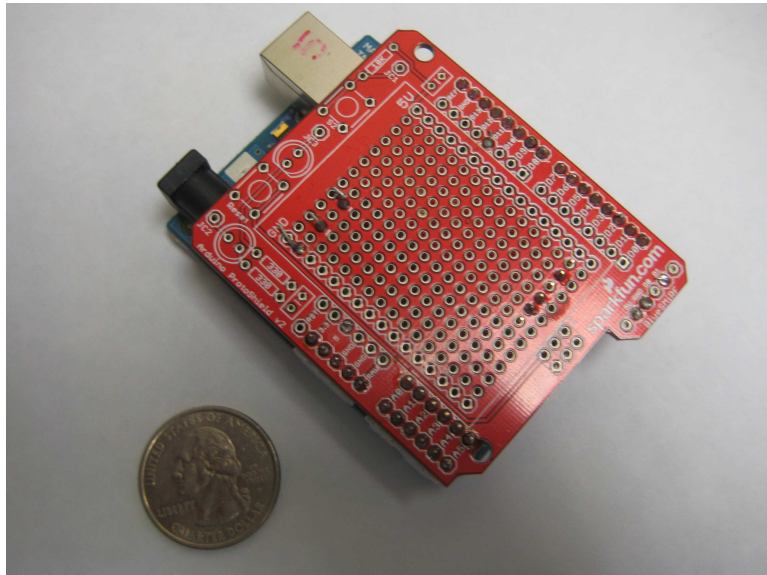


Figure 2.2: The device after mounting the shield on top of the Arduino board

2.2.1 Phase One Main Components

1. **Arduino uno development board:** For the first Phase of the project I used the Arduino Uno board which is the latest revision of the basic Arduino USB board. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.

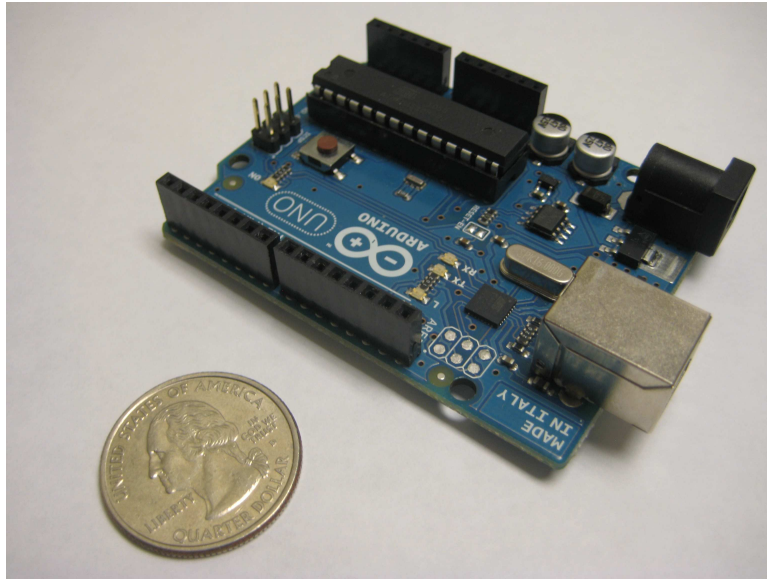


Figure 2.3: Arduino Uno development board

The Arduino Uno is a micro-controller board based on the Atmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the micro-controller [4].

The board can operate on an external supply of 6 to 20 volts. The voltage regulator gives the regulated power supply used to power the microcontroller and other components on the board, while a 3.3V generated by the on-board regulator powers up the accelerometer component as shown in the diagram.

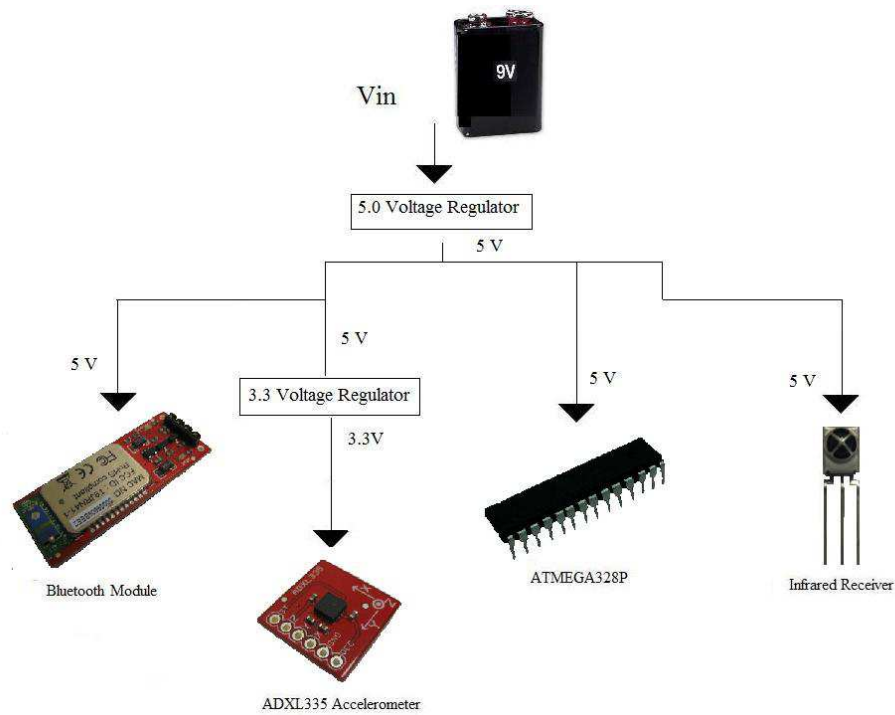


Figure 2.4: Voltage distribution for main components in phase 1 hardware

2. **ADXL335 Accelerometer** The accelerometer used is the ADXL335 which is a small, thin, low power complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of $+ -3$ g mini-mum. It contains a poly-silicon surface-micro machined sensor and signal conditioning circuitry to implement open-loop acceleration measurement architecture.[5] The output signals are analog voltages that are proportional to acceleration. The Arduino analog pins will be used for the X, Y, Z axis pins while the other 2 pins will be used as VCC and GND.

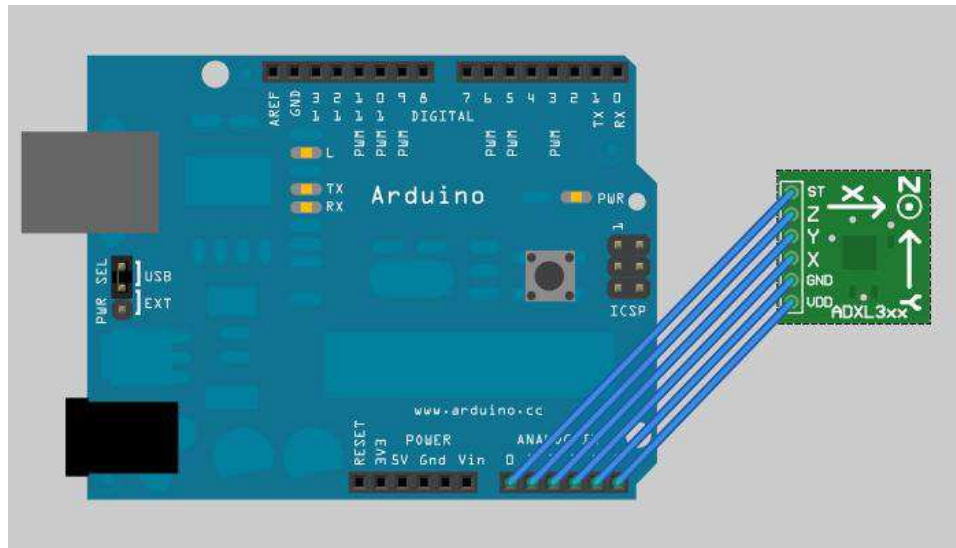


Figure 2.5: ADXL335 pin connections to the Arduino board

3. **36KHz Infrared Receiver:** GP1UE270RKVF Infrared Receiver was used, which is an IC IR Detector 36KHZ infrared receiver, it has three pins; VCC, GND, and output pin. The output pin is connected to pin 11 of the Arduino board. VCC was connected to the 5V pin on the Arduino board since its operating voltage ranges between 2.7 and 5.5V.
4. **Bluesmirf Gold Bluetooth:** The Bluetooth module used in both hardware phases, is the Bluesmirf gold Bluetooth which is an FCC Approved Class 1 Bluetooth Radio Modem delivers up to 3 Mbps data rate for distances up to 100m. The Bluetooth module has 6 pins, 3 of these pins will be used to connect to the Arduino; VCC, GND, and RX.

The TX pin is not used since we are only transmitting and not receiving data back to the device. The other 2 pins CTS and RTS are shorted together when it is needed to put the Bluetooth in setup mode to change the default baud rate if needed.

2.3 Phase 2: Printed circuit board

The initial prototype from phase 1 helped me see what is needed to be modified in the design, what changes needed to be done, power consumption issues, and how to make the device smaller and compact without affecting the functionality. The second phase is using all the knowledge and lessons learned from phase 1, to design a printed circuit board that has lower power consumption and a smaller size. I also took a step further by designing my board to fit in an enclosure to be presented as a product that can be commercially distributed in the market.

Operating Environment:

- Input Voltage Range 9.0 to 12.0 volts.
- input Voltage used 9.0 Volts.
- Typical transmission range for the Bluetooth module : 200 feet

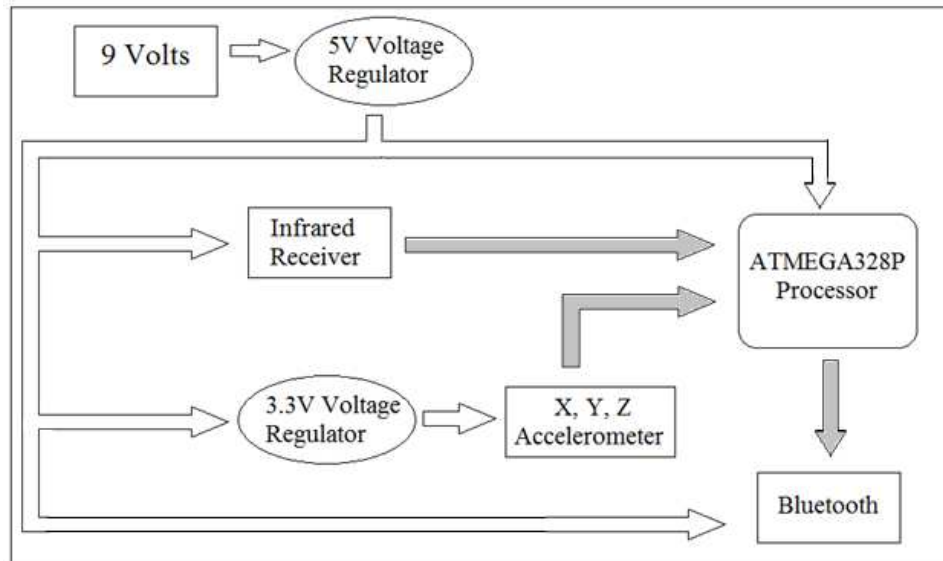


Figure 2.6: Second phase hardware building block

2.3.1 Hardware Components

Surface mount components as well as through a hole components were used. I tried to use small components to keep the board small, but still large enough to be manually soldered.. Figure 2.7 and Figure 2.8 show the components used in building the circuit board.

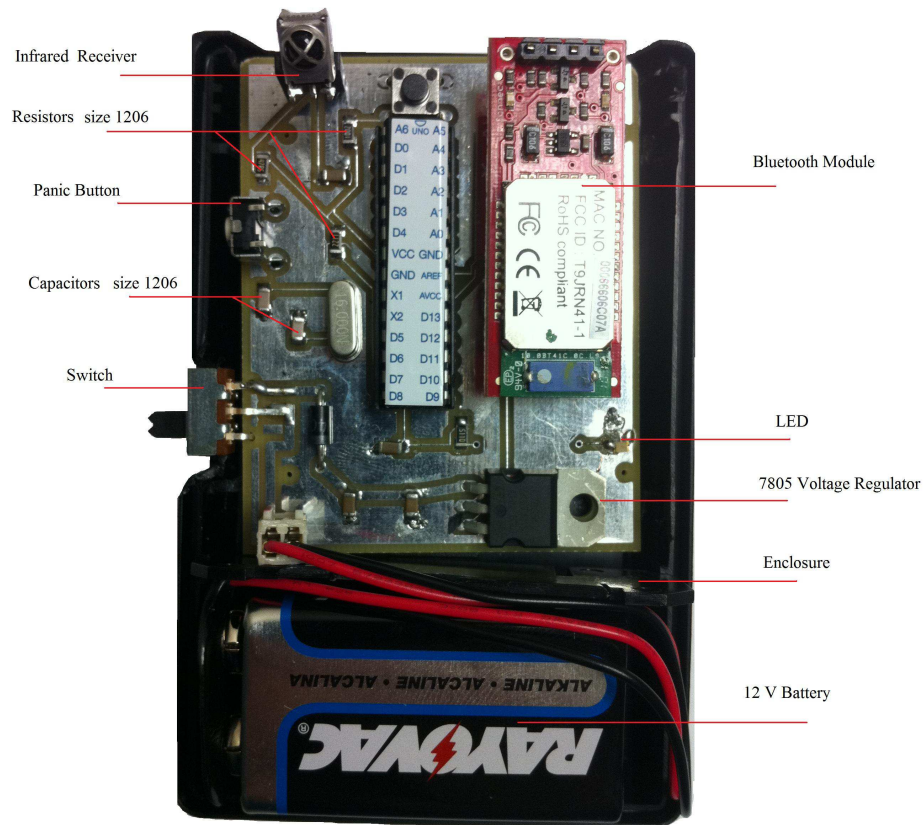


Figure 2.7: PCB components with Bluetooth module

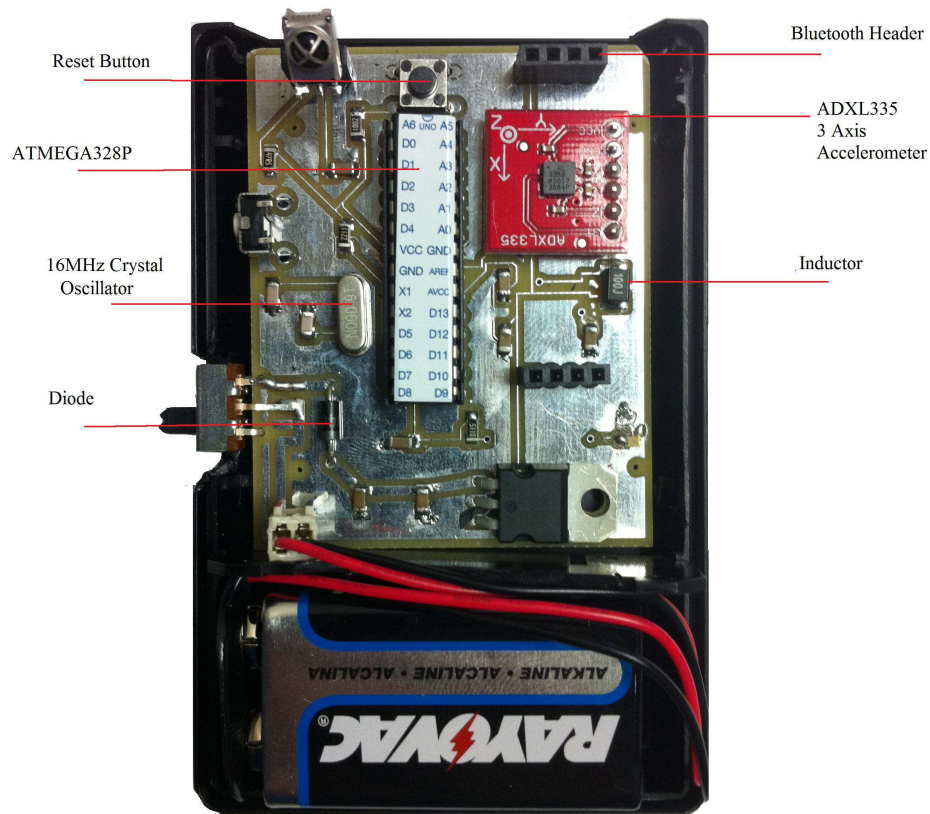


Figure 2.8: PCB components without the Bluetooth module

2.3.2 Phase 2 Design Considerations

- Power Distribution:** The device has different components, some needs a 5 V supply voltage and one component uses a 3.3 V. In order to supply the right voltage for each component, voltage regulators were used. Voltage regulator's job is to reduce the input voltage to the lower voltage required for the component and automatically maintain a constant voltage level. Typically voltage regulators are surrounded by heat sinks since they generate significant heat.
- Circuit decoupling:** all of the modules should have some form of decoupling if it does not exist already on the breakout board. A decoupling capacitor is a capacitor used to decouple one part of an electrical network (circuit) from an-

other. Noise caused by other circuit elements is shunted through the capacitor, reducing the effect they have on the rest of the circuit. Electrolytic and ceramic capacitors were avoided at the output of the voltage regulators since they could cause instability.

- **Diode usage:** Diodes allow electricity to flow in only one direction, which restrict the flow of current. When electricity flows in one direction, it is more or less unrestricted. When the flow is reversed, however, the diode stops it. The diode is used as a reverse battery protection
- **Good Grounding:** Grounding is fundamental to the operation of many circuits. The more ground path, the lower the impedance is. In this design, both top and bottom layers of the board has wide ground layers separating wires, holes and components.
- **ADC supply voltage:** AVCC is the supply voltage pin for the A/D Converter. It should be connected to VCC through a low-pass filter. A good design separates them on the board so when you have a mixed mode circuit or board you can run your power lines separately to avoid crosstalk noise, since digital switching noise may affect your analog signals and vice versa. From the data sheet they suggest a low pass filter when using the AVCC power from the digital power, and this allows the dc through while blocking the high frequency digital surges as demanded from the switching circuits.

2.3.3 Hardware Design

Usually a hardware design goes through multiple stages before ending up in the final printed circuit board design. Because of time constraints; I went from schematic to printed circuit board without going through breadboard prototyping. Eagle Cad was used to design the schematic and layout for the printed circuit board. The dimensions

of the board are 65 x 52 mm and are chosen to fit an enclosure that has a 12 Volts battery compartment.

2.3.4 Infrared Receiver Hardware Connection

The infrared receiver has 3 pins; ground, VCC and output pin. Figure 2.9 shows the pin connections between the infrared receiver and the ATMEGA328P microcontroller.

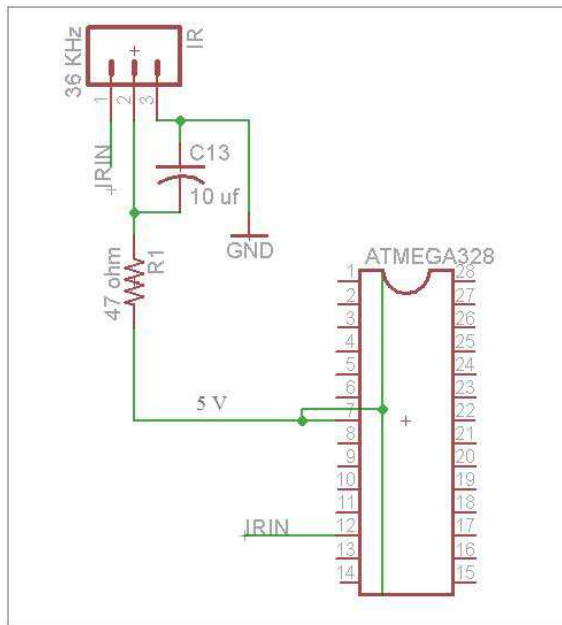


Figure 2.9: Pin connections between the microcontroller and the IR

The infrared receiver has input operating voltage range between 2.7 and 5.5V, therefore pin 7 from the microcontroller is used as a supply voltage for the infrared receiver since it is connected to the 5V power supply. The output pin of the IR is connected to the microcontroller digital pin 12, while the third pin is connected to ground. These infrared receivers have filters, output shapers and circuits to decode the received signal.

2.3.5 Panic Button Hardware Connection

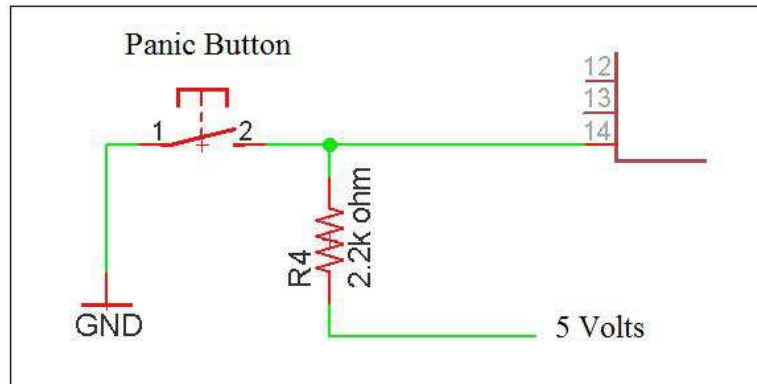


Figure 2.10: Panic button connection

A Panic button was implemented so that it sends a command through Bluetooth once its pressed. The way this works is by connecting one lead of the panic button to two things; first is a 5 Volts through a 2.2 K Ω resistor, and the other to one of the pins on the microcontroller which is pin 14 in this case. Digital read is done on pin 14 to check if its high (5 Volts) or low (Zero Volts), while the other lead is connected to ground. When the panic button is not pressed, the voltage will go through the resistor to pin 14 and reads high, on the other hand when the push button is pressed, the electricity will take the path with the least resistance and go to ground and so pin 14 will read zero voltage. The microcontroller is programmed to send a certain command if zero voltage is read on pin 14 (panic button is pressed).

2.3.6 Bluetooth and Accelerometer Hardware Connections

The accelerometer is connected the same way it was connected in phase one. The only difference is that we had to substitute for the 3.3V onboard regulator that is present in the Arduino board. A 3.3 voltage regulator was used to reduce the supply voltage of 5V from pin 28 on the microcontroller to 3.3V. Decoupling circuits were also used to have a steady input to the accelerometer. I used a 4 pin header for the 4 pins of the Bluetooth module (VCC, GND, TX and RX), the Bluetooth can be easily

mounted on the header and supported by another 4 pin header to hold the end of the Bluetooth module.

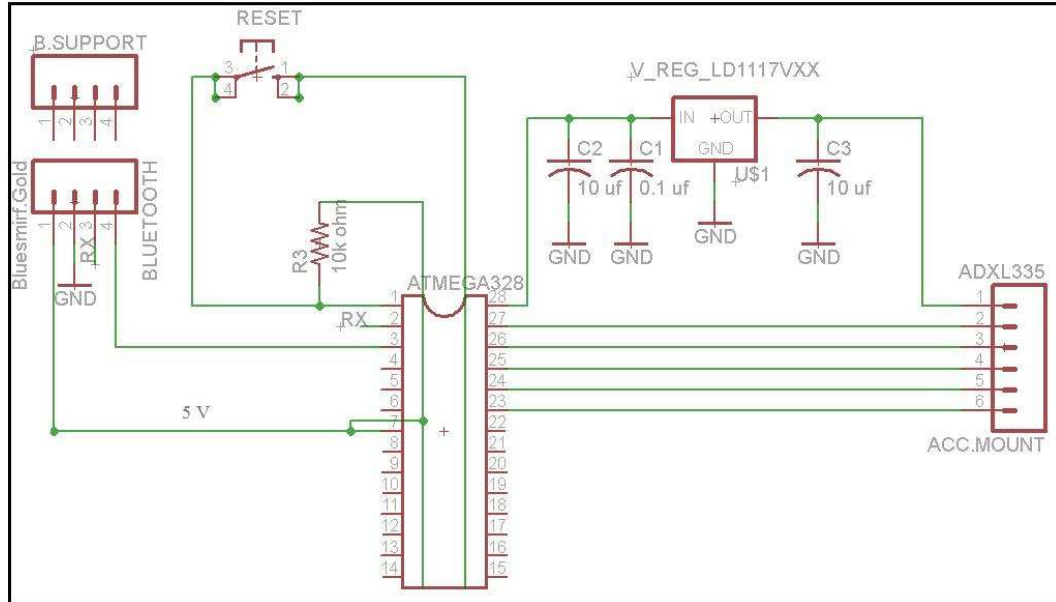


Figure 2.11: Accelerometer/Bluetooth hardware connections

There are two reasons for this design:

- The headers will raise the Bluetooth module, and keep a space underneath it to place the accelerometer, so that two modules can be placed on top of each other to save space.
- The Bluetooth can be detached from the header, so that the same header can be used by the USB to Serial breakout board to achieve the serial communication and upload the code to the microcontroller since the two boards use the same pins.

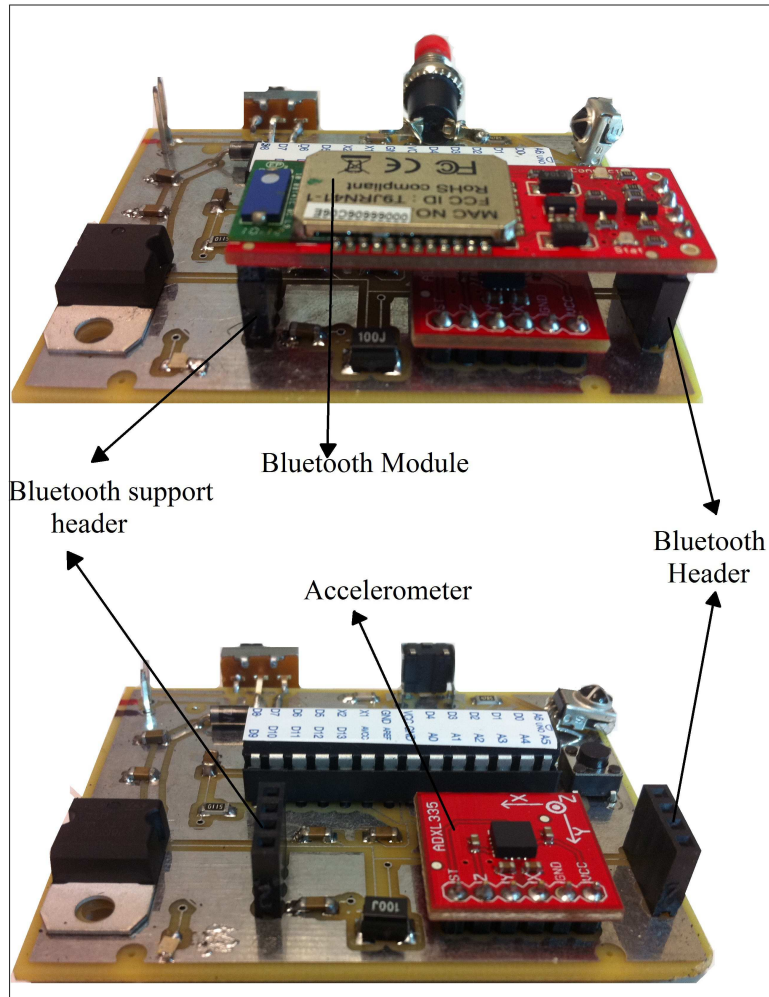


Figure 2.12: Accelerometer /Bluetooth modules placement on PCB board

3 Processing sensor's data

Infra-Red is a normal light with a particular color. Humans cannot see this color because its wave length of 950nm which is below the visible spectrum. An IR remote works by turning the LED on and off in a particular pattern. However, to prevent interference from IR sources such as sunlight or lights, the LED is not turned on steadily, but is turned on and off at a modulation frequency (typically 36, 38, or 40KHz)[6]. Modulation is the answer to make our signal stand out above the noise. With modulation we make the IR light source blink at a particular frequency. The IR receiver will be tuned to that frequency, so it can ignore everything else. You can think of this blinking as attracting the receiver's attention. Humans also notice the blinking of yellow lights at construction sites instantly, even in bright daylight.[7]

In serial communication we usually speak of 'marks' and 'spaces', which are decoded into 1-s and 0-s depends on the protocol that's being used. If you know the encoding algorithm, you can determine the code value. For example, televisions decode the infrared signal received from the remote control into sequence of 1-s and 0-s; for each button pressed you get a unique code that can be used to trigger a specific function such as turning the volume up or down.

3.1 Decoding the infrared signal

Even though many TV remotes and other transmitters that send an infrared signal on the same frequency for example 37KHZ, which means they pulse 37 thousand times per second when sending 1 and do not pulse when sending zero. The signals transmitted from one source do not control other receivers, since the receivers use different protocols to decode these signals such as: Sony protocol, JVC, Panasonic, RC5 protocol and many more. These protocols become standards and are adopted

to be used to decode the infrared signals. To demonstrate the process of decoding of an infrared signal, I will use the Sony protocol as an example.

3.1.1 Sony Protocol Example

The transmitted signal has a frequency of 40 KHz, if we use an oscilloscope to look at the signal received from a Sony remote control, we will see a train of High and Low pulses that varies in length. If we zoom in at the High signal which represents a Mark, we will see thousands of small ON/OFF signals that transition very rapidly throughout the Mark signal. The code transmitted consists of 12 bits (5 bit address that represents the command and a 7 bit address represents the address). The signal starts with a header that consists of 2400 microseconds ON signal followed by 600 microseconds OFF signal which represent the start of the signal. The protocol use an encoding method of the bits called pulse width encoding, in which logic one is presented as 1200 microsecond ON signal followed by 600 microsecond OFF signal, while logic zero is presented as 600 microsecond ON signal followed by 600 microsecond OFF signal.

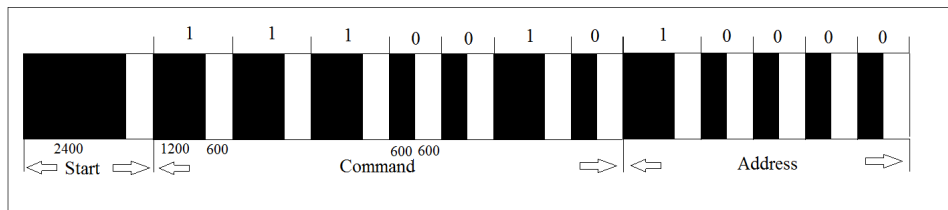


Figure 3.1: Pulse width encoding for Sony protocol

3.2 Infrared Receiver Code Implementation

The A750R Room Locator decoded signal is more complex than the ordinary infrared signal. The signal transmitted carries information about the tag ID, room location, privileges for whom can access certain rooms and other information. This complex signal is hard to decode with any known protocol. So our approach is to uniquely

distinguish each signal transmitted. If we turn each unique sequence of pulses into a unique value, then this value will indicate the room location.

The flow chart in figure 4.3, summarize the operation of decoding the infrared signal. This code is based on a multi-protocol infrared remote library developed by Ken Shirriff [6]. This library was modified to serve the purpose of our project. Interrupt routine is called to decode the infrared signal which was implemented as a finite state machine style algorithm.

The algorithm starts with an idle state waiting for a gap to end, once an ON duration is received the mark state is activated to capture the timing of the on duration. Same thing happens when there is an OFF duration, a space state is activated to measure the time of the off duration. The measurements of the ON and OFF period continue till we receive a long OFF period which indicates the end of the received signal, a stop state is activated.

The raw measurements are put in a buffer to be decoded, the code looks at the sequence of the space signals and compare them according to their length and assign a 0, 1, or 2 if the length is equal, longer or shorter. Same way with the mark signals. The resulting 0-s, 1-s and 2-s are hashed into a 32 bit value. This gives a unique code for each room. The device searches a look up table to match the code to the room number.

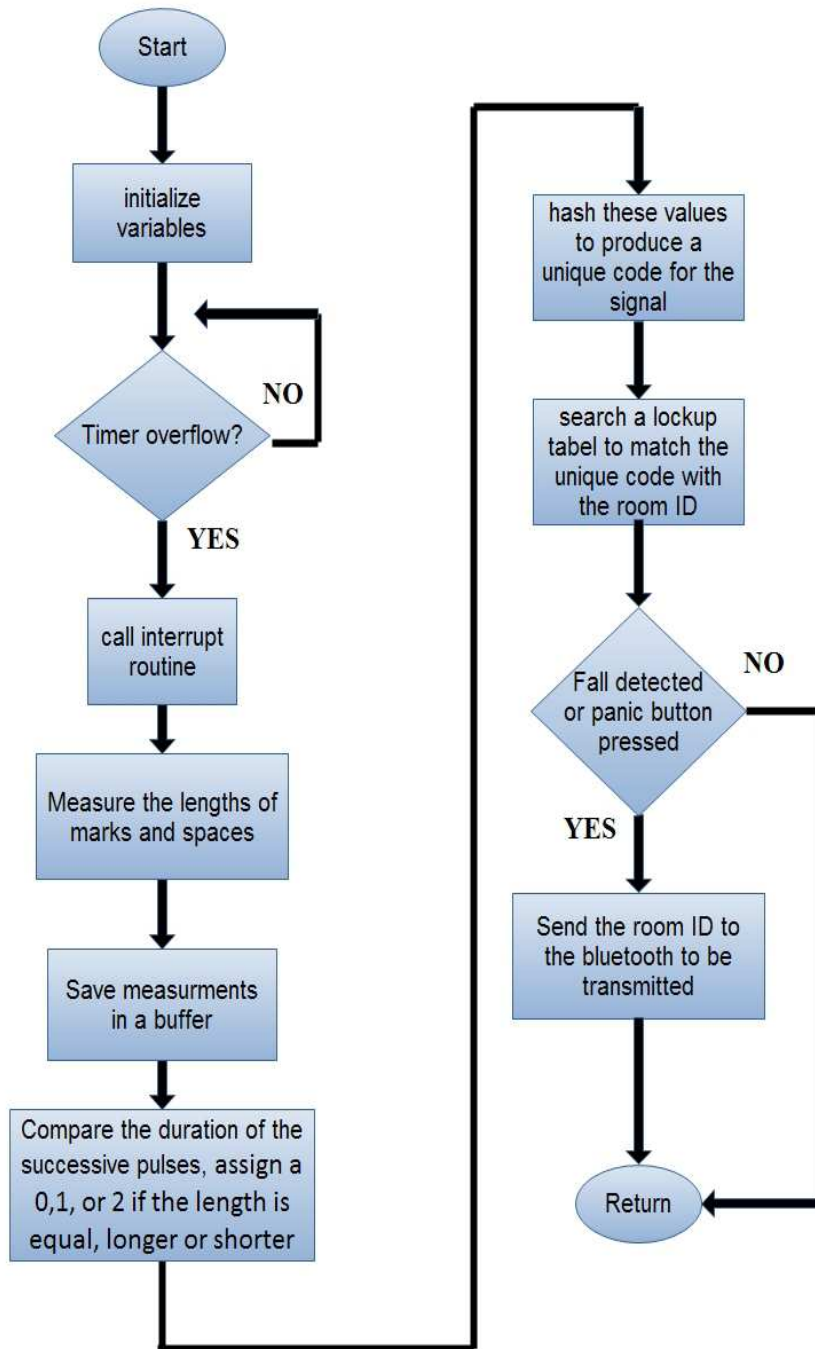


Figure 3.2: Infrared signal Decoding operation

3.3 Converting Accelerometer values into G units

The ADXL335 is a complete 3-axis acceleration measurement system. This accelerometer outputs the acceleration on each axis as an analog voltage between 0 and 5 volts. In order to make use of these values, we need to convert them into acceleration values in G units.

There are many factors to consider when we convert to acceleration values; we needed to have more technical information about the sensor and the ADC such as: The ADC resolution, the ADC reference voltage, the zero g bias of the sensor, And the Arduino's supply voltage.

The sensor gives analog outputs which is received by the ADC in the Arduino board and represented as a binary number. These bits are then converted into mV and finally into g.

The ADC is an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. The output which is a string of bits is proportional to the magnitude of the voltage seen on the analog pin.[8]

The voltage "Range" of the Analog Input is a hardware setting. It determines the maximum voltage that can be connected to the Analog Input. We first need to convert these raw data provided by the ADC to a voltage range using this formula:

$$x = \frac{\text{ADCoutput} * \text{VoltageReference}}{\text{ADCresolution}} \quad (3.1)$$

- ADCoutput: is the binary number that is read from the ADC
- VoltageReference: is the ADC reference voltage which is 3.3 volts in our case

that means the ADC will read a maximum value of 3.3 volts.

- ADCresolution: The "Resolution" of the ADC data is the number of bits used to represent the data. In this case the ADC is 10 bit

⇒ ADCresolution = 2 "raised to the power of the ADC Data Resolution"

$$(2)^{10} = 1024 \tag{3.2}$$

Next, we need to subtract the Zero-g Bias Level which specifies the output level when there is no acceleration (zero input). Analog sensors typically express this in volts (or mV). Zero-g Bias is specified at a particular supply voltage and is typically half the supply voltage, this information can be found in the sensor's data sheet.

The sensor's power supply is 3.3 Volts

⇒ Zero-g bias output is $3.3V/2 = 1.65$ Volts

⇒ $X = x - \text{Zero-g bias output}$

Final step is to convert the voltage reading we calculated into g units. This depends on the sensitivity of the sensor which is the ratio of change in acceleration (input) to change in the output signal. Sensitivity is specified at a particular supply voltage and is typically expressed in units of mV/g for analog-output accelerometers. From the data sheet the ADXL335 has a sensitivity of 300mV/g

$$X(g - units) = \frac{X}{\text{Sensor Sensitivity}} \tag{3.3}$$

X is the accelerometer value in g units. A simple algorithm can be implemented to check whether the acceleration pass a certain threshold value and so a fall is detected, or the acceleration is not high enough to be considered as a fall, which is usually occurs as a result of our daily activity movement.

3.3.1 Fall Detection Algorithm

The total acceleration can be calculated as:

$$total - acceleration = \sqrt{(x)^2 + (y)^2 + (z)^2} \quad (3.4)$$

in order to know the proper threshold value for a fall, we need to attach the sensors on subjects and check the total acceleration values for different daily activities as well as falls, so we can distinguish between regular movements and falls. The threshold value was chosen in the range between 2.8V-3.3V to do simulation and testing on the device since we did not perform tests on human subjects to get an accurate threshold value. It's recommended for future work to perform tests on human subjects in different situations such as sitting, standing, walking...Etc to implement better algorithms to detect falls.

3.4 The Choice of The Wireless Sensor

The wireless transmission method was chosen based on a few specifications. The device had to be able to transmit a reasonable range so only one wireless base station would be required to receive data for a normal size house. Next, the device had to have low power consumption so that a small battery can last for a reasonable amount of time. Finally, the device had to have a flexible network structure for data transmission, and a built-in encryption engine would be beneficial later in production.

A few wireless transmission approaches met these requirements; Bluetooth, the 802.11(Wi-Fi) standard and the Zigbee protocol. Out of those three, two were chosen to proceed with, mainly based on cost and ease of uses. Both of Zigbee and Bluetooth consume low power and can be easily implemented into a design. In figure 5.1. a brief overview of the features and specifications are given.

	ZigBee	Wi-Fi	Bluetooth
Range	10-100 meters	50-100 meters	10 – 100 meters
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Point to hub	Ad-hoc, very small networks
Operating Frequency	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	2.4 and 5 GHz	2.4 GHz
Complexity (Device and application impact)	Low	High	High
Power Consumption (Battery option and life)	Very low (low power is a design goal)	High	Medium
Security	128 AES plus application layer security		64 and 128 bit encryption
Typical Applications	Industrial control and monitoring, sensor networks, building automation, home control and automation, toys, games	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets

Figure 3.3: The key characteristics of Zigbee, Wi-Fi and Bluetooth [9]

ZigBee is a low data rate, low complexity, and has low power consumption. The low power-usage allows longer life with smaller batteries, while Bluetooth has higher power consumption and higher data rate. Those completely different sets of characteristics place the two technologies into different market positions as the type of applications that each one is applied for do not share the same requirements. ZigBee is mainly focused on automation and control, while Bluetooth targets on offering connectivity for mass user market shares (laptops, mobile phones, multimedia and entertainment related devices).

Since the fall detection device is not going to be transmitting data unless a fall is detected or the panic button is pressed, the power consumption was insignificant trade off to the ability to interface the device with different devices that have Bluetooth capabilities already built in such as cellular phones and laptops.

It is worth mentioning here that Bluetooth SIG formally adopted Bluetooth Core Specification Version 4.0, which adds Bluetooth Low-Energy (LE) to what is now being called Classic Bluetooth.

Bluetooth low energy (BLE) is a feature of Bluetooth 4.0 wireless radio technology, aimed at new, principally low-power and low-latency, applications for wireless devices within a short range (Up to 50 meters / 160ft). This facilitates a wide range of applications and smaller form factor devices in the healthcare, fitness, security and home entertainment industries.[10]

3.4.1 Comparison Between Zigbee and Bluetooth Low Energy

Bluetooth low energy and Zigbee use 2.4GHz part of the spectrum. Both technologies have lower power consumption, they function over a short range and they do not cost much. Bluetooth low energy is usually used when you have devices with sensitive power requirements that needs to be constantly connected to the internet which enabled new types of devices to operate for long time on batteries. These devices are used in health care, security, home entertainment and many other areas.

As far as robustness of both technologies, Bluetooth low energy excels in this area more than Zigbee. Bluetooth low energy and classic Bluetooth employs a fast frequency hopping scheme, in which Bluetooth alternate from different frequencies in order to find a clear channel, by making sure that any interference if existed will not last long. This scheme also provide security since its hard to tell what frequency Bluetooth is using to transmit the packets, so its hard to interfere with signal. Zigbee, in contrast, does not have a similar mechanism which makes it less robust.

As far as latency and throughput, Bluetooth low energy has a throughput of 305kbps and a latency of 2.5ms, while Zigbee's throughput is 100kbps with latency of 20ms. Looking at these numbers we can conclude that Bluetooth low energy has better throughput and better time delay.

As far as power consumption, Bluetooth low energy uses a synchronous connection; which means when the Bluetooth (master) goes to sleep all slaves go to sleep, and when the master wakes up, both master and slaves wake up synchronously at the same time. Which means that there is no constant scanning and checking all the time if there is a new message. On the other hand Zigbee use mesh nodes , which requires more power consumption since there is always checking and waking up to check for new messages .

For future work, I recommend using Bluetooth low energy to replace the classic Bluetooth for wireless communication.

4 Code Development and System Integration

In this section I will discuss how the device can be interfaced with different hardware/software to achieve the goals we needed to accomplish in this project.

Figure 4.1, gives a clear framework of the entire project. It shows the different software, hardware and the tools used to upload the code to the hardware in phase 1 and phase 2. The figure also shows the wireless transmission sensor used at the sender side as well as the tools used to receive the data on the receiver side and the different interfaces we implemented.

This section is divided into sub sections, where I will discuss interfacing with the Arduino development environment to upload the code to the microcontroller as well as the other interfaces on the receiver side which includes:

- the HyperTerminal interface
- Matlab interface for graphical representation.
- Android smart phone interface for independent living environment.
- FastFind Software interface for assistant living environment.

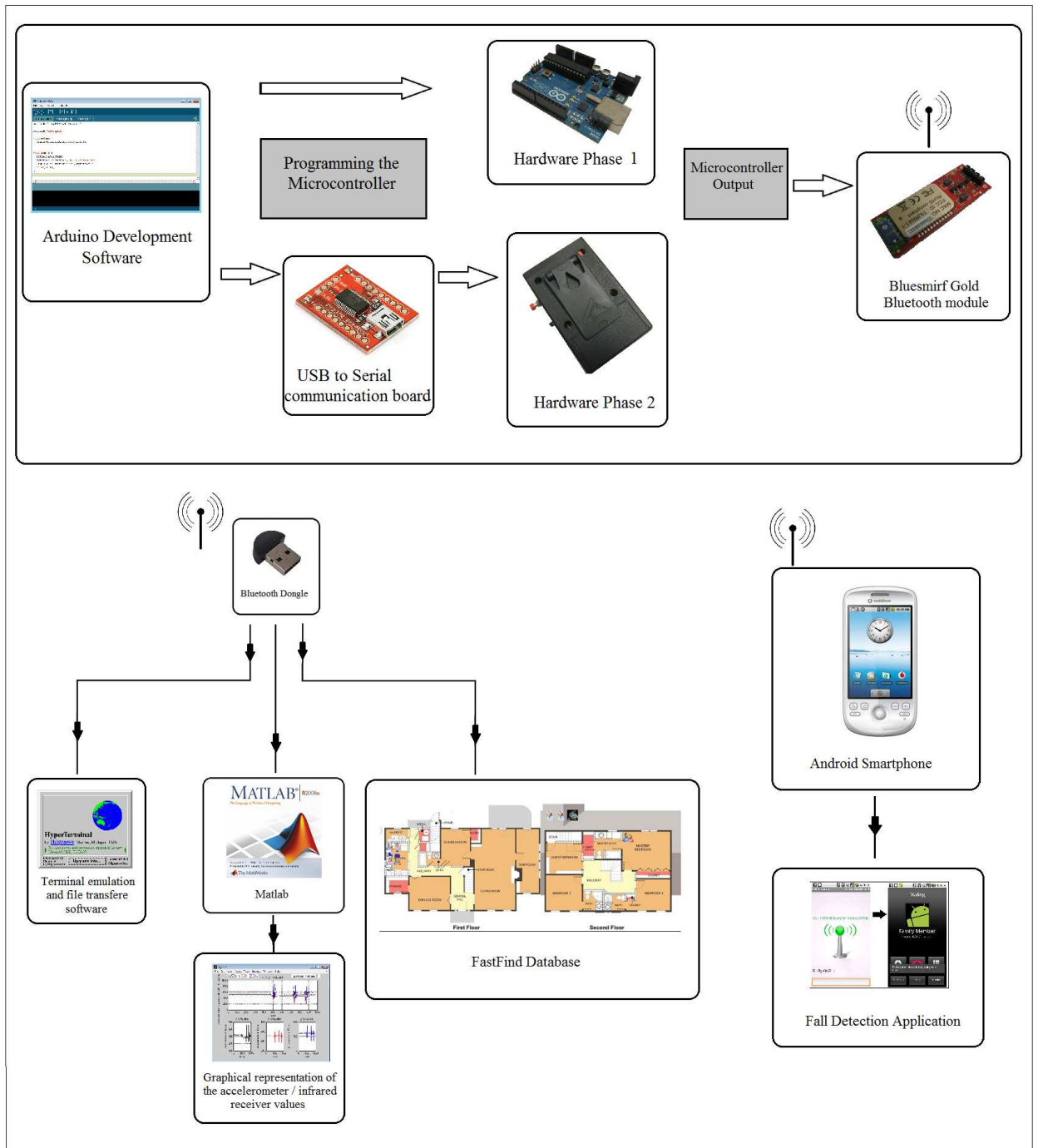


Figure 4.1: Fall detection device interfaces

4.1 Uploading The Code to The Microcontroller

The Arduino development environment was used to upload the code to the microcontroller for both Hardware phases. This development environment has a text editor where you write the code, it connects directly to Arduino hardware after choosing the right board type and the right serial port from the menu to achieve the communication and upload the code.

Uploading the code to the hardware in phase 1 was achieved directly by connecting the Arduino board through A to B Male/Male type peripheral cable to the Arduino development environment. For phase 2, I had two options to upload the code on the ATmega328p microcontroller:

- Option 1: since the microcontroller is connected to the PCB through a 28 pin socket, the microcontroller can be freely detached from the socket and put on the Arduino Uno board socket and the code can be uploaded directly like phase 1. Once the code is uploaded the microcontroller can be detached from the Arduino Uno board and placed back on the PCB socket.

- Option 2: is to use a USB to Serial breakout board to achieve the serial communication and upload the code. In order to connect the breakout board to the microcontroller, I used the same header that the Bluetooth module use since it contains all the pins needed (VCC, GND, TXD, and RXD). The Bluetooth module can be easily detached from the 4 pin header, and the board can be connected to the pins using jumper cables. After connecting VCC and GND, the RX which is pin 2 of the microcontroller connects to the TX of the serial board and the TX which is pin 3 of the microcontroller connects to the RX of the serial port, and then the code can be uploaded.

4.2 Interfacing with HyperTerminal

HyperTerminal is a software that was used to achieve communication between the device and the computer through a serial port in order to verify the output of the microcontroller; such as verifying the change in the numerical values of the accelerometer when the device is moving and verify whether the microcontroller output for the room location match the actual room ID transmitted from the room locator.

The software was also used to send certain commands back to the device to change the settings of some of the components used; For example HyperTerminal was used to send certain commands to the Bluetooth module to put it in a setup mode and then change the default baud rate from 115200 bps to 57600 bps to make it compatible with the Android smart phone. It was also used to change the Bluetooth default sleeping mode to sniffing mode to reduce power consumption.

Figure 4.2, shows the HyperTerminal window where the accelerometer and the room ID is read from the microcontroller through serial port to the terminal.

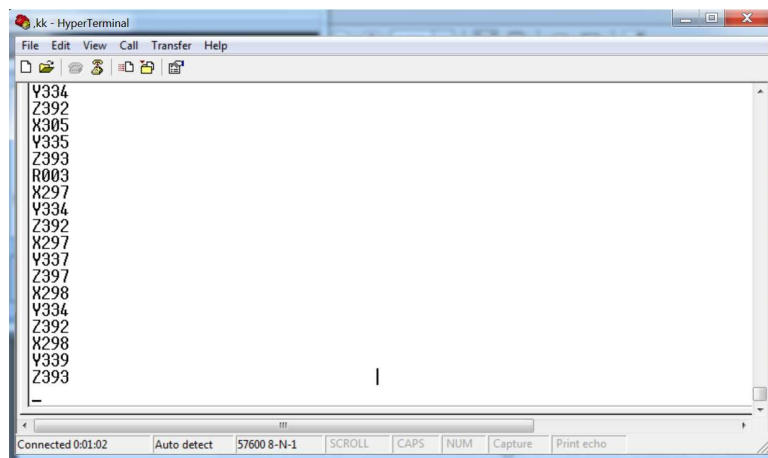


Figure 4.2: HyperTerminal interfaces

4.3 Interfacing with Matlab (Graphical Representation)

Matlab R2009a was used to interface with the fall detection device for graphical presentation of the accelerometer values as well as showing the room ID. This graphical presentation is important to run analysis on the data received, locate the threshold points for falls and daily activities, and checking if the accelerometer values are within the right range to make the necessary calibrations needed.

Bluetooth dongle on the PC side is used to receive the data from the device; the process of receiving data through serial port goes through these steps:

1. Creating a serial object. This interface is established through a serial port object which we create using the SERIAL function.
2. Set up communication parameters by using the set function. The serial port object supports functions and properties that allow you to configure serial port communications such as baudrate, databits, and stopbits.
3. Open serial port using fopen, which open the serial port for read access
4. Once the receiving is done, we close and delete the port.

Figure 4.3, shows a flowchart of the code implementation in Matlab. The software starts scanning the data using FSCANF function, and parses them according to their tag using the REGEXP (regular expression) function that matches strings of text. The values from the fall detection device are sent pre-tagged to be identified. For example the accelerometer values in the X, Y and Z directions are tagged X250, Y350, Z250, while the room Id is tagged with the letter R, for example room 3 is sent from the fall detection device as R003. Once a match is made the software skips the tag and converts the string following the tag to a number and refreshes the new value on the graph using the SET function.

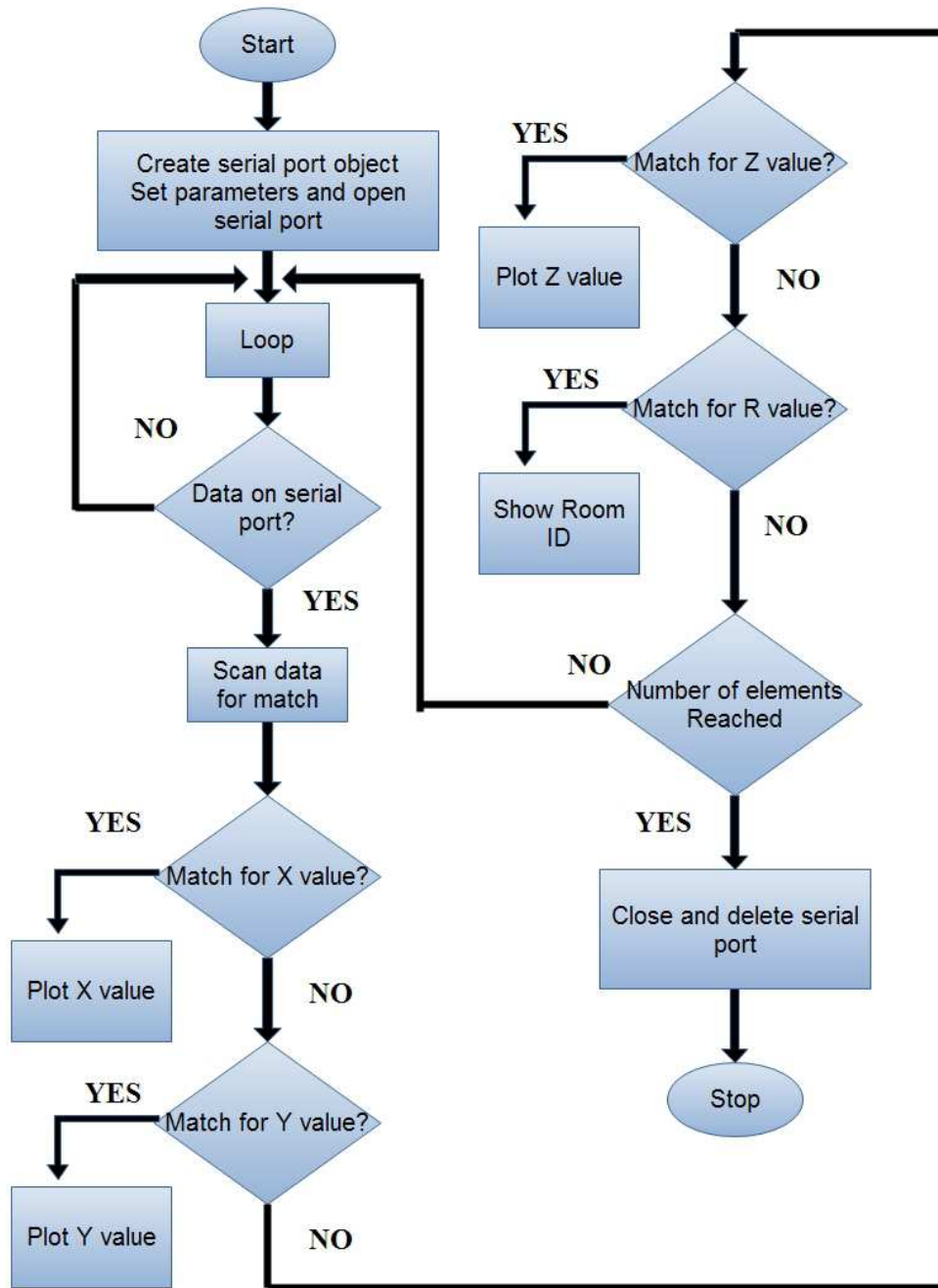


Figure 4.3: Matlab code flowchart

The function PLOT was not used to plot the values in real time, since every time a new value is received Matlab would create a new graphics object for each point which causes Matlab to become irresponsive with time after creating all these graphics objects; instead one graphics object is created and its properties are updated with the new values, so that we don't have to remove the old values and preallocate their space in memory.

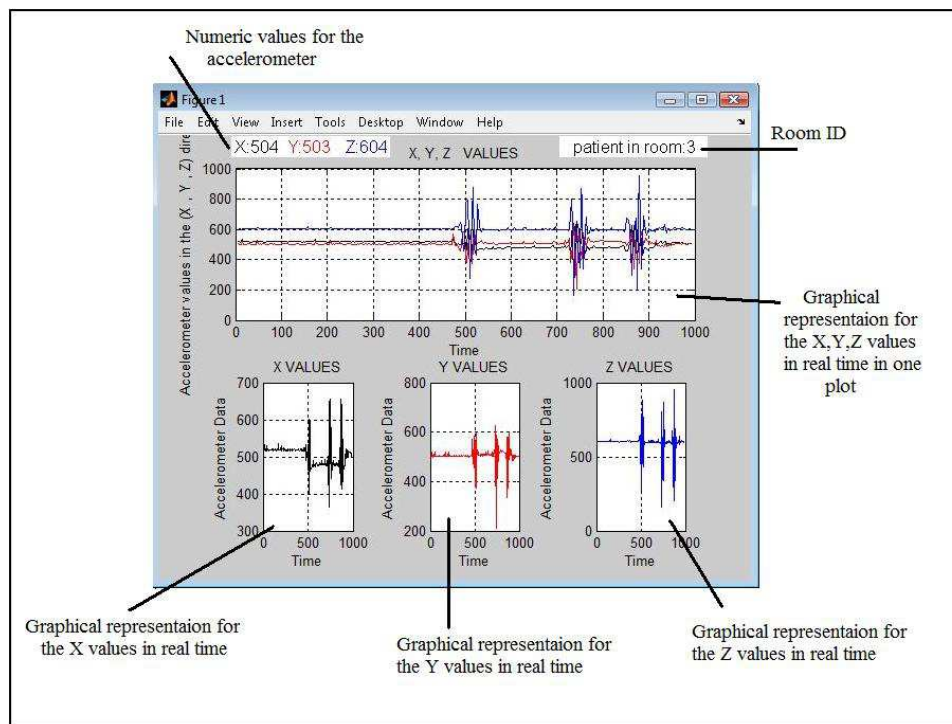


Figure 4.4: Matlab graphical representation for the accelerometer and IR values

4.4 Independent living for the elderly

4.4.1 Interfacing with a smart phone

Independent living for the elderly is an important topic that we tried to tackle in this project. A variety of independent living products and services are now available to help seniors remain in their own homes despite their changing physical needs. This helps delay -and in many cases, totally avoid- moving into assisted living or nursing

homes. These products have one thing in common; which is the high cost that made them difficult to be commercially successful for personal use.

The high cost comes from the fees that you have to pay monthly for the service as well as the cost of the equipment. The service includes a dispatcher that is available 24 hours a day 7 days a week waiting for a call to send the help needed, this dispatcher can send help if the elderly push the button and ask for help. In many cases seniors fall and their head hit the wall or the floor and become unconscious and cant push the button or ask for help which make these products unreliable.

In my project, I worked on solving the two main problems, the reliability of the product and the high cost. In my device there are two ways to ask for help, A manual way by pushing a panic button like the other products in the market, in which the device send a command through a wireless sensor to the base unit which initiates a call for 911 or for a family member. The other way is automatic through a 3-axes accelerometer and algorithms that analyze the accelerometer output and detect a fall, once the fall is detected the device send a command through Bluetooth to ask for help. Having an automatic and manual way to ask for help makes this device more reliable, and can ensure that the elderly gets the medical help needed whether he/she was able to push a button or not.

Most people use cell phones that have Bluetooth capabilities. By the close of 2005; wireless subscriptions hit nearly 2 billion on a worldwide basis, with cellular mobile dominating the wireless technology field according to a trends study from Deloitte Research. By establishing a connection between the Bluetooth modules in the device and the cell phone once a fall is detected, we can initiate an emergency

call directly from the phone to 911 or to family members. For the fall detection device we chose to use Bluetooth so that we cut the cost of having a base unit to initiate the emergency call and the need to have a dispatcher to send the help needed.

4.4.2 Interfacing With The Android Phone

The Android smart phone was chosen to implement the application that will interface with the fall detection device. Before developing the application, I had to change the baud rate of the Bluesmirf Gold module so that it matches the built in Bluetooth baud rate on the android device

4.4.3 Changing The Baud Rate of the Bluetooth Modem

The baud rate for the Bluetooth module needed to be changed to be compatible with the android Bluetooth baud rate. The first communication between the device and the android phone showed weird characters on the android phone which indicates that the baud rate needed to be adjusted.

The default baud rate of the Bluesmirf gold Bluetooth module is 115.2k. After different attempts to change the baud rate; I noticed that changing its value to anything lower than 57.6K, the message shown on the android device is "cut up". 57.6K was the right baud rate in for the Bluetooth module in which the whole message was received successfully.

The Arduino Uno board was used to change the default baud rate of the Bluetooth module, figure 4.5 shows the pin connections between the Arduino and the Bluesmirf gold modem.

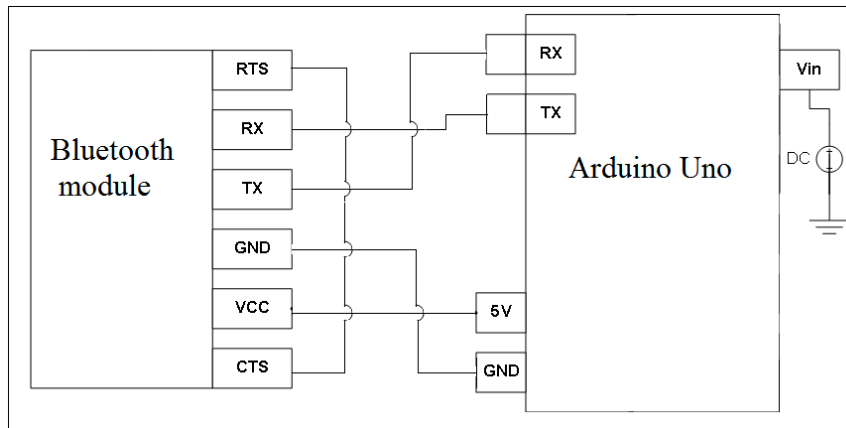


Figure 4.5: Pin connections between the Arduino Uno and the Bluetooth modem

After the connections are made, the Bluetooth module shows RED light blinking on the board, which means the board is connected to power but not paired with any devices. Through the serial monitor of the Arduino we connect to the Bluetooth module using the default baud rate of 115.2K as shown in figure 4.6. Using the serial monitor the command \$\$\$ is sent to the Bluetooth module to put it in the command mode; a fast blinking red light on the Bluetooth module board shows that the module is in the command mode and you will immediately see CMD on the serial monitor display. To change the baud rate to 57.6K i use the command "SU,57", if the baud rate is changed on the module, you will immediately see AOK command, otherwise, you will get ERR. To make sure the baud rate has changed, we send the command "D" to show the changes in the Bluetooth module settings.

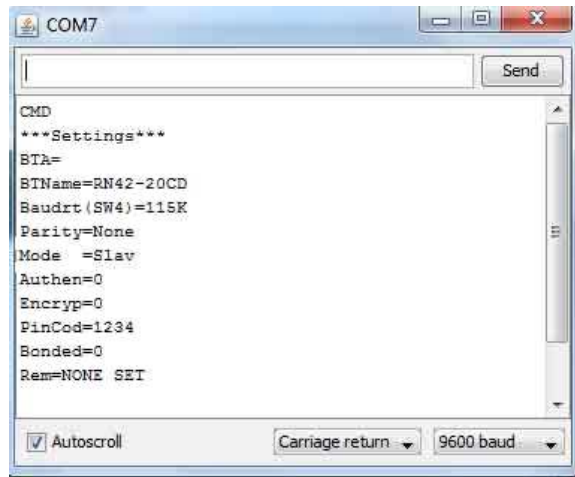


Figure 4.6: Bluetooth module Baudrate default settings

To exit the command mode command `-<cr>` is sent; the slow red light blinking on the board shows that the module exited the command mode.

4.4.4 Android Application

I modified the Google Android sample app called "Bluetooth Chat" so that the android phone is able to receive commands from the fall detection device. According to the command received, the application triggers an event. The event is calling a family member in case the panic button is pressed or call another number such as emergency number in case of fall detection. The application is also modified to send an optional text message that contains the address where the fall happened and any information needed to send the help. I also modified the permissions in the sample code so that the application is able to initiate a call or send a message from inside the application.

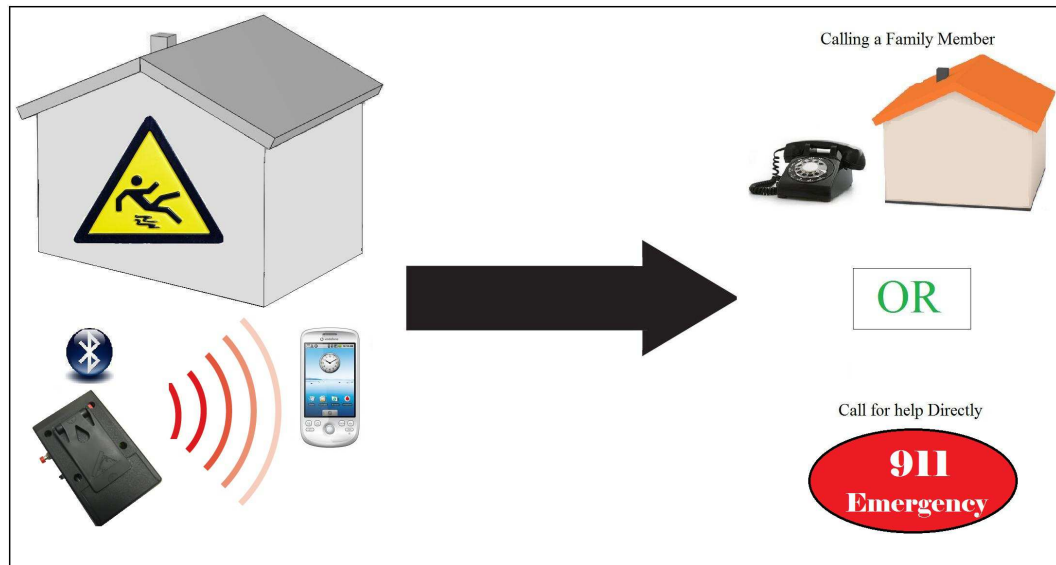


Figure 4.7: Interfacing with smart phone to request help

Figure 4.7, shows the process of calling for help using an Android phone. The device communicates through Bluetooth to the Android phone in case of a fall or if the panic button is pressed. First, the application looks for Bluetooth devices in range and connects to the Bluetooth module of the fall detection device which is usually found under the name FireFly. Once the two devices are paired the screen shows a message "connected to FireFly" .



Figure 4.8: pairing the android app with the fall detection device

The Android phone is ready to receive commands from the fall detection device; the application is programmed to call 911 directly for medical help, or call a family member. The Android can also call a certain number depending on whether a fall is detected or the panic button is pressed. For example in case of a fall the device send a command "F" (for fall) to the Android phone, which triggers an action in The Android phone to call 911 for help.



Figure 4.9: Fall detection on the Android smart phone

On the other hand if the panic button is pressed, the device sends a command "P" (for panic) to the Android phone. This action can be treated as if the elderly person needs an assistant of some kind or help but it's not an emergency situation. In this case the Android phone will call a family member instead of 911, so that the family member can assist the situation and call for medical assistance if needed.



Figure 4.10: Request help through panic button on the Android smart phone

More than one panic button can be placed on the device depending on the situation. An IP camera can also be used in the elderly home and can be turned on by a family member once he/she receives a call for help so they watch what is happening from anywhere using their computer or their smart phone and can properly evaluate the situation and take a proper action.

4.5 Interfacing With FastFind

FastFind is a software that helps nursing home personnel track, manage, and optimize the usage of assets using RFID technology. It may also be used to track nursing home personnel and patients or to monitor events that need special care. [11]

FastFind consists of the following components :

- A750 Room Locator: The A750 Room locator transmits infrared pulses to report the room ID, these pulses are transmitted every one second [12]
- IR-Enabled Active Tags: these tags are attached to assets to track the current location and movement of these assets.
- RF-Reader: The reader receives data transmitted by the active tags, this includes the room ID that is received from the A750 room locators.

The FastFind software contains three separate applications:

- A reader manager allows system administrators to specify and control RFID readers. The reader manager can handle multiple RFID readers connected to a local area network.
- A database manager tool allows system administrators to bring up or shut down the database server, to start the database empty, and to export or import a database.
- FastFind application allows users/administrators to track and monitor assets and people.[12]

4.5.1 The Tracking Process

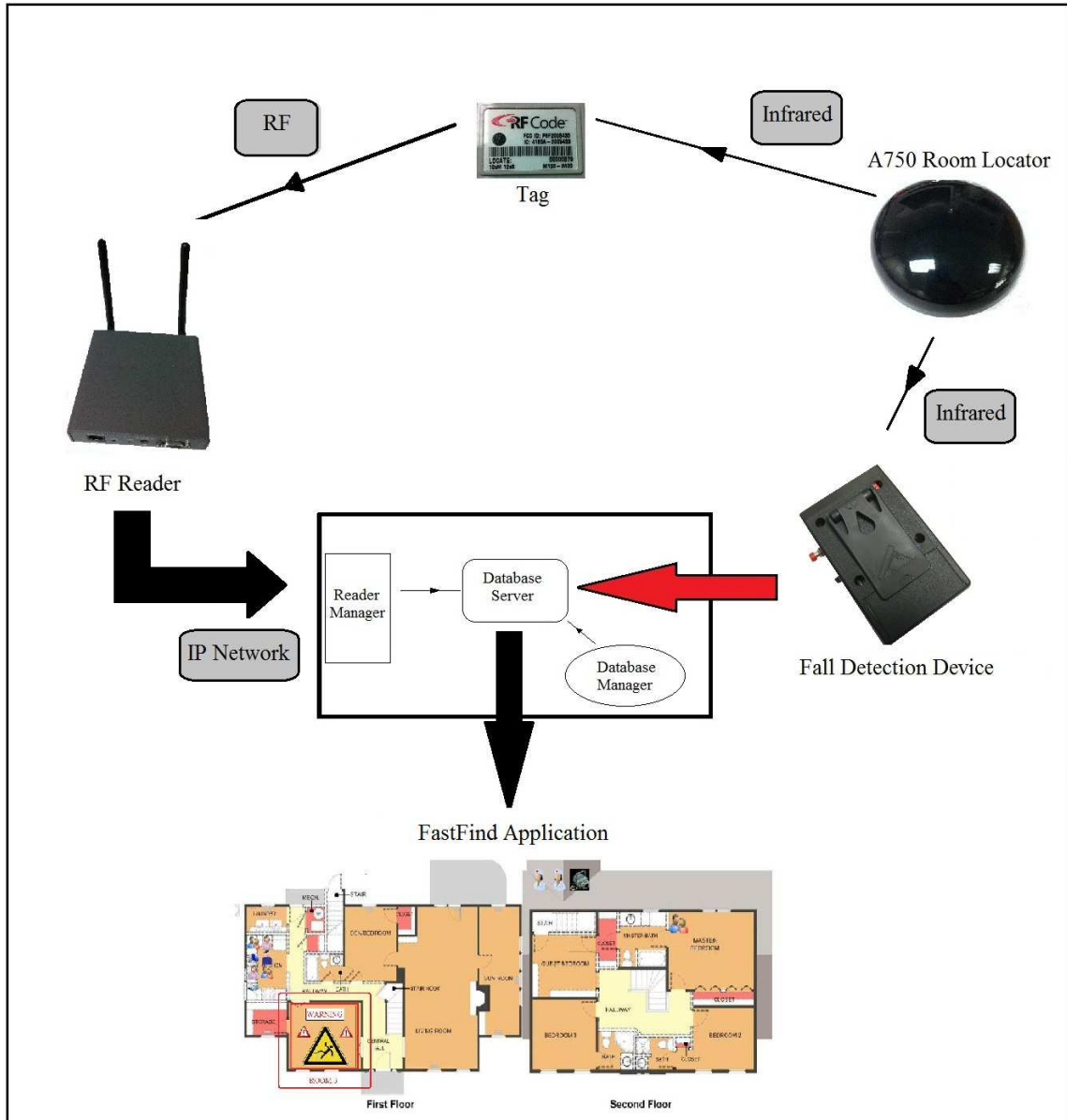


Figure 4.11: Tracking process for the FastFind software

Figure 4.11, shows the tracking process for the FastFind software with the fall detection device interface. The software runs on Microsoft Windows platform and uses RFID readers and room locators from RFCODE Inc. The software controls multiple RFID readers connected to a local area network and reads tag information using those

readers. A room locator sends its locator ID to individual tags within the same room using infrared light. A tag receiving a locator ID sends that ID together with its own tag ID to surrounding readers. This is how the FastFind software identifies which room a tag is in.

One of our goals is to implement the fall detection device in an assistant living environment or hospitals without adding extra cost. Its important to know where the fall was detected to send immediate help for the elderly especially when you have hundreds of rooms and multiple floors. A new component is added to the device to solve this problem; An infrared receiver that is capable of decoding the infrared pulses transmitted from A750 room locators which made interfacing the fall detection device with the fastfind software possible.

4.5.2 FastFind Fall Detection Application

The fall detection device continuously decoding the infrared signal and running algorithms to analyze the accelerometer values to check for a fall. If a fall is detected or the panic button is pressed, the device check the room ID and sends this information to the fastfind application through bluetooth.

On the fastfind side, a bluetooth dongle is used to receive the information from the fall detection device through a serial port. The application starts the initialization process by creating a serial port object and set the parameters such as baud rate, stop bits, flow control . ReaderDBManager Class is defined to match the room ID received with the room location (for example if the room ID received is 3, the ReaderDBManager check the correspondding location for room 3 in the database such as the kitchen or the storage room and that is used to identify the location not just the Room ID).

Bluetooth ID number is used to uniquely identify different fall detection devices. Once the communication between the FastFind application and the fall detection device is achieved, the application checks the Bluetooth unique ID number; if the Bluetooth ID number is assigned to a patient name, the operator has the option to assign a new patient name or continue with the name that is already assigned. If that Bluetooth ID is not registered a pop up screen shows up asking the operator to enter the patient name to assign it to the fall detection device.

If a fall is detected or a panic button is pressed the application starts an event to parse the received data to take the proper action. Figure 4.12, shows the event that is initiated by the application once a fall is detected or the panic button is pressed.

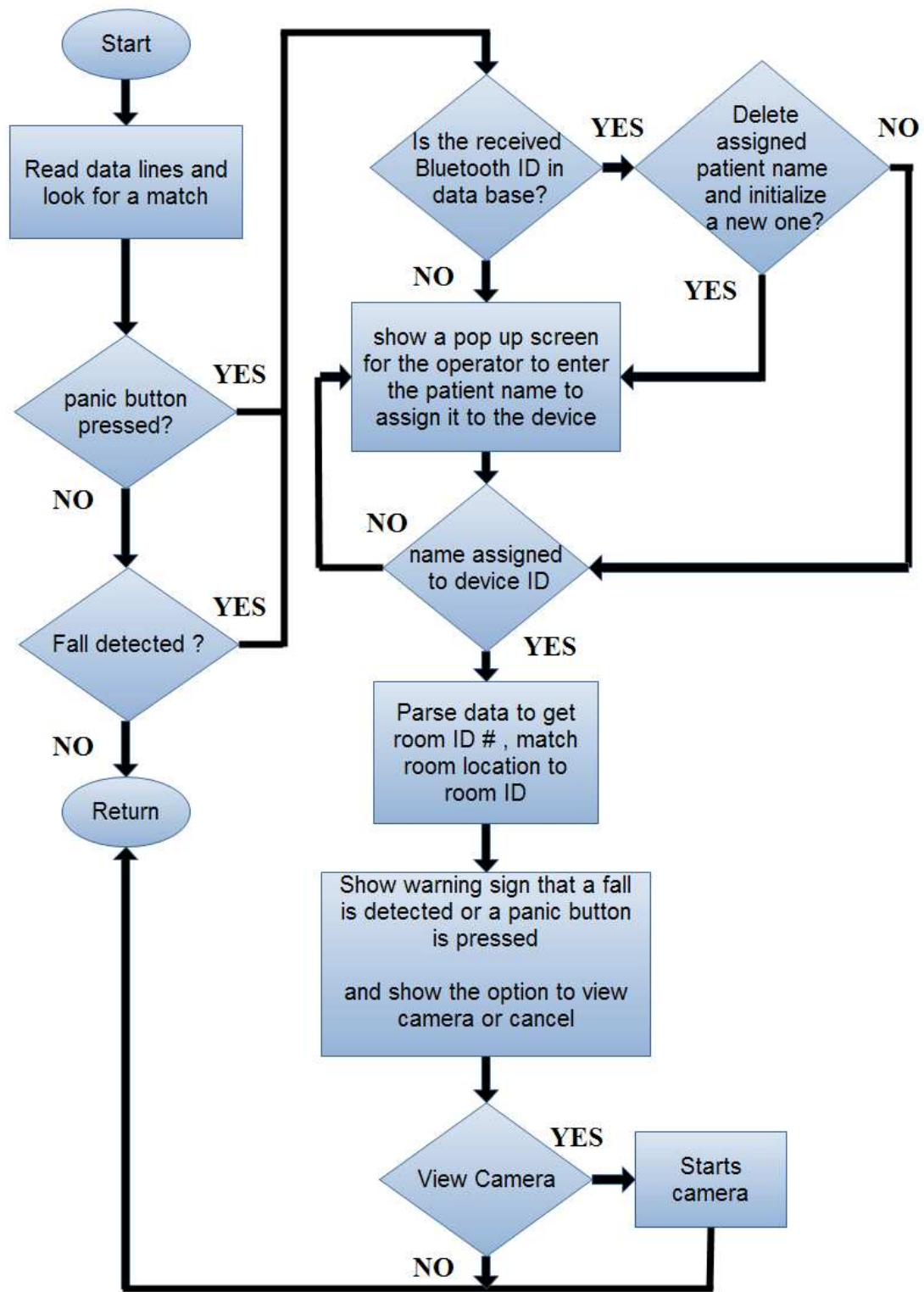


Figure 4.12: Tracking process for the FastFind software

Once the event starts, the application starts parsing the data to check if there is a fall or if the panic button is pressed, once either situation happens the application extracts the room ID and shows a warning sign to the operator that shows a fall is detected or a panic button is pressed. The pop up screen also shows the room ID, room location, patient name, and the option to view an IP camera from that location. If the operator chooses to view the camera, a 2nd pop up screen shows up that has a view of the camera with other options such as a 2 way chat and the ability to adjust the view of the camera.

The camera used is a TRENDnet TV IP422W Wireless Day/Night Pan/Tilt Internet Camera Server with 2-Way Audio Network camera. This camera gives the operator the ability to hear and talk to people in the camera's viewing field, day or night from any Internet connection with the option to zoom and Pan the camera side-to-side in 330 degrees and tilt up-and-down 105 degrees.



Figure 4.13: FastFind fall detection application with camera view

5 Conclusion

5.1 Summary

The healthcare industry is in need of better ways to improve the care of both clinical and assisted living patients. This project is a good starting point for a product that will solve vital monitoring needs as well as provide data for medical researchers to find new treatments and solutions to aid and monitor the growing population.

The designing and development of the fall detection device went through 2 hardware design phases. In Phase one a development board was used with all components soldered into a shield which was mounted on top of the development board with all the components in the inside to provide protection for the components in case of a fall. A printed circuit board was designed and developed in the second phase, the device has a smaller size, lower power consumption, and was designed to fit an enclosure with a belt attachment to be presented as a commercially developed product.

Applications and softwares were developed to build a communication bridge between the fall detection device and other systems. These interfaces used to find solutions to detect falls related injuries in an independent living environment as well as an assistant living environment. An application developed on the Android smart phone was used to communicate with the fall detection device. The application automatically dial an emergency number to request help in case of a fall or in case the senior pressed the panic button, which is a great help for the elderly who choose to live independently. For hospitals and nursing homes, the fall detection device is interfaced with the FastFind application to keep track of the patients without adding extra cost or extra equipments. An infrared receiver was added to the device to decode the infrared signal from room locators that are used in these environments, so

that the device is able to detect falls, find the location of the fall and transmit this information through Bluetooth to the FastFind software. In case the panic button is pressed or a fall occurs the interface will show the name of the patient, location of the patient and the option to have a live video feed of that location.

Developing this platform was very useful to compare different technologies and learn their limitations and advantages. Further research is still needed to improve this platform to meet the standards of the consumer quality and use.

5.2 Recommendations And Future Work

This project still has a long way to go to be ready for commercial use, there are a lot of things that can be added to this device to make it more reliable. Listed below are some of the things that can be considered for future development on this device.

- The size of the device can be much smaller; the device was designed to fit the smallest enclosure I was able to find that has a 9v battery compartment. The board can be made half the size and a custom enclosure can be made with openings for the switch button, panic button, and the infrared receiver.
- Bluetooth low energy module can be used instead of the classic Bluetooth module to have a low power consumption device that can have a longer battery life.
- More sensors can be added to make the device more reliable. A pressure sensor can be added to tell how far the device is from the ground which can identify the user's position, whether the elderly is standing or laying on the floor which can tell for sure if a fall occurred or not which reduces false alarms.
- The fall detecting device provides a structure with wireless capabilities so that more functionality can be added easily to the device by adding different sensors

such as a temperature sensor or a blood pressure sensor. The final product is one powerful portable device capable of providing continues healthcare monitoring for patient's vital parameters.

- Human subject testing is recommened by mounting the device on different areas of the body to find an accurate threshold value which is used as a parameter to differentiate between falls and daily life activates, as well as finding out the best place to attach the device whether its around the neck, waist or other areas.
- The three axis accelerometer ADXL335 can be replaced with a Gyroscope/accelerometer combo board that incorporates a dual-axis gyroscope and Analog Devices triple axis ADXL335 accelerometer in a tight footprint. By combining the gyroscope and accelerometer sensors, the board enables us to easily incorporate roll, pitch, and tilt measurements into the device which adds more reliability to the algorithm used to detect falls

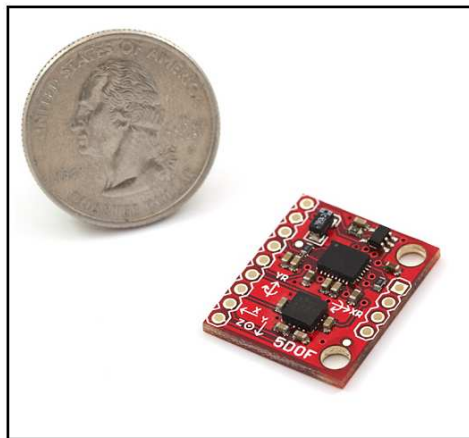


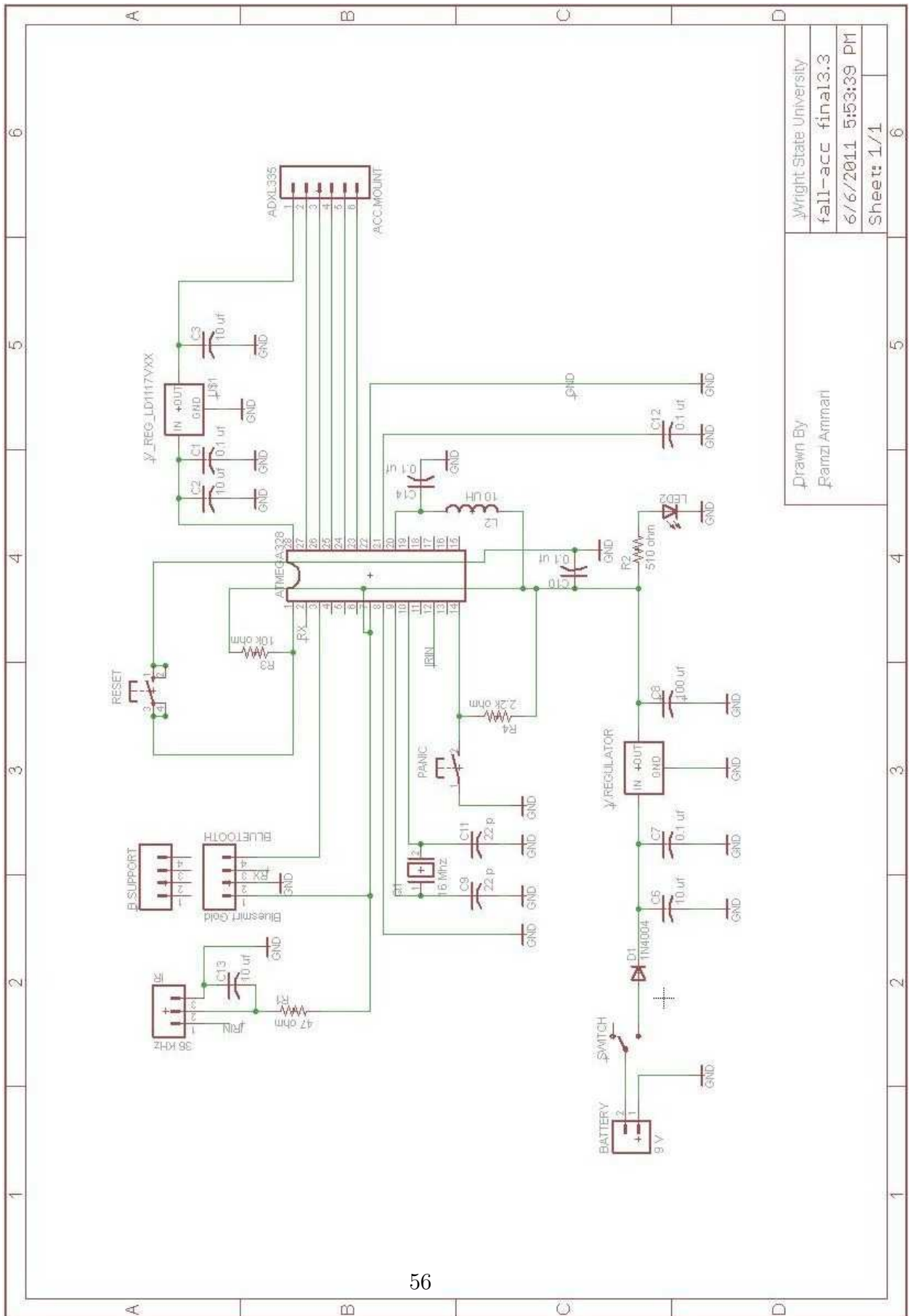
Figure 5.1: Gyroscope/accelerometer combo board [13]

- Its important to provide protection to the elderly after the fall occurs. Developing a mechanism to protect the elderly if a fall happens can be the focus of future work. For example deploying a cushion which can serve as protection in case of a fall.

6 Attachments: Schematic

Attached is the schematic for phase 2 PCB.

- Fall detection device phase 2 printed circuit board schematic



Wright State University
 fall-acc final3.3
 6/6/2011 5:53:39 PM
 Sheet: 1/1

Drawn By
 Ramzi Ammani

7 Bibliography

References

- [1] Department of Health and Human Services. Fatalities and injuries from falls among older adults - united states, 1993-2003 and 2001-2005. pages 1221-1224, November 2006. Morbidity and Mortality Weekly Report.
- [2] Rubenstein LZ, Robbins AS, Josephson KR, Schulman BL, Osterweil D. "The value of assessing falls in an elderly population. A randomized clinical trial". Annals of Internal Medicine 1990;113:308-16.
- [3] G Pterolle, P Fraisse, M Mavros, and I Etxeberria. Automatic fall Detection and activity monitoring for elderly. Sponsored by the CRAFT project.
- [4] Arduino Uno, Arduino uno Overview, www.arduino.cc/en/Main/ArduinoBoardUno
- [5] Analog Devices, ADXL335 Accelerometer datasheet, www.analog.com/static/imported-files/data_sheets
- [6] Multi-Protocol Infrared Remote Library for the Arduino, Retrieved August 29, 2009, from www.arcfm.com
- [7] IR remote control theory, Retrieved September 23, 2009 from www.sbprojects.com/knowledge/ir/ir.htm
- [8] Analog to digital converter, Retrieved July 22, 2011 from http://en.wikipedia.org/wiki/Analog-to-digital_converter
- [9] The comparison of Wi-Fi, Bluetooth and ZigBee, The key characteristics of Zigbee, Wi-Fi and Bluetooth, <http://www.sena.com/blog/?p=359>
- [10] Bluetooth low energy, Retrieved July 16, 2011 http://en.wikipedia.org/wiki/Bluetooth_low_energy

- [11] FastFind, FastFind Version 1.0.2 User Manual, August 23, 2010
<http://www.rfcode.com/>
- [12] A750 Room Locator, User Manual, <http://www.rfcode.com/>
- [13] Gyroscope/accelerometer combo board. <http://www.sparkfun.com/products/9268>