March 10, 2016

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Colombia
V5A 1S6

Re: ENSC440W – Design Specifications for a Smart Room Control System

Dear Dr. Rawicz,

The enclosed document describes the design specifications related to our *Smart Room Control System*, a product of *MOTUSCONTROL*. We are aiming to implement a room system that will control and secure electrical devices within the room based on hand-gesture recognition. This product will enhance people's control of their own space by using meaningful hand gestures and smartphones.

The following design specification provides a detailed explanation of the prototype design of our product. It also includes a detailed test plan for both the hardware and software components. The design specification will serve as a design guide for the development of the prototype.

MOTUSCONTROL consists of five diverse and highly qualified engineers: Moha Alharbi, Saad Alkhalifah, Ryadh Almuaili , Adrian Fettes, and Yuhui Jin. Please feel free to contact me for any questions or concerns. I can be reached by phone at (604) 500-5416 or by email at salkhali@sfu.ca.

Sincerely,

Saad Alkhalifah
Chief Executive Officer
MOTUSCONTROL

Enclosure: *Design Specification for Smart Room Control System*

MOTUS CONTROL

Design Specifications for Smart Room Control System

Project Team:
Moha Alharbi
Saad Alkhalifah
Ryadh Almuaili
Adrian Fettes
Yuhui Jin

Contact Person:
Moha Alharbi
malharbi@sfu.ca

Submitted to:
Dr. Andrew Rawicz
Mr. Steve Whitmore
School of Engineering Science
Simon Fraser University

Issued Date:
March 10, 2016

Revision:
1.0

Abstract

As a physically impaired person or as a someone close to a physically impaired person, you are aware of the difficulties that these people face in their everyday lives. Everyday tasks that are seemingly easy to perform by people without disabilities are extremely difficult or in some cases impossible to perform by physically disabled people. The aim of our project is to give disabled people the ability to complete simple, everyday actions with no more difficulty than anyone else.

Presently, so much new technology is aimed at bringing people to greater heights, providing us with new senses, new abilities. Google glass would provide us with a constant stream of information about our surroundings, and keep us constantly aware of the happenings of our digital lives. The Oculus Rift allows new entertainment opportunities such as Virtual Reality movies and games. Fitness tracking watches and other devices allow people to constantly monitor their bodies in a way which was, until recently, impossible. It is easy to forget, as these new senses and controllers are created, that some people do not have the same base capabilities that everyone else takes for granted.

The smart room control system is solution that aims to give disabled people the ability to control them with simple hand gestures. Some abilities which our solution will grant the user are to turn lights on and off, change the thermostat, check other parts of their house for movement, and be alerted if an alarm goes off. Our hope is that by removing the burden of completing these common tasks, we can significantly improve the quality of life for the differently abled.

MOTUS CONTROL

Table of Contents

MOTUS CONTROL

List of Figures:

List of Tables:

Glossary

VCC: Collector Supply Voltage

SMS: Short Message Service

V: Volts

AC: Alternating Current

DC: Direct Current

WiFi: Wireless Fidelity

GSM: Global System for Mobile Communications

Hz: Hertz

mm: Millimeter

cm: Centimeter

°C: Degree Celsius

UART: Universal Asynchronous Receiver/Transmitter

SPI: Serial Peripheral Interface

SD card: Secure Digital Card

TCP - Transmission Control Protocol

WEP - Wired Equivalent Privacy

WPA: Wireless Fidelity Protected Access

1. Introduction

The Smart Room Control System is designed to give the elderly and disabled people freedom to control the surrounding devices. The smart room is equipped with microcontroller that communicates with the wearable sensor, server and intended devices. Therefore, the user wears the wearable sensor to send a signal to the microcontroller in order to control the intended device. Moreover, the user has the ability to control lights, temperature and many other devices to improve the user's quality of life in performing basic day to day activity. Each hand gesture is predefined in the microcontroller and corresponds to specific devices, turning them on and off. Additionally, the caretaker can monitor the user's environment by checking the smartphone application. As a result the microcontroller updates the status of devices to the server to reflect the change in the smartphone application. Figure 1 below provides an overview of the system's framework to convey a general idea for both *MOTUSCONTROL* members and non-technical consumers.

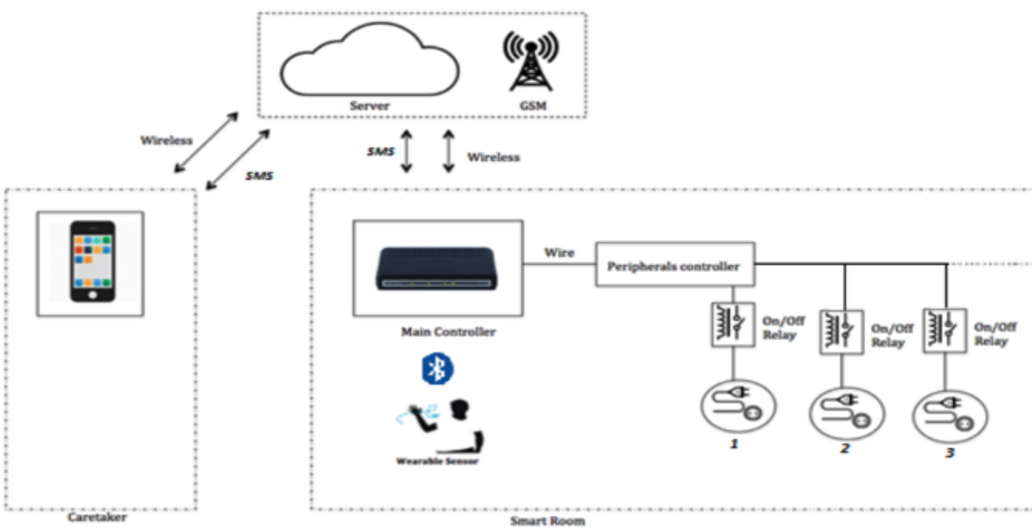


Figure 1: High-Level Model of Smart Room Control System [1][2]

1.1 Scope

This document provides an in-depth explanation of the technical aspects of each component of the Smart Room Control System design. It will follow the functional requirements provided by Functional Specification for the Smart Room Control System. The design details will provide deeper understanding in how those functional requirements will be implemented. This document describes the proof-of-concept and prototype models of the *Smart Room Control System*. Therefore, only core functional requirements will be mentioned.

1.2 Intended Audience

The design specification will be used and implemented by all members of *MOTUSCONTROL* throughout the development phases. Additionally, this document serves as a guideline for the design and implementation process to ensure compliance with procedure. The procedures listed below will be also followed for testing the product as a final evaluation.

2. System Requirements

A high level overview of the system is presented in this section. The detailed design specification of each unit that composes our *Smart Room Control System* is discussed in their own subsections. However, design details that are common on all parts of the *Smart Room Control System* are discussed here.

2.1 System Design

The *Smart Room Control System* solution provides the ability to control the peripherals surrounding the system's user by using simple hand gestures. Figure 2 below provides an overview of the system's framework to convey a general idea for both *MOTUSCONTROL* members and non-technical consumers.

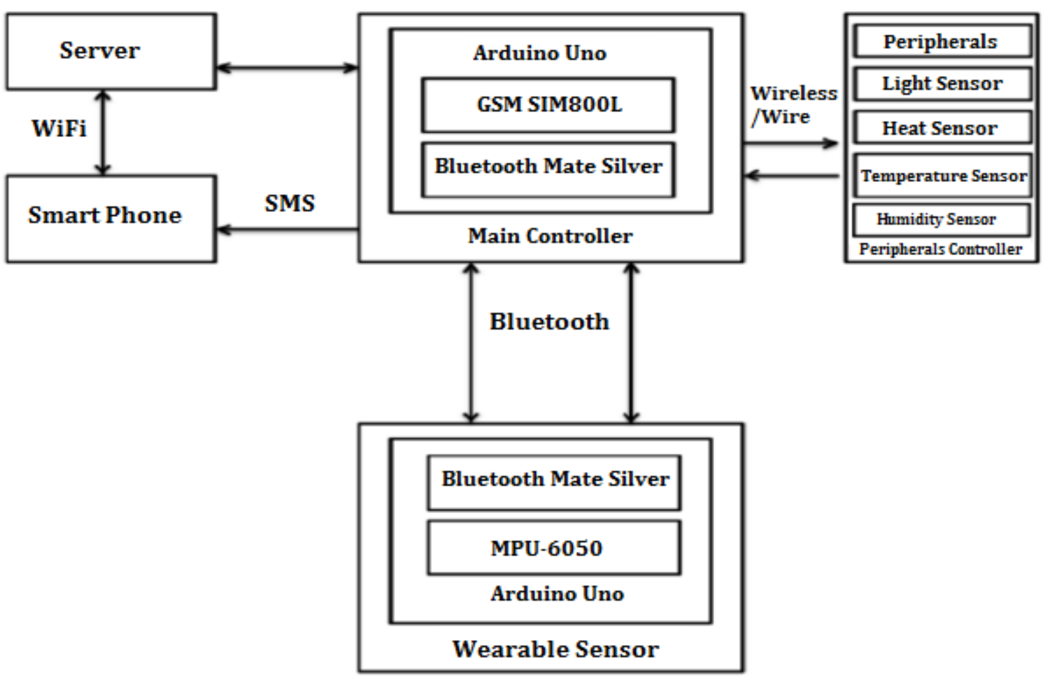


Figure 2: System Block Diagram

Depicted in Figure 2 above, the *Smart Room Control System* is triggered by a hand motion sent from the wearable sensor to control a device inside the room. The Arduino Uno processor, in the main controller unit, will receive the data of the hand motion from the wearable sensor. The hand motion will then be processed and passed to the peripheral's controller to control the device associated with the hand motion sent by the user. Alternatively, the room can be accessed and monitored remotely by the systems caretaker using the smartphone application. The connection between the application and the smart room will be through a server. The server will be updated frequently to reflect the status of the devices and sensors inside the smart room.

2.2 Microcontroller Design

To control the processes and the interactions between the different subsystems, *MOTOUSCONTROL* has decided to utilize an Arduino Uno microcontroller to form the two main units of the *Smart Room Control System.* The microcontroller will be part of the main controller unit and the wearable sensor unit. The Uno is a microcontroller based on the ATmega328P as shown in Figure 3. It has 14 digital input/output pins, 6 analog input pins, a 16 MHz clock speed, a USB connection, a power jack, an ICSP header and a reset button. The board operates at a recommended voltage supply that ranges from 7–12 volts. The Arduino Uno is a suitable choice for our proof of concept, due to its various free code libraries [3].

Figure 3: Arduino Uno [4]

The technical specifications of the microcontroller Arduino Uno are summarized in Table 1.

Table 1: Technical Specifications of Arduino Uno [3]

Microcontroller	ATmega328P
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (2 pins for Serial Communication)
Analog Input Pints	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

2.3 Bluetooth Module

Sparkfun Bluetooth Mate Silver is chosen in our design due to its affordability and compatibility with different types of devices. Two Bluetooth modules are used; one is integrated with the main controller and the other one is implemented with the hand controller. Figure 4 shows the Bluetooth Mate Silver module.

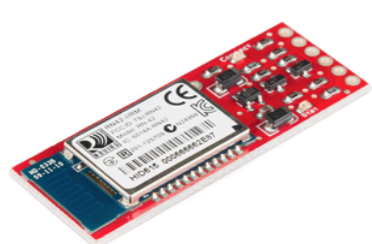


Figure 4: Bluetooth Mate Silver Module [5]

The Bluetooth in the main controller acts as a master while the one in the wearable sensor acts as a slave. The communication protocol between the controllers is done in a master-slave structure and in a secured Wireless Personal-Area Network [6]. Figure 5 shows the master-slave structure of the Bluetooth Mate Silver.

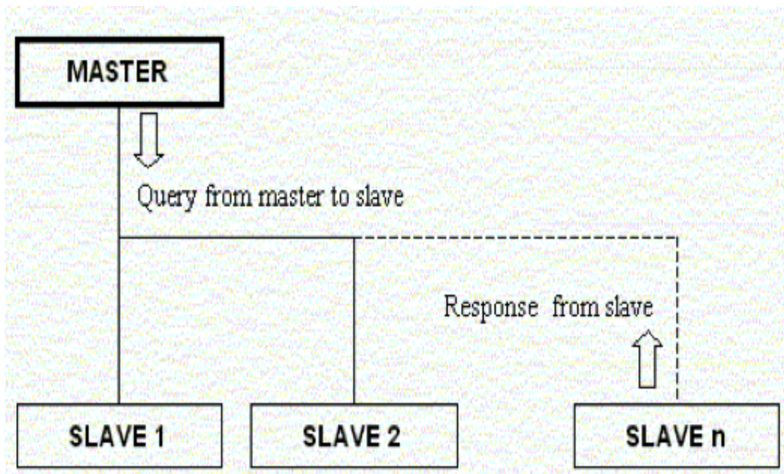


Figure 5: master-slave structure of Bluetooth Mate Silver [6]

The master Bluetooth controls the slaves and waits for its response to perform the desired action. The connection between the Bluetooth module and Arduino Uno is summarized in Table 2.

Table 2: The connection between the Bluetooth Module and the Arduino Uno

Bluetooth mate silver		Arduino Uno
VCC	↔	3.3V
GND	↔	GND
TXD	↔	SCL (Digital pin 1)
RXD	↔	SDA (Digital pin 2)

The technical specifications of the Sparkfun Bluetooth Mate Silver are summarized in Table 3.

Table 3: The technical specification of Bluetooth module [5]

Part/Item	**Bluetooth Mate Silver**
FCC Approved	Class 2 Bluetooth® Radio Modem
Low Power Consumption	25mA avg
Hardy Frequency Hopping Scheme	Operates in harsh RF environments like WiFi, 802.11g, and Zigbee
Connection Type	Encrypted connection
Frequency	2.402~2.480 GHz
Operating Voltage	3.3V-6V
Operating Temperature	-40 ~ +70C

3. Main and Peripherals Controller

The main controller consists of a microcontroller integrated with a GSM shield to enable sending SMS from the device. This unit is the heart of our *Smart Room Control System*, and it is responsible for bridging the connections between all the components of our system. The main controller will receive a command from the wearable sensor, which identifies the hand gesture movement. The command will then be passed to the peripheral's controller to control the device associated with the command sent by the user. The main controller will update our server regarding the status of the peripherals so that our smartphone application gets updated.

3.1 Switch Controller

This section provides a high-level overview design of both main and peripheral controllers. The section includes the electrical and physical design approaches taken to meet the functional specifications of the system. The main and peripheral controllers of the *Smart Room Control System* consist of a microcontroller, five LEDS, five Relays and a ULN2003. Figure 6 shows the high-level design showing the important components and their interface to each other.

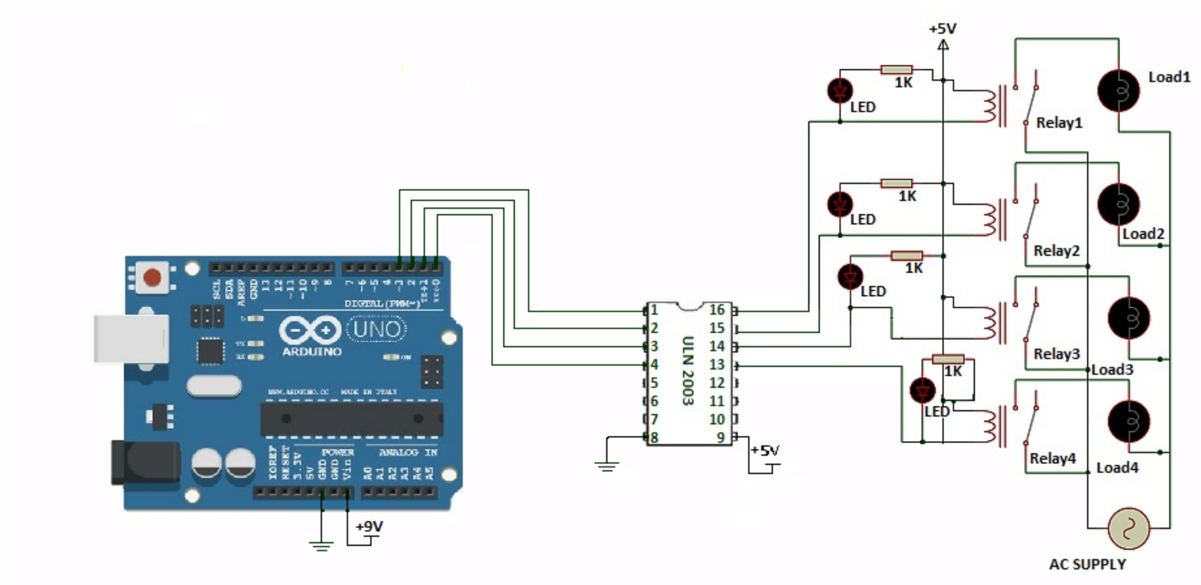


Figure 6: Switch Controller System Overview [7]

The main controller, illustrated in section 2.2, is used to provide interface between the wearable sensor and the peripherals. LEDs are used to indicate the relay output status.

3.1.1 Relays

Relays are electromagnetic devices that enable low power signals to control devices with high power. In other words, the electrical relays open or close a high power circuit, and they can handle most of the devices used in houses [8]. This specific type is chosen because of its relatively low cost and high reliability. Additionally, it satisfies our electrical requirements, which accepts 110/120 AC, as well as, 5V DC switching signal coming from the main controller. The selected relay is shown in Figure 7.

Figure 7: 5V DC Songle Electrical Relay [9]

The specifications of the electrical relay are summarized in Table 4.

Table 4: Specifications of the electrical relay [10]

Item	SONGLE Electrical Relay
Working Voltage	From 4.75V to 5.25V
Current	185mA
Maximum Pull-In Voltage	3.75V
Maximum Operation Time	15ms
Operating Temperature	From -25°C to 70°C

3.1.2 ULN2003A

MOTUSCONTROL utilizes ULN2003A transistor in the peripherals controller to work as a relay driver. ULN2003A is an array of seven NPN Darlington transistors, which will be the gateway connection between the main controller and the peripherals controller. This unit will be responsible for receiving the switching signals from the microcontroller included on the main controller, and then will pass it to the relay to switch a load. ULN2003A was selected because of its low price compared to other similar products and its open source designs. Figure 8 shows a top view of the pin connections of the ULN2003A transistor including a schematic of one of the channels.

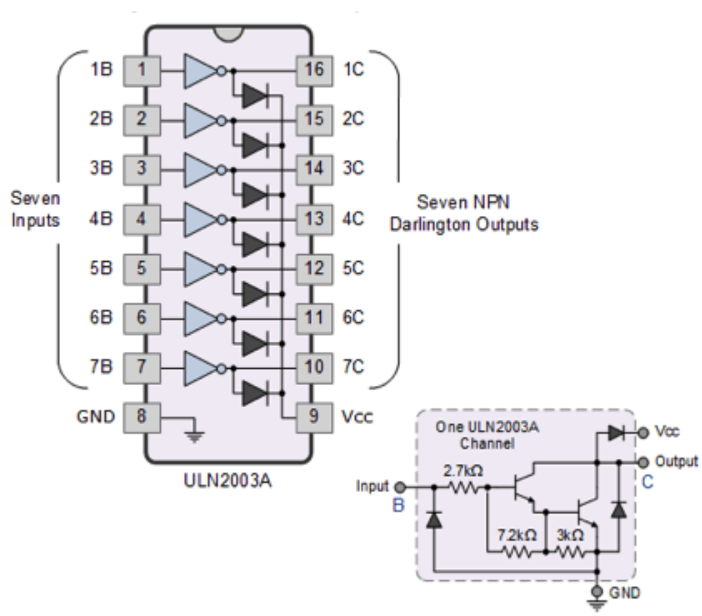


Figure 8: Top view of the pin connections and a schematic of one of the channels in ULN2003A [11]

Each driver is capable of delivering a high output voltage with common-cathode clamp diodes for switching inductive loads. The collector current output is 500mA with the ability of withstanding peak currents of 600 mA. The ULN2003 transistor requires a Vcc supply of 5 volts to operate, which will be fed from the Arduino Uno microcontroller included on the main controller [12].

3.2 Sensors

There are four sensors in our design for the room control feedback system. We have a light sensor, a heat sensor, a temperature sensor, and a humidity sensor. Each of these sensors are connected via 22-gauge copper wire to our Uno board via our breadboard in our first prototype. Our second prototype also uses 22-gauge copper wire, as well as a small PCB with the wires soldered in to connect the sensors to the main controller.

Our first sensor is the BH1750FVI light sensor chip (block diagram shown in Figure 9). This sensor is used to measure the ambient visible light where it is situated. There are only four pins required for simple continuous measurement of light, which we connect to our Arduino Uno. The power pins are connected to the board's power source pins, and the SCL (Serial Clock) and SDA (Serial Data Signal) pins are connected to the Arduino control pins. We will connect the pins initially via copper wires to a breadboard, then later on through a simple PCB. To facilitate usage of this chip, we will use an open source library built for interfacing with the chip via Arduino boards [13].

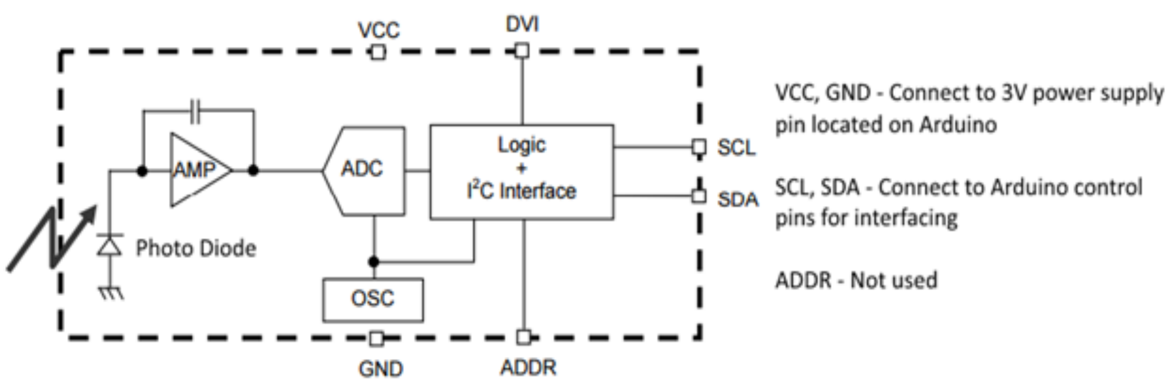


Figure 9: Block diagram of the BH1750FVI light sensor chip [14]

Our second sensor is a simple Phototransistor (part number EAPLP05RDEA1), which detects radiation in the 760-1100 nm range. This is the range for infrared radiation, allowing the Phototransistor to detect heat and flame. By connecting it with a resistor to the 5V pin of the Arduino, we can measure the output and use this to determine if there is a fire present. When there is, the relative Collector Current will rise sharply, allowing us to signal to the main controller that there is danger (see Figure 10). For our first prototype, this sensor will be directly on the breadboard, and for the second it will be mounted and soldered onto the PCB.

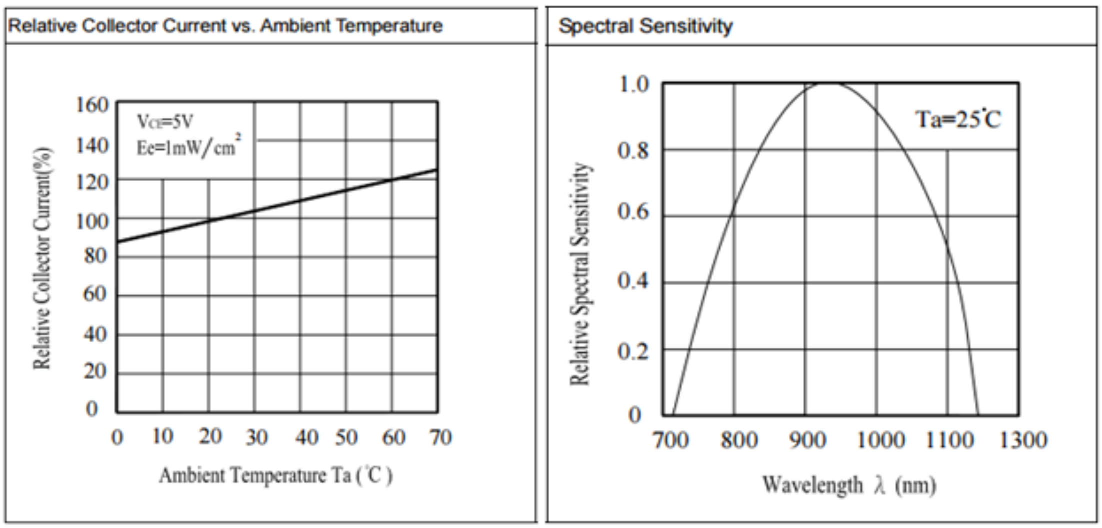


Figure 10: Electro-optical characteristics curves of the selected Phototransistor [15]

Our third sensor is the LM35 Precision Centigrade Temperature Sensor. This sensor measures ambient temperature in the -55°C to 150°C range. This is the chip which will return back to the Arduino a different voltage depending on the current temperature of the room (see Figure 11). It operates at 5 V, and so we will connect the Vs+ pin to the 5 V power supply pin on the Arduino. The Vout pin we will connect to one of the control pins of the Arduino. On our first prototype, this will be through a breadboard, and on the second, through a small PCB. This chip requires no special calibration.

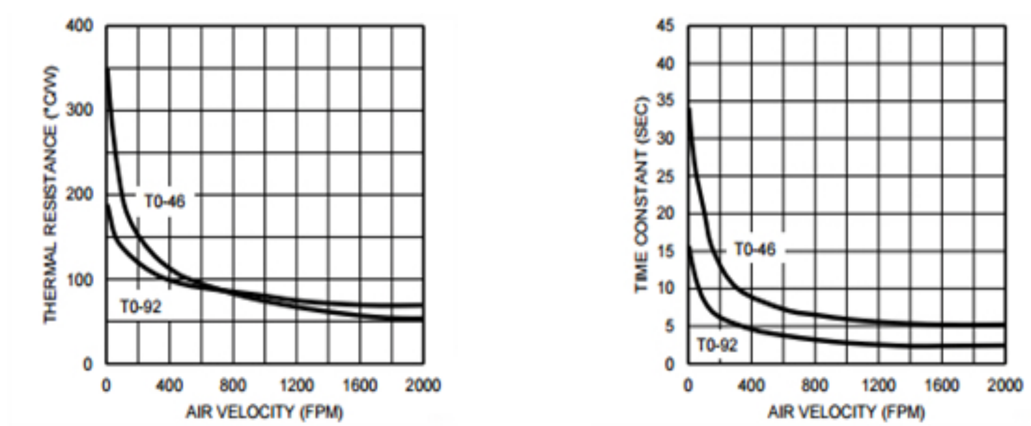
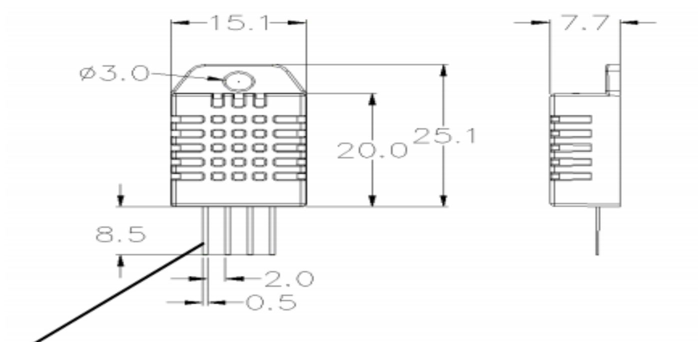


Figure 11: Typical characteristics of the selected LM35 Precision Centigrade Temperature Sensor [16]

Our fourth and last sensor is the DHT11 humidity sensor. This sensor measures the percentage of water in the surrounding ambient air. It can measure anywhere from 20-80% humidity with 5% accuracy. We can connect the Vcc pin to either the 3V or 5V power pins of the Arduino, and the ground pin to ground. The Data pin we will connect to one of the Arduino's data pins for the purpose of measuring the humidity. On our first prototype, this will be through a breadboard, and on the second, through a small PCB. This chip requires no special calibration.



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD----power supply
2	DATA--signal
3	NULL
4	GND

Figure 12: Dimensions (mm) and connection pins of the DHT11 sensor [17]

3.3 WiFi Module

Our design requires the use of a WiFi Shield to transfer the data from the main controller to the server and then to the phone application. Figure 13 shows the selected WiFi module, WiCC300 WiFi shield, which will be used in our design.

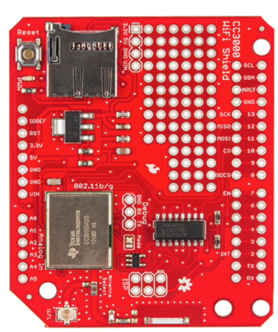


Figure 13: CC300 WiFi Shield [18]

We chose the CC300 WiFi shield because of its compatibility with the Arduino Uno. It has the same number of pins as Arduino Uno in the outer frame. As a result it can easily be placed on Arduino Uno, using its internal pins to operate the WiFi shield. Instead of using more UART communication methods, the WiFi shield utilizes an SPI interface allowing us to control the flow of data as desired. Moreover, it has a reset button, microSD card slot, and built in antenna. The antenna determines the range of frequency that can be used. It also has the option of installing an external antenna to expand the range of frequency, and also several libraries can be found and installed in Arduino Uno to run the WiFi shield [18]. Wiring Ardunio Uno with the WiFi shield is summarized in Table 5.

Table 5: Connection between WIFI Shield and Arduino Uno

CC300 WiFi shield		Arduino Uno
VCC	⟷	4.5 - 12 V
GND	⟷	GND
MOSI	⟷	Digital pin 11
MISO	⟷	Digital pin 12
CS	⟷	Digital pin 10
INT	⟷	Digital pin 2
SCK	⟷	Digital pin 13
EN	⟷	Digital pin 7

The specifications of the CC300 WiFi Shield are summarized in Table 6.

Table 6: The technical specification of WIFI CC300 Shield [18]

WIFI Type	**CC330 WIFI Shield**
Host Interface	SPI @ 16MHZ
Throughput (TCP)	~4Mbps
Security Mode	WEP, WPA/WPA2
Board	Prototyping Area
Onboard	WIMAX Antenna
Optional	External antenna connection

3.4 GSM

GSM SIM800L Module is chosen in the project design in order to achieve the texting message functionality. This model is chosen for the project because it meets Canadian GSM frequency standards. Figure 14 shows the GSM module using in the project.

Figure 14: GSM Module and Antenna [19]

Figure 15 illustrates the components of the selected GSM module.

Figure 15: Components of GSM Module [19]

The GSM module will be attached to the main controller and will be programmed in a way that a text message will be sent to user's cell phone after receiving critical event alert (e.g. fire alarm).

The selected GSM module meets the Canadian GSM transmitting frequency. It supports GSM 850/900/1800/1900 MHz bands (Canadian GSM bands are 850/1900 MHz) [20]. Some other features of the GSM module are listed in Table 7.

Table 7: Key Features of SIM800L GSM Module [19]

Feature	Implementation
Power Supply	3.4V ~ 4.4V
Frequency Bands	Quad-bands: GSM 850 / EGSM 900 / DCS 1800 / PCS1900 Compliant to GSM Phase 2/2+
Transmitting power	Class 4 (2W) at GSM 850 and EGSM 900 Class 1 (1W) at DCS 1900 and PCS 1900
Temperature range	Normal operation: -40 ~ +85 C
Physical characteristic	Size: 15.8*17.8*2.4mm Weight: 1.35g
Real Time Clock	Support RTC

The pins used to connect the GSM module with Arduino Uno are listed Table 8.

Table 8: Connection between the GSM module and the Arduino Uno

GSM		Arduino Uno
VCC	↔	4.4 V
RST	↔	Reset
TXD	↔	SCL (Digital pin 1)
RXD	↔	SDA (Digital pin 2)
GND	↔	GND

3.5 Physical Design

The main controller and peripheral controller will be secured in a protective case that is not electrically conductive to prevent any harm to the user. The two controllers will be placed 5 cm distance away to avoid any electrical or thermal problems. The case has pin leakage, which provides an easy access to other components, and it also has ventilation holes to allow airflow and heat dissipation.
The case is designed using Solidworks, a 3D CAD design software, as shown in Figure 16.

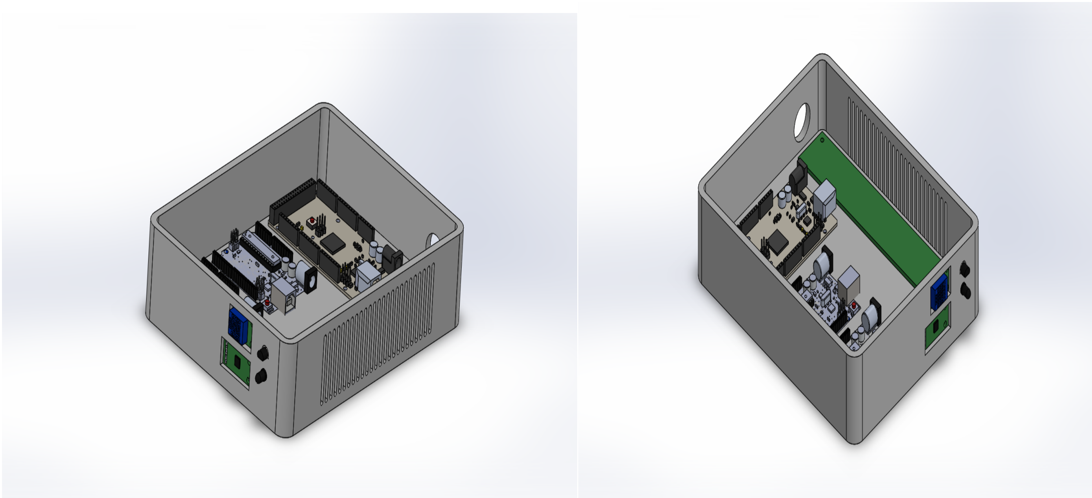


Figure 16: The designed case for the main and peripherals controllers

3.6 Functional Requirements

A summary of the functional specifications that the design of the main and peripheral controller satisfies is included in Table 9.

Table 9: Main and Peripherals controller satisfied functional requirements

Requirement	Description	Solution/Justification
[R12-ii]	The product shall be operated with a power supply of 120/110V, 60Hz AC.	Peripherals will be connected to relays that will be connected to electrical sockets. Electrical sockets in Canada usually supply the required power.
[R23-i]	The microcontroller shall receive motion data from the wearable sensor.	Main controller and wearable sensor contains Bluetooth modules that will communicate to transmit data.
[R24-i]	The microcontroller shall communicate with the peripheral's controller based on the received data from the wearable sensor.	ULN2003A is responsible for receiving the switching signals from the main controller, and then will pass it to the relay to switch a load.
[R30-i]	The microcontroller shall send SMS to the caretaker of the system user in case of emergency when the internet is not working.	GSM module SIM800L is connected to the main controller and will send SMS to the caretaker.
[R34-iii]	The main controller must be secured by an enclosed box and placed on the wall with the main room switches.	The designed case in Figure 16 will contain the main controller.
[R56-ii]	Peripheral's controller shall be connected with smoke sensor, motion sensor and lights over wires.	We have a light sensor, a heat sensor, a temperature sensor, and a humidity sensor connected via 22-gauge copper wire to our Uno board via our breadboard in our first prototype.

4. Wearable sensor

The wearable sensor is responsible for obtaining the coordinates of the hand movement using the Inertia Measurement Unit sensor, MPU-6050. The data will be sent via Bluetooth, illustrated in Section 2.3, to the main controller to control various peripherals connected to our *Smart Room Control System*. The circuit implementation of the wearable sensor is shown in Figure 17.

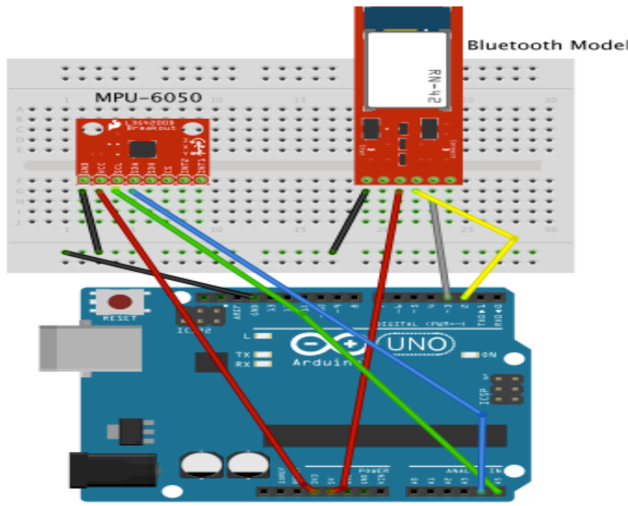


Figure 17: The circuit implementation for the wearable sensor

4.1 Inertial Measurement Unit

An inertial measurement unit is widely used in self-balancing robots and smartphones to indicate changing orientation. The selected inertia measurement unit, MPU-6050, is chosen since it is a 6 DOF (Degrees of Freedoms) containing a 3-axis gyroscope and a 3-axis accelerometer in a single package to measure both the changing tilt angle and the rate of change in angle orientation. Additionally, the MPU-6050 is very accurate comparing the other IMU sensors [21]. The SparkFun MPU-6050 is shown in Figure 18.

Figure 18: SparkFun MPU-6050 Model [22]

The main purpose of the MPU-6050 is to continually sense the hand movement and to transmit this movement to the main controller in form of X, Y, and Z coordinates. The main specifications of MPU-6050 are summarized in the Table 10.

Table 10: SparkFun MPU-6050 Specifications

Part/Item	MPU-6050
Input Voltage	2.375V-3.46V
Serial Interfaces Supported	I2C Protocol
Operating Current	3.9 mA
Sensitivity	Up to 131 LSBs/dps
Gyro full-scale range	±250, ±500, ±1000, and ±2000 dps
Accelerometer full-scale range	±2, ±4, ±8, and ±16 g
Dimensions	25.5 x 15.2 x 2.48mm
Output	Digital

For the prototype design, the arm's angular movement was used for gesture recognition by the IMU sensor. The arm's angular shift is detected by the gyroscope, and its relative change is detected by the accelerometer. The arms position and orientation was mapped in 3D space to determine gesture patterns that will serve as input gestures to execute a predefined action. When the users tilt their hands in any orientation, the MPU-6050 senses the movement and transmits this movement to the main controller. The 3D representation of the data collected by the IMU sensor is shown in Figure 19.

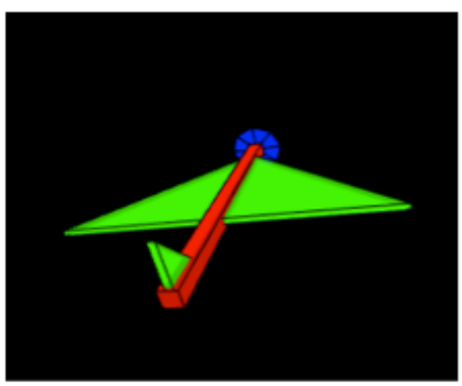


Figure 19: 3D modeling of the data collected by MPU-6050

The data collected from the IMU sensor is processed in our embedded system, Arduino Uno. The compiling software used is Arduino sketch.

MOTUS CONTROL

4.2 Physical Design

The design of the wearable sensor, which will be worn by the user, has a few stages of design. Figure 17 shows the design of the first prototype, which we will construct. For a final, marketable product, we have researched and found some suitable designs, which would lend them well to our project.
The physical design of the wearable sensor depends highly on the shape and size of the other components of our sensor. To ease the time and effort of prototyping, both our first and second prototypes use Arduino boards to handle the outputs from the MPU-6050, and send them via a Bluetooth module to our main controller. These three modules take up a lot of space, limiting our design choices and the comfort of the user.

The first prototype will use the largest board, the Arduino Uno. For this prototype, we chose not to construct any casing, because of the cost and time involved for a part, which would not be part of the final product. Instead, we sew a small amount of fabric to the bottom of the board, via the 4 standoff holes, to insulate it from the top of the user's hand. Then, we attach this fabric to two 40cm long strips of Velcro that are wrapped around and attached to the wrist. We chose Velcro for this purpose because it is inexpensive, easy to use, and strong enough to easily hold on the board. This design will mainly be used to test and troubleshoot the motion controls while we work on the final product.

The second prototype will use Arduino Nano instead of the Arduino Uno. This is because Arduino Nano is much smaller than the Arduino Uno. The second prototype uses a case made out of hard plastic to carry the Arduino and the other required chips. For this prototype, we construct a small case out of hard plastic to surround and protect the board. We chose to construct a case for this version to properly house the various components, and provide a more complete product for demo purposes. This case attaches to the board via 4-40 thread nylon standoffs, via the 4 standoff holes at each corner of the board. The full case also uses 40cm long Velcro strips to attach to the user's wrist securely. The Velcro is attached to the bottom of the case as shown in Figure 20.

The Bluetooth module and the MPU are also housed in the case. They are attached to the case in a similar method to the Nano, via their two standoff holes, using 4-40 thread nylon standoffs. We connect them to the actual Arduino via insulated wires and a small PCB. We use a small PCB to simplify the wiring between the Arduino and the other parts. In our second prototype, we also have space in the case for a 9 Volt battery, which provides power for the electronics. This is attached with a simple battery snap, which is soldered on to the board.

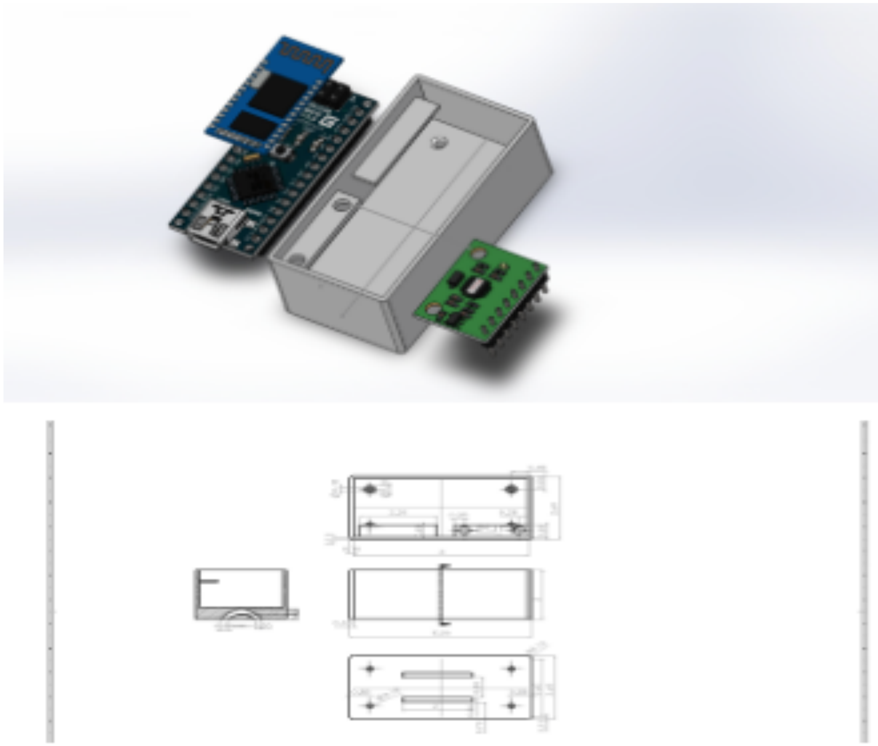


Figure 20: The designed case for the second prototype

For a final, marketable design, we have created a few concepts that would be a lot less cumbersome than our prototypes. If we had the ability to design our product from scratch, with only the features we need, we could greatly reduce the size of the electronics. Design A in Figure 21 is similar to a small smart watch. It is attached via a plastic or leather strap to the wrist. Design B has thinner electronics spread out over an elastic wristband. One such band that would be suitable for our purposes would be the GE-Fusion Band, a wearable armband with embedded motion sensing technology [23].

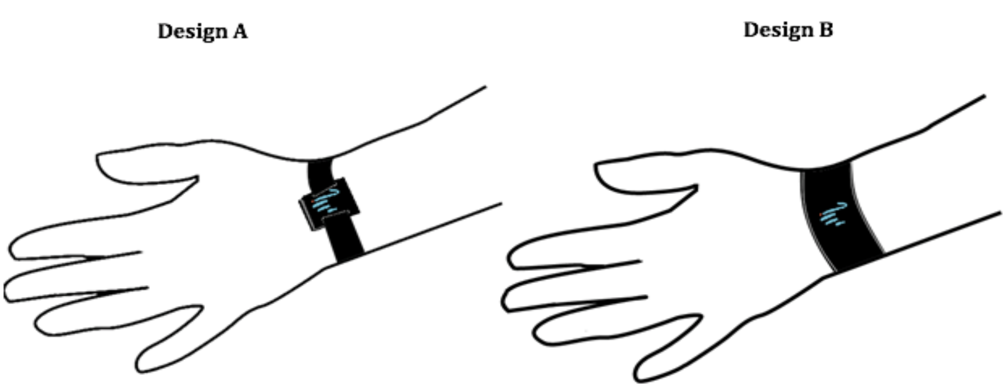


Figure 21: The final marketable design case for the wearable sensor

4.3 Battery

The first prototype of the wearable sensor is created to demonstrate the functionality and the usability of the product. The first prototype wearable sensor device is powered by an Energizer NH22-175 Rechargeable 9V Battery. This battery is mainly chosen because it can power the wearable sensor directly and does not require a voltage regulator. Figure 22 shows the battery used to power the device.

Figure 22: Energizer NH22-175 Rechargeable 9V Battery [24]

In the second prototype model, two 3V rechargeable lithium coin cell batteries will be used to power the wearable sensor. They are connected in series to provide a voltage supply of 6V to meet the input voltage requirement of the Arduino Nano. The batteries are placed in a designed plastic battery holder that is glued inside enclosure for stability to provide a safe and secure product to the user.

4.4 Functional Requirements

A summary of the functional specifications that the design of the wearable sensor satisfies is included in Table 11.

Table 11: Wearable sensor design satisfied functional requirements

Requirement	Description	Solution/Justification
[R37-i]	The sensor shall be able to be worn on the user's hand or wrist.	In all of our prototypes the sensor will be worn, as can be seen in figure XX
[R39-i]	The sensor shall be able to send motion data back to the main controller.	Arduino board will handle the outputs from the MPU, and send them via a Bluetooth module to our main controller
[R49-ii]	The sensor shall be able to fasten securely on the user's hand or wrist and only deliberately be removed.	We chose Velcro strips for this purpose because it is strong enough to easily hold on the board

5. Software System

The software component of the *Smart Room Control System* is based on Android app as well as a server that communicates between the application and the main controller.

5.1 Phone Application

The phone application will allow the user to control connected home appliances and get the working status of the home appliances. Moreover, the user will receive a phone notification in the case of some peripherals, such as the flame sensor, being activated.

The phone application will be developed using JAVA programming language and the official Android IDE Android Studio. This application will be able to connect with server through WiFi or 4G network. It can download data from server and upload data to server.

The phone application will have a starting page, which allows user to login with existing credentials, or register with associated product. The user will go into the main menu once their credentials are correct. In the main menu, user will be able to add new connections to home appliances, control home appliances, check the status of home appliances and remove existing connections of home appliances. Furthermore, the phone application will have a help center providing a detailed step-by-step user manual. The help center will assist user to operate the phone application. Moreover, the phone application will have a setting center, which will allow the user to adjust the setting of notifications. Figure 23 shows the flowchart of the phone application.

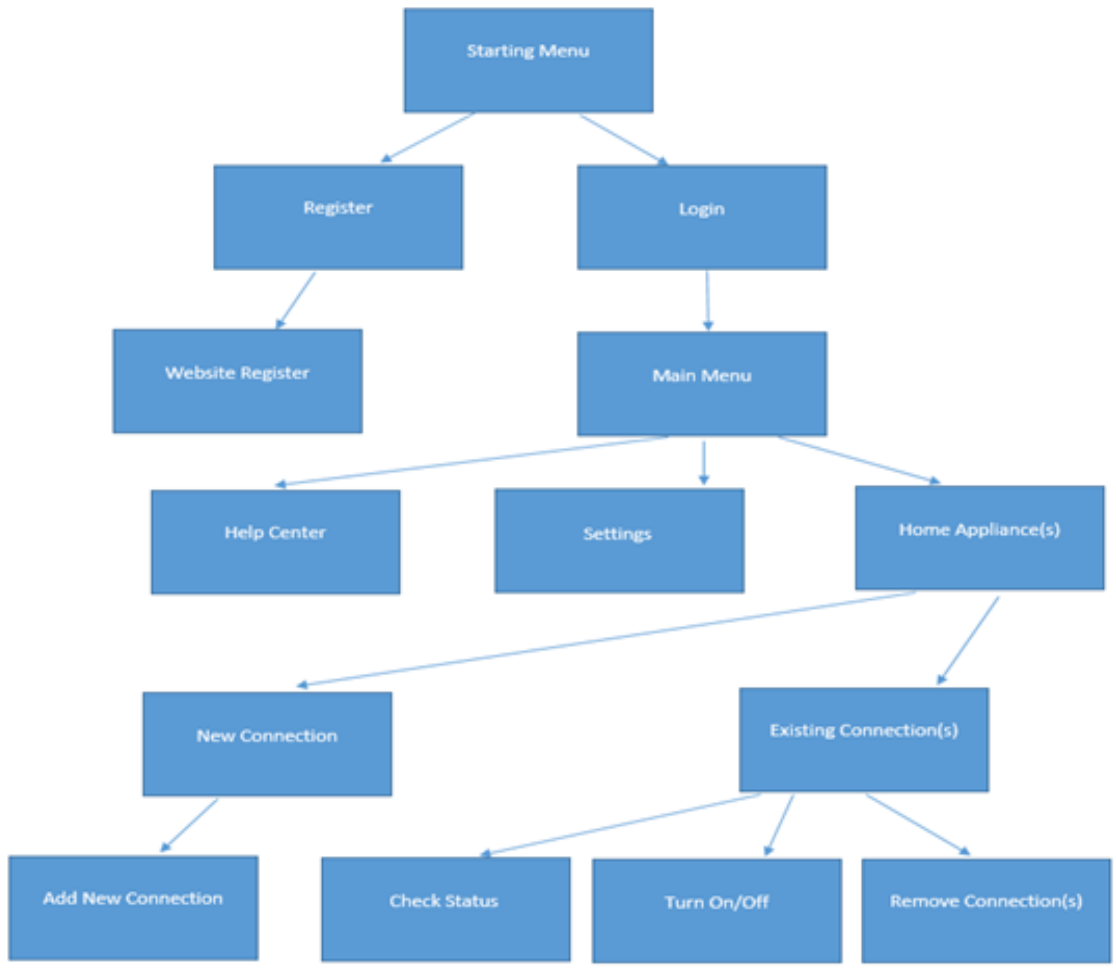


Figure 23: Flowchart of the phone application

5.2 Server

The server should be able to communicate with both the phone application and the main controller. The server will receive on/off commands and check status commands from the phone application and send them to the main controller. Then, it gets the data from the main controller containing the working status of attached peripherals appliances and sends it back to phone application. The data transmission indication is shown in Figure 24.

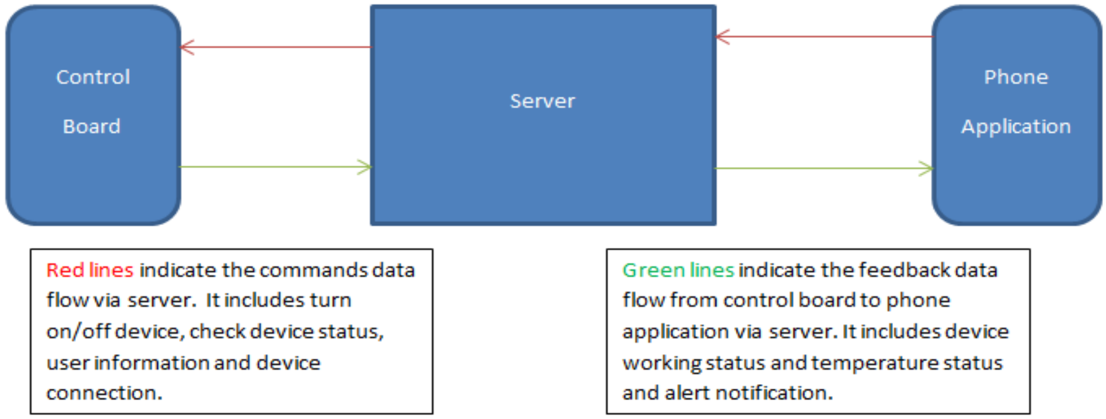


Figure 24: Data transmission indication

Table 12 summarizes the information sent from/to the server.

Table 12: The Data Content Transmitted via Server

Information sent by server to phone application
Device working status
Temperature status
Alert notification
Commands sent by phone application to server
Turn on device
Turn off device
Check device status
User info
Device connection

The server will be developed by using Apache Tomcat, which contains an HTTP server and can be installed as web server. Data transmission will follow the TCP/IP protocol. The phone application sends request messages to the server and then gets the response message from server. The data transmission between server and Phone Application is shown in Figure 24.

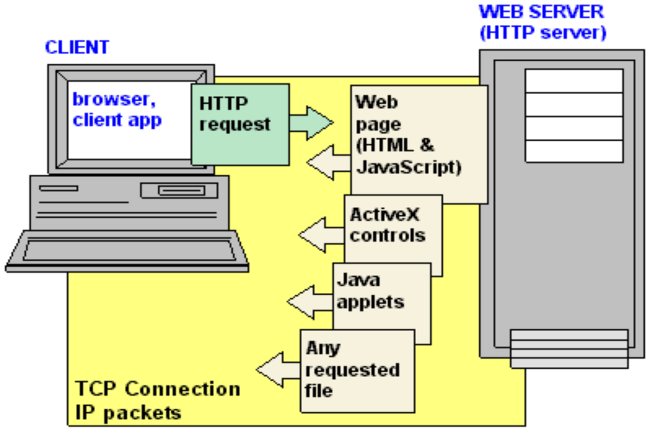


Figure 25: Data Transmission between Server and Client (Phone App) [25]

The server also has a database to store user information. In our *Smart Room Control System*, Microsoft Access will be used to implement database server since the data size is relatively small. All user information should be stored and the working status of every connected home appliance should be stored as well.

5.3 Functional Requirements

A summary of the functional specifications that the design of the software system satisfies is included in Table 13.

Table 13: Software design satisfied functional requirements

Requirement	Description	Solution/Justification
[R66-i]	The server shall be able to transmit data to mobile application from the main controller.	Tomcat software is used to set up a server to follow the TCP/IP protocol to achieve the data transmission.
[R67-i]	The server shall be able to transmit data from mobile application to the main controller.	Tomcat software is used to set up a server to follow the TCP/IP protocol to achieve the data transmission.
[R68-ii]	The server shall be able to store data.	Microsoft Access is used to create the database to store data.
R[69-i]	The mobile application shall allow user authentication.	The mobile application is designed to prompt user to enter credentials before use.
R[72-i]	The application shall notify the user when it receives relevant data from the server.	The application algorithm is designed that it can send alert notification once received text message from GSM module.

6. System Test Plan

To ensure that all functional requirements are satisfied in the *smart room control system*, *MOTUSCONTROL* has decided to undergo series of tests as per the following subsections.

6.1 Software Testing

MOTUSCONTROL will perform unit testing for the software required inside *the smart room control system,* as well as, the smartphone application. Unit testing is a technique used for testing certain functions and units of the code, which allows to easily preserving the functionality of the code and reduces the introduction of bugs. This approach will help us identify any logical or algorithmic problems we might encounter during writing our code. Every new function that we write will have a corresponding test case particularly dedicated to check its performance. The accumulation of these test cases for every new function that we write will be a test suite. The test suite is a package of multiple test cases that we can run at any time to ensure the intended functionality and optimal performance of our code as a whole. Figure 26 illustrates the functions that are included in the test case class.

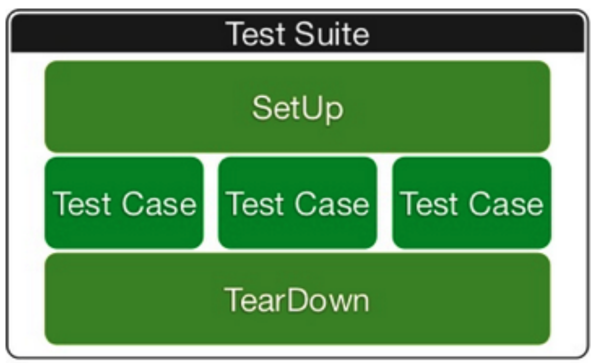


Figure 26: Test Suite Diagram [26]

6.2 Hardware Testing

The Smart Room Control System also depends on the success of the following units: the wearable sensor is able to generate different coordinates, the main controller is able to detect the coordinates sent by the wearable sensor, and the switch controller is able to receive the switching signal and perform the operation. All of those systems are dependent on the reliability of each hardware device and their ability to send and receive data. Therefore, hardware systems will need to have their components manually tested, and we will conduct software testing to ensure reliable transmissions between subsystems.

6.3 Integration Testing

Unit testing for the *Smart Room Control System* will only test specific modules, independent of the system as a whole. Since *MOTUSCONTROL* solution consists of many modules continuously interacting with one another, we cannot rely solely on unit testing. For example, our sensor modules may successfully pass unit tests to ensure data is being transmitted. However, without testing the sensor modules with our data access module, we cannot ensure the data is being received without error or packet loss. Unit testing alone cannot ensure data integrity across our system. We will need to implement integration testing to thoroughly test these "interface" scenarios – specifically the ones where modules interact.

7. Conclusion

MOTUSCONTROL is creating the *Smart Room Control System* to give disabled people and elders the ability to control devices that are out of their reach. The solution that *MOTUSCONTROL* provides is to provide control of the devices and peripherals surrounding the system's user by using simple hand gestures. Our system will not only improve the quality of life for the disabled, but also their safety. No longer will they need to stretch to reach a thermostat or light switch, or worry about hearing their fire alarm. The ability to send information via wireless and cellular to a caretaker ensures that in the case that something goes wrong, our system ensures a quick response.

The design specifications presented in this document define the proof-of-concept model of the *Smart Room Control System*. This document serves as a guideline to the development and testing stages of our system. Details regarding the hardware and software components of the product are included, as well as the justifications of the current design choice. All design specifications will be tested to ensure that the functionality of the *Smart Room Control System* performs as expected. We expect to successfully complete the first prototype by mid April.

MOTUS CONTROL

Reference

[1] Bestbuy.ca, "Modems : Hubs, Switches & Modems - Best Buy Canada", 2016. [Online]. Available: http://www.bestbuy.ca/en-CA/category/modems/21105.aspx. [Accessed: 15- Feb- 2016].

[2] N. Heinz, "Relay - HomoFaciens", *Homofaciens.de*, 2016. [Online]. Available: http://www.homofaciens.de/technics-base-circuits-relay_en_navioff.htm. [Accessed: 13- Feb- 2016].

[3] "Arduino - ArduinoBoardUno", Arduino.cc, 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed: 7- Mar- 2016].

[4] Arduino.cc, 2016. [Online]. Available: https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg. [Accessed: 6- Mar- 2016].

[5] S. Silver, A. R3, A. 5V/16MHz, A. R3, L. Board and S. Gold, "SparkFun Bluetooth Mate Silver - WRL-12576 - SparkFun Electronics", *Sparkfun.com*, 2016. [Online]. Available: https://www.sparkfun.com/products/12576. [Accessed: 28- Feb- 2016].

[6] Pacontrol.com, "PAControl.com - Modbus", 2016. [Online]. Available: http://www.pacontrol.com/Modbus.html. [Accessed: 28- Feb- 2016].

[7] B. Malik, "Relay driver circuit using uln2003 and its applications", Microcontrollers Lab, 2015. [Online]. Available: http://microcontrollerslab.com/relay-driver-circuit-using-uln2003/. [Accessed: 06- Mar- 2016].

[8] W. Storr, "Electrical Relay and Solid State Relays", Basic Electronics Tutorials, 2013. [Online]. Available: http://www.electronics-tutorials.ws/io/io_5.html. [Accessed: 08- Mar- 2016].

[9] "What is an Electromagnetic Relay?", *wiseGEEK*, 2016. [Online]. Available: http://www.wisegeek.com/what-is-an-electromagnetic-relay.htm. [Accessed: 02- Mar- 2016].

[10] "Grove - SPDT Relay(30A) - Wiki", *Seeedstudio.com*, 2016. [Online]. Available: http://www.seeedstudio.com/wiki/Grove_-_SPDT_Relay(30A). [Accessed: 06- Mar- 2016].

[11] W. Storr, "Output Interfacing Circuits Connect to the Real World", *Basic Electronics Tutorials*, 2015. [Online]. Available: http://www.electronics-tutorials.ws/io/output-interfacing-circuits.html. [Accessed: 7- Mar- 2016].

[12] "ULN200x, ULQ200x High-Voltage, High-Current Darlington Transistor Arrays", *TEXAS INSTRUMENTS*, 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/uln2003a.pdf. [Accessed: 10- Mar- 2016].

[13] "claws/BH1750", GitHub, 2015. [Online]. Available: https://github.com/claws/BH1750. [Accessed: 08- Mar- 2016].

[14] ROHM Semiconductor, "Digital 16bit Serial Output Type Ambient Light Sensor IC", BH1750FVI datasheet, Nov. 2011 [Revised Nov. 2014].

[15] EVERLIGHT Americas, "5mm Phototransistor T-1 3/4", EAPLP05RDEA1 datasheet, Oct. 2013.

[16] Texas Instruments, "LM35 Precision Centigrade Temperature Sensors", LM35 datasheet, Aug. 1999.

[17] Aosong ELectronics Co., Ltd "Digital-output relative humidity & temperature sensor/module", AM2303 datasheet, Mar. 2016.

[18] S. CC3000, W. ESP8266, B. Straight, A. R3, A. Shield and 2. RP-SMA, "SparkFun WiFi Shield - CC3000 - DEV-12071 - SparkFun Electronics", *Sparkfun.com*, 2016. [Online]. Available: https://www.sparkfun.com/products/12071. [Accessed: 28- Feb- 2016].

[19] "SIM800H Hardware_Design", *A company of SIM Tech*, 2016. [Online]. Available:https://www.adafruit.com/datasheets/sim800h_hardware_design_v1.00.pdf. [Accessed: 6- Mar- 2016].

[20] "The big three Canadian network frequencies | Koodo Community Forum", *Koodo Community Forum Community*, 2016. [Online]. Available: https://community.koodomobile.com/koodo/topics/the_big_three_canadian_network_frequencies. [Accessed: 10- Mar- 2016].

[21] "Arduino Playground - MPU-6050", *Playground.arduino.cc*, 2016. [Online]. Available: http://playground.arduino.cc/Main/MPU-6050. [Accessed: 02- Mar-2016].

[22] S. MPU-6050, "SparkFun Triple Axis Accelerometer and Gyro Breakout - MPU-6050 - SEN-11028 - SparkFun Electronics", *Sparkfun.com*, 2016. [Online]. Available: https://www.sparkfun.com/products/11028. [Accessed: 02- Mar- 2016].

[23] J. Cannan and H. Hu, "A Wearable Sensor Fusion Armband for Simple Motion Control and Selection for Disabled and Non-Disabled Users.", *4th Computer Science and Electronic Engineering Conference (CEEC)*, pp. 216-219, 2012.

[24] “ENERGIZER NH22-175”, *ENERGIZER*, 2016. [Online]. Available: http://data.energizer.com/PDFs/nh22-175.pdf. [Accessed: 10- Mar- 2016].

[25] "Web server Definition from PC Magazine Encyclopedia", *Pcmag.com*, 2016. [Online]. Available: http://www.pcmag.com/encyclopedia/term/54342/web-server. [Accessed: 10- Mar- 2016].

[26] R. Media, 'Unit testing in iOS featuring OCUnit, GHUnit& OCMock', Slideshare.net, 2015.[Online]. Available: http://www.slideshare.net/robotmedia/unit-testing-in-i-os-15491405. [Accessed: 13- Oct- 2015].

Appendix: Test Plan

Main and peripheral controller

Test Object	Status (Pass/Fail)	Date	Comment
Main controller is able to receive data from the wearable sensor via Bluetooth.			
Main controller is able to receive data from server via WiFi.			
Main controller able to pass the received data to the peripherals controller.			
Peripherals controller is able to switch on/off the device corresponding to the signal received from the main controller.			
Main controller is able to send sensors data to server via WiFi.			
Main controller is able to send a SMS to the user.			

Software System

Test Object	Status (Pass/Fail)	Date	Comment
The server is able to transmit data to mobile application from the main controller.			
The server is able to transmit data from mobile application to the main controller.			
The application shall be able to provide the operating status of each connected home appliance.			

Wearable Sensor Unit

Test Object	Status (Pass/Fail)	Date	Comment
The sensor is able to collect the hand-motion data.			
The sensor is able to send the hand-motion data to the main controller via Bluetooth.			
The sensor shall be able to fasten securely on the user's hand or wrist.			