EE175AB Final Report

Intelligent Home Control System

Department of Electrical Engineering, UC Riverside

Project Team Member(s)	Yancong Deng, Yujie Cao, Qixin Huang
Date Submitted	March 18, 2016
Section Professor	Dr. Ping Liang
Revision	Revision 1.0
URL of Project Wiki/Webpage	https://www.youtube.com/watch?v=8kTkS-dI4EE
Permanent Emails of all team members	Yancong Deng: yancongdeng@hotmail.com Qixin Huang: vallenhqx@gmail.com Yujie Cao: caoyujiejay@gmail.com

Summary

This report presents the research and design of a multifunctional Internet Connected Home Control System. In this report, we are going to show the design procedure, implementation, encountered problems, and final results in detail.

Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
0.5	First draft	Yancong Deng, Yujie Cao, Qixin Huang	03/04/2016	
1.0	Correction on Section 1 – 10.	Yancong Deng, Yujie Cao, Qixin Huang	03/18/2016	

Table of Contents

1 * Executive Summary

The overall goal of our project is to design an Intelligent Home Control System (IHCS), which allows users to control electrical devices in a household through the Internet. Our system is able to control a variety of devices, including the main door lock, garage door, air conditioner, light bulbs, and water sprinkler. This project is our attempt to introduce a new living style that is not only convenient, but also energy efficient. We want this project to provide a maximum satisfaction in a living environment. Users are able to conveniently obtain real-time housing information and control all electrical devices with a simple touch of a finger.

In our design of the IHCS, there is a main controller connecting with all electronic devices and sensors. The main controller acted as the major Internet server, which is connected to WIFI router within the area of the house. It is able to respond to requests received from the Internet within 2 seconds. The main controller communicates with the slave controller, which locates near the main door, with a range of 50 feet. The Bluetooth 4.0 module is used on the slave controller to enable the communication with Android devices. Any Android devices with Bluetooth 4.0 compatibility should detect the Bluetooth module within a range of 50 feet.

In IHCS, we introduced the Automatic Door Lock system. We use Bluetooth communication between Android devices and Bluetooth module on Microcontroller to accurately identify the unique devices allowed access. The system is able to calculate the approximate distance between Android devices and Microcontroller by analyzing the Received Signal Strength Indicator (RSSI). With the combination of the information and the list of authorized users, IHCS is able to control the door lock to open automatically when an authorized user is present within the proper range of the main door. The auto connection with the Android application and the Bluetooth module should be less than 5 seconds.

Another key feature in our system is the Internet Connected Real-time Monitor and Control System. The main controller is programed to control electrical devices, collect sensor data, and host a Web server. The web server is programed to display real-time housing data and control panel on the website. It allows users to send requests from HTTP website for controlling actions or monitoring purpose. The web server is able to respond within 2 seconds.

Our system also enables the ability of Home Automation. IHCS is able to set up automatic control system based on users' desires with corresponding the collected sensor data. For example, IHCS can be set up an automatic lighting system according to the instant brightness and detected motion inside an area. IHCS can also determine optimal humidity level of the grass and automatically control the water sprinkler to water the soil to the desired level.

As the test result, our system used Universal Asynchronous Receiver/Transmitter (UART) to carry out the communication between the server controller and the slave controller within a distance of 50 feet. The initial response time of the web server is between 0.5 to 1 second, and the refresh response time is between 1 to 3 seconds. Moreover, the Bluetooth signal can be proper detected in the range within 60 feet. The Bluetooth can obtain connection, if it is in proper range, between 5 to 7 seconds, which is longer than our original expectation.

After a comprehensive implementation and testing procedure for our system, we obtained satisfactory results and performance data of our system.

Yancong Deng is responsible for this section.

2 * Introduction

2.1 * Design Objectives and System Overview

(Yancong and Yujie wrote the Design Objectives and System Overview)

IHCS belongs to the category of Internet of Things (IoT). The Internet of Things connects devices to the Internet, enables data collection, and wireless control over the Internet. IoT allows people to interact with electronic devices around them through the Internet. It provides a new way for human beings to feel and control the physical surrounding.

In fact, IoT has a huge potential market. The research firm IDC predicts that the global IoT market will grow to 1.7 trillion dollar by the end of 2020. Similarly, Cisco, a well-known Information Technology (IT) company, also predicts that there will be more than 50 billion devices connected to the Internet in 2020. With a large growing population of smart mobile devices, the IoT will build a firm foundation and experience an explosive growth in following years.

The objective of our project is to design and implement a prototype of a Home Control System with ability to control multiple electrical devices and monitor real-time information in a house. The goal of our project is not only to provide a more convenient living experience, but also enhance housing security and energy efficiency. The automation ability of IHCS can make up our careless mistakes in people's daily lives. For instance, when a user forgets to close lock the door, our system is able to detect the unlocked door and close it automatically.

There are two major control units in IHCS, which are the main controller and slave controller. The main controller connects to the Internet through WIFI module while the slave controller operates through Bluetooth module and communicates with Android application.

Our system is able to control varieties of electrical devices, such as light, air conditioner, sprinkler, electrical door lock, and garage door. IHCS monitor the house by collecting data from different kinds of sensors, such as ultrasonic sensor, motion sensor, brightness sensor, humidity sensor, and temperature sensor. The system also simultaneously updates the real-time condition of the household on the web server.

The major control system of IHCS can be divided into two parts. One is the automatic control of the electrical door lock. The automatic control function is achieved by using Bluetooth communication and Android application. Android devices communicate with mircocontroller through Bluetooth protocol. When a authorized Android device presents within 10 feet from the main door, the system is able to identify the device and open the door automatically. Another major control system is the web server, which controls the rest of components. The main controller collects real-time data from sensors and displays on the website. Users is able to easily control all devices and review their current condition on the website.

2.2 * Backgrounds and Prior Art

(Yancong Deng and Yujie Cao wrote this section.)

There are many IT companies making similar Home Control System or services in the market. For example, Time Warner Cable has a product named "Intelligent Home", which provides monitored protection, living indoor and outdoor cameras, lights and thermostat adjusting system, and door security. Control4, a Home Automation company, focus on the smart home system, which includes Smart Home and Home Theater. It provides services to control thermostats, door lock, window curtains. Moreover, Creston, a engineering company in London, also provides home automation solution for customers including lighting control, shading control, video, and audio.

In comparison with the above Smart Home System, our system has several advantages. IHCS is more user friendly. Our system can be customized to control varieties of devices according to customers' desire. IHCS is able to add new control system to a commercial product without disabling the existing system.

Our control system gives a view of the monitored home environment, including insights on the home temperatures, electrical devices, and humidity. Customers are able to view this information online. Meanwhile, IHCS can provide specific automation design that is suitable for the customer. Our system can develop automation programs to control the devices based on customers' needs.

However, there are two shortcomings of our control system. First, all companies above have developed mobile applications for users to access to their Smart Home system while ICHS only used Android application for door lock automation. Second, our system does not have the audio and Home Theater control.

2.3 * Development Environment and Tools

(Yancong Deng wrote the Development Environment and Tools.)

The major program in Arduino Mega and Bluno was developed in Arduino Integrated Development Environment (IDE). The Arduino IDE used C/C++ programming language. The Arduino program is compiled using AVR-G ++ compiler. The Arduino program also used the buildted in Arduino Serial Monitor in Arduino IDE for debugging purpose,.

The Door Lock Control circuit and relay circuit was designed and implemented in Printed Circuit Board (PCB) in the electrical engineering lab in University of California, Riverside (UCR). The circuits are tested using Oscilloscope and Digital Multi-meter in electrical engineering lab.

All sensors, including motion, brightness, humidity, ultrasonic, temperature sensors, were tested in the electrical lab using Arduino IDE Analog Input and Serial Monitor and Multi-meter in electrical engineering lab.

The Android application was developed in the Eclipse Software Development Kit (SDK). The Android application was programed and tested in Samsung Galaxy S4.

2.4 * Related Documents and Supporting Materials

A **media access control address** (**MAC address**), also called physical address, is a unique identifier assigned to network interfaces for communications on the physical network segment. MAC addresses are used as a network address for most IEEE 802 network technologies, including Ethernet and WiFi.

Bluetooth Low Energy (BLE), or Bluetooth 4.0, works in frequency between 2400 to 2483.5 MHz. BLE contains 40 channels, 2 MHz Channel Bandwidth, and 1 Mbps nominal Data Rate.

2.5 * Definitions and Acronyms

(All team members contribute in the Definitions and Acronyms.)

AWG: American Wire Gauge

BLE: Bluetooth Low Energy, or Bluetooth 4.0

MAC: media access control address

FSM: finite state machine

UUID: Universally Unique Identifier

PCB: Printed Circuit Board

UCR: University of California, Riverside

RSSI: Received Signal Strength Indication

BLE: Bluetooth Low Energy, or Bluetooth 4.0

UART: Universal Asynchronous Receiver/Transmitter

I/O: Input and Output

RFID: Radio-frequency Identification

EEPROM (Electrically Erasable Programmable Read-Only Memory)

I2C: Inter-Integrated Circuit

SSID: Service Set Identifier

SPI: Serial Peripheral Interface

bps: Bits per second

WIFI: A local area wireless computer networking

OS: Operating System

EE: Electrical Engineering

CS: Computer Science

JAVA: Computer programming language

USB: Universal Serial Bus

IDE: Integrated development environment or interactive development environment is a software application that provides services to computer programmers for software development

IEEE: Institute of Electronic and Electrical Engineers

LAN: Local Area Network

PHP: Server-side scripting language, web development language, and purpose programming language

JavaScript: it is high-level, dynamic, un-typed, and interpreted programming language.

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.

SDK: Software development kit

RSSI: Received Signal Strength Indication

TCP/IP: Transmission Control Protocol and Internet Protocol. Computer networking model and set of communication protocols used on the Internet and networks.

WEP: Wired Equivalent Privacy

WPA: Wi-Fi Protected Access

WPA2: Wi-Fi Protected Access II

AC: Alternating Current

DC: Direct Current

C/C++: programing language

3 * Design Considerations

3.1 * Assumptions

(Yancong and Yujie wrote the system Assumptions.)

For proper operation of our system, there are several assumptions have to be made. We have to assume that the power or electricity has to be ON at all time. The Arduino Web Server has to be reliably connected to the WI-FI router at all time. The end product should include a control system and the monitor capabilities, which includes main door lock, garage door, sprinkler, light, and air conditioner.

In order to perform the Automatic Door Lock system, Android devices must be running in Android Operating System (OS) 4.3 or later version with Bluetooth 4.0 compatibility. Android devices should be able to correctly detect Bluetooth RSSI. When a authorized Android device presents within 10 feet of the Bluetooth module, the system should reliably enable the automatic connection and control.

In order to properly control the main door through web server, two microcontrollers has to obtain proper communication within the range of 50 feet. Two controllers have to share common ground at all time to ensure that the newly added circuitry does not affect the original control circuit. In addition, for other automatic control system, all sensor readings should be maintained in the stabilized level without unexpected fluctuation.

Our control system assumes that the sensor reading of the main door lock is always matched its physical state. For instance, if the system senses the main door lock is currently closed, The physical state of the lock should also be closed.

3.2 * Realistic Constraints

(All team members contributed on the Realistic Constraints.)

In the implementation of IHCS, we have encountered several realistic constraints in both hardware and software of our system.

The web server program required large Flash memory and Static Random Access Memory (SRAM) on the microcontroller. This constraint prevented the use of lower cost microcontrollers with small memories, such as the Arduino Uno. In addition, Arduino Mega and Bluno need input voltage range between 7-12 VDC, therefore, the sustainable power supplies are needed. The Arduino Mega microcontroller, which used ATmega2560, is operating on 5 VDC. The Input and Output (I/O) pins of Arduino can handle maximum 40 mA of current, which is the limitation for applications that required higher current.

The Bluetooth 4.0 module restricts the detection on Bluetooth Universally Unique Identifier (UUID) and MAC address. In this case, the Bluetooth module can only be used as slave device when it communicates with Android devices.

The monitoring system requires excessive collection of data from multiple sensors. The accuracy of the sensor readings can be a large constraint for our system. For example, the current sensor's readings was interfered by other current and provide a large fluctuation. As the result, we have to use an additional brightness sensor to double check the light's ON/OFF state.

In order to control the air conditioner, our system adds a parallel control system to the existed thermostat. However, the thermostat has its own integrated circuitry. The system is unable to detect the performance or state of the thermostat. Our control system can only acquire information that shows whether the air conditioner is ON or OFF.

The readings of the humidity sensor are stable and constant when the soil is moisturized; by contrast, this reading has a big changing range when the soil's humidity is dry.

The Ultrasonic distance sensor has about 5% of false readings. As a result, in our Finite State Machine (FSM), the system needs to double check each reading before taking any action or changing any value of the state.

Moreover, since the controller should respond to requests from web client at any time, we should limit the use of delay function in the main program. Instead, we need to implement a Timer function to solve this problem.

3.3 System Environment and External Interfaces

(All team members contributed in this section.)

We programed both our servo system and main door control system by using Arduino IDE under MAC operating system or windows 10 operating system. Qixing Huang programed the Android App by using eclipse under MAC operating system. We chose Wi-Fi as a communication method between controller and human through a webpage interface. We chose RS-232 protocol as a communication method between our main controller Arduino Mega and our main door controller Bluno. We chose Bluetooth 4.0 as a communication method between our main door controller Bluno and our Android Phone Samsung Galaxy 3.

3.4 * Industry Standards

(All team members contributed in this section.)

The IEEE 802.11 standard specifies the wireless connectivity for fixed, portable, and moving stations within a local area. The IEEE 802.11 standard requires that all devices that operate under the term Wi-Fi can communicate to each other. The WIFI shield 101 module we used in our design needed to be compliant with the IEEE 802.11 standard.

UART: A universal asynchronous receiver/transmitter, abbreviated UART is a computer hardware device that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as TIA (formerly EIA) RS-232

RS-232: In telecommunications, RS-232 is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE(data circuit-terminating equipment), such as a modem. The

RS-232 standard is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin out of connectors. The current version of the standard is TIA-232-F Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, issued in 1997. UART is used as a communication method between our two controllers in our system.

SPI: The Serial Peripheral Interface (SPI) bus is asynchronous serial communication interface specification used for short distance communication, primarily in embedded systems. The interface was developed by Motorola and has become a de facto standard. SPI is used for our Wi-Fi shield.

Bluetooth 4.0: The Bluetooth SIG completed the Bluetooth Core Specification version 4.0 (called Bluetooth Smart) and has been adopted as of 30 June 2010. It includes Classic Bluetooth, Bluetooth high speed and Bluetooth low energy protocols. Bluetooth high speed is based on Wi-Fi, and Classic Bluetooth consists of legacy Bluetooth protocols. Bluetooth low energy, previously known as Wibree, is a subset of Bluetooth v4.0 with an entirely new protocol stack for rapid build-up of simple links. As an alternative to the Bluetooth standard protocols that were introduced in Bluetooth v1.0 to v3.0, it is aimed at very low power applications running off a coin cell. Chip designs allow for two types of implementation, dual-mode, single-mode and enhanced past versions. The provisional names Wibree and Bluetooth ULP (Ultra Low Power) were abandoned and the BLE name was used for a while. In late 2011, new logos “Bluetooth Smart Ready” for hosts and “Bluetooth Smart” for sensors were introduced as the general-public face of BLE. Bluetooth 4.0 is used as a communication method between our control system and Android Phone.

Bluetooth 4.0 Specification:
Range: ~ 150 meters on open field
Output Power: ~ 10mW (10dBm)
Max Current: ~ 15mA
Latency: 3 ms
Topology: Star
Connections: > 2 billion
Modulation: GFSK @ 2.4 GHz
Robustness: Adaptive Frequency Hopping, 24 bit CRC
Security: 128bit AES CCM
Sleep current ~ 1µA
Modes: Broadcast, Connection, Event Data Models
Reads, Writes

3.5 * Knowledge and Skills

(All team members contributed in this section.)

Our senior design project required different kinds of knowledge, which contains hardware design, software programming, and electrical power. Some of the knowledge and skills were introduced to our team members in related electrical engineering courses in University of California, Riverside and Huazhong University of Science and Technology. In order to achieve our design objectives, all members must contribute based on their skills and personal interests. Each member has to perform individual research and self-learning to improve his knowledge and skills on related subjects.

The Electrical Engineering courses in UCR provides fundamental knowledge for this project. The related courses are listed below:

EE1A: Engineering Circuit Analysis 1,

EE1B: Engineering Circuit Analysis 2,

EE100A: Electronic circuits,

CS10: Introduction of Computer Science for Science, Math, and Engineering 1

CS12: Introduction Computer Science for Science, Math, and Engineering 2

EE120A: Logic Design

EE120B: Intro to Embedded Systems

EE128: Data Acquisition, Instrumentation, and Process Control

Yancong Deng was responsible for building a Electrical Door Lock Control System, design on WIFI Web Server, the HTTP website design, PCB design of H bridge circuit and relay circuit, communication between Arduinos, installation of sensors and wiring for the entire system.

Basic circuit analysis, EE01A and EE01B, was used to design electrical circuit for both DC motor and relay circuit. Logic Design and Embedded Systems, EE120A and EE120B, provide the fundamental knowledge on the major logic of the program and system flow. Programming methods and skills, which were introduced in CS10, CS12, and EE120B, was heavily used in the Arduino program. Moreover, EE128, Data Acquisition and Process Control, also provided an important technical knowledge in the control system and communication between microcontrollers. It also provides knowledge for Yancong to implement the Electrically Erasable Programmable Read-Only Memory (EEPROM) and Watchdog Timer function to ensure the stability of the program.

In order to add a new control system into the commercial electrical door lock, Yancong has to design an additional control circuit without disabling the existing door lock system. Studying the datasheet and understanding the basic working condition of the door lock is required to add a new control system. Moreover, in order to implement the the WIFI Server on Arduino, Yancong had done individual researches on Web Server, HTTP protocol, fundamental knowledge of basic network. Understanding the HTTP protocol helped Yancong to setup the connection between WIFI 101 module to the Internet. Yancong also designed a simple webpage using Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS).

Qixin Huang was responsible for building Android application for Bluetooth connection, the second version of webpage, design of PCB board, and wiring for the system.

The basic conception of data communication and data acquisition was taught in EE128. The general experimental testing techniques was also introduced in multiple engineering courses. The skills used for power system was taught in Electrical Engineering Practice. The basic knowledge for sensors was taught in the Principle and Design of Sensors.

In order to build Android application, Qixin had to learn Java programming. Coding in Java was entirely new experience for our team members. Qixin had to spent extra time to do individual research for learning programming structure for Android application. Qixin also learned the programming structure for Arduino in Arduino IDE to set up the communication between Android devices and controller. For designing HTTP website, Qixin learned HTML, CSS and simple JavaScript languages. These programming languages for website design were also new for us. The CSS file is stored in google drive and HTML file is run in main controller.

Yujie Cao was responsible for Automation Systems, which include Auto Garage Light System, Auto Door Lock System, Auto Home Light, Auto Garage Door System. Yujie also worked on Webserver, the first version website design, design of PCB board of relay circuit, communication between Arduinos, and implementation of sensors and wiring.

For this project, Yujie had applied fundamental engineering knowledge on Engineering courses. The courses include the Circuit Analysis, Electronic circuits, Electronic circuits, C/C++ programming, Logic Design, Embedded Systems, and Data Acquisition, Instrumentation, and Process Control.

In addition, Yujie has also done research on the following topic: HTTP protocol, router's port forwarding techniques, changing the LAN devices into WAN devices, Website design using HTML and CSS, fuse, PCB design, and Timer function for microcontroller.

3.6 * Budget and Cost Analysis

Part one (main door and electrical lock) Expenses	
Electrical lock	$69.99
Bluno	36.90
Relay	1.00
Electronic components	5.00
IP Camera	71.40
Electrical box	12.99
Total Cost	**$197.28**

Part two (Arduino, devices and sensors) Expenses	
Arduino Mega 2560	$13.19
20 AWG Wire (200 feet)	21.00
14 AWG Wire (100 feet)	14.99
PCB Mount Screw Terminal Block	5.66
Cat5e Ethernet patch cable(100 feet)	17.00
8 Channel DC 5V Relay	6.98
In-Line Fuse with holder	7.00
Electrical box	22.99
Sensors	12.00
Light switches	6.00
Total Cost	**$126.81**

Summary of Project Expenses		
Group Budget		$500.00
Part Expenses		
Part one	197.28	
Part two	126.81	
Total Expense		324.09
Remaining Balance		**$175.91**

3.7 * Safety

(Yancong and Yujie wrote the Safety.)

Since Our project involved high voltage, 110 V AC, there are safety concerns when designing the system. For example, the high voltage or current can cause damage on human body or serious injury. If leakage of electricity happens, the entire system can be damaged. Our system has to prevent any high voltage wiring that is exposed. All wire connection should be fully covered by insulating material. All relays and control circuits should be placed into an electrical box. To protect the system from over-current, fuses are required to add into the circuits. When the current is exceeded the limitation, the fuses can properly cut off the electrical power and isolate the components from the system.

Another safety consideration about the project is when Arduino program malfunctions in unexpected cases. In these cases, the system may not respond properly. To avoid the unexpected performance of the system, the program should include Watchdog Timer function to recover the program if any unexpected situation happens in the program. Since the system uses WIFI shield, the system only work under a WIFI environment. If the WIFI connection is lost, our system should make sure users can still obtain original ways to control all devices. The system should be able to reconnect to the Internet automatically when WIFI is available.

When all the operations of the system fail, we have to make sure all devices will not cause any harm to either users or their properties. The program should cut off all control system to make sure that user can be obtain manual control when them system fails.

Since our system controls large electrical devices in a house, property safety should take into consideration. For instance, when the system closes the garage door, it should be able to notice if there is a person under the door. If there is, the system should stop the operation immediately. Another concern is that when the control system is entirely broken, people inside the house should be able to use physical way to exit the house in emergency situation.

3.8 * Documentation

(Yujie wrote this section.)

Our project has two stages, which are research and project implementation.

In the first stage, we tried to find the proper controller, proper components, proper sensors for our project. During this stage, we did a lot of research online. Whenever we found useful information such as finding a new module that is suitable for our project, we would save and share the link with other partners immediately through email, Facebook, and Wechat.

Initially, before implementing the project, we already designed the objective of our project. We draw most of our diagram by using power point, paint, Pspice, and draw.io depending on the specific needs. We saved all of our diagrams on google drive under a folder named Diagrams.

Every Thursday night, we had a meeting to check what we did since last Friday, to discuss what problems that we met, and to set up the tasks that we want to finish in the next week. In addition, based on research knowledge, we brainstormed to modify the objectives of our project along the way.

In the second stage, we designed our electrical circuits, and finite state machine on computer first. Then, after implementing our electrical components, we took photos for every installed circuits and added name tag for every signal wire within the circuit.

In our program, the first section of our program is used for us to record all occupied analog/digital pins and their corresponding sensors or electrical devices for us to keep tracking and debugging. We stored and updated all versions of our program on google drive whenever we added a new function or a new feature to our program.

4 Experiment Design and Feasibility Study

4.1 Experiment Design

4.1.1 4.1.1 Main door lock:

(Yujie wrote this section.)

Objective:

Main door lock we installed is Kwikset PowerBolt2. Yancong adds a new control circuit to the existing lock control system. In order to test the additional door control circuit, we need to do following test:

Task 1: Check whether the door lock can be open through microcontroller Arduino Mega.

Procedure:

Step 1: make sure the door lock was closed.

Step 2: send open signal through Arduino Mega to open the door lock.

Expected result: the main door lock will be open after signal being sent.

Task 2: Check whether the door lock can be closed through microcontroller Arduino Mega.

Procedure:

Step 1: make sure the door lock was open.

Step 2: send close signal through Arduino Mega to close the door lock.

Expected result: the main door lock will be closed after signal being sent.

Task 3: Check whether we can read the door lock's open state through microcontroller.

Procedure:

Step 1: make sure the door lock was open.

Step 2: Using microcontroller to read the current state of the door lock.

Expected result: the variable "lock state" will show that the door lock is open.

Task 4: Check whether we can read the door lock's closed state through microcontroller.

Procedure:

Step 1: make sure the door lock was closed.

Step 2: Using microcontroller to read the current state of the door lock.

Expected result: the variable "lock state" will show that the door is closed.

Task 5: Check whether the control circuit can close the door lock automatically after detecting the lock has been opened for 3 minutes.

Procedure:

Step 1: make sure the door lock was open.

Step 2: wait for 3 minutes.

Step 3: check the current lock state through microcontroller.

Expected result: the lock state should be closed.

4.1.2 4.1.2 sensors:

(Yujie wrote this section.)

Objective:

In our project, we use motion sensors, ultrasonic sensors, brightness sensors, temperature sensor, current sensors, and humidity sensor. We want to test each sensor's property, functionality, sensitivity, reading scales.

Task 1: Check motion sensor's detection range through microcontroller Arduino Uno

Procedure:

Step 1: put the motion sensor perpendicular on the table.

Step 2: run the sample test code on the Arduino Uno

Step 3: use Arduino Serial Monitor to output the reading value of the motion sensor

Step 4: gradually move backwards and moving hands at the same time to check whether the motion sensor can detect the motion.

Step 5: stop moving backwards when the reading become constantly zero.

Expected result: the distance should be about 5 meters depending on each individual sensor.

Task2: Check the ultrasonic sensor's range through Arduino Uno

Procedure:

Step 1: put the ultrasonic sensor on the table.

Step 2: run the sample test code on the Arduino Uno

Step 3: use Arduino Serial Monitor to output the value of the ultrasonic sensor

Step 4: gradually move backwards to check the output value of the sensor

Step 5: stop when the value has more than 5 percent error than the actual value or stop when the accurate reading range is more than 5 meters.

Expected result: the ultrasonic sensor is able to provide accurate reading within 5 meters.

Task3: Check the brightness sensor's reading through Arduino Uno at different time.

Procedure:

Step 1: Put the ultrasonic sensor on the table.

Step 2: Run the sample test code on the Arduino Uno

Step 3: use Arduino Serial Monitor to output the value of the ultrasonic sensor

Step 4: do the first three steps at day time.

Step 5: do the first three steps at night time.

Step 6: place the brightness sensor near by the lamp, and do the first three steps

Expected result: recording readings at different time for setting up parameters later.

Task4: Check the temperature sensor's reading through Arduino Uno at different time.

Procedure:

Step 1: put the ultrasonic sensor on the table.

Step 2: run the sample test code on the Arduino Uno

Step 3: use Arduino Serial Monitor to output the value of the temperature sensor

Step 4: compare the temperature sensor's value with actual temperature reading from thermometer.

Expected result: the value from temperature sensor is closed to the actual temperature value from the thermometer. The difference between the two values should be less than 0.5 Celsius

Task5: Check the current sensor's reading through Arduino Uno at different time.
Procedure:
Step 1: install the current sensor around the actual lamp cable.
Step 2: run the sample test code on the Arduino Uno
Step 3: use Arduino Serial Monitor to output the value of the current sensor
Step 4: turn on the light and record the actual value.
Expected result: when the light is on, the reading is positive. when the light is off, the reading is zero.

Task6: Check whether or not the current sensor's reading gets disturbed by the cables around the actual measuring cable.
Step 1: put the current into the actual lamp cable.
Step 2: run the sample test code on the Arduino Uno
Step 3: use Arduino Serial Monitor to output the value of the current sensor
Step 4: turn on the light and record the actual value.
Step 5: turn on the light, and the turn on the light whose cable near the cable that we are measuring. Then record the reading.
Expected result: the current sensor's reading should be same all the time no matter nearby cables have current or not.

Task7: Record the humidity sensor's reading through Arduino Uno while putting the humidity sensor at different humidity level environments including: air, table, soil, water.
Procedure:
Step 1: run the sample test code on the Arduino Uno
Step 2: put the humidity sensor in the air, and recording the value.
Step 3: put the humidity sensor in the grass without watering for three days, and recording the value.
Step 4: put the humidity sensor in the grass getting watered everyday, and recording the value.
Expected result: the humidity sensor's reading in the water should be largest, the reading in the air should smallest.

4.1.3 4.1.3 Internet shield selection:

(Yujie wrote this section.)
Objective: We have three internet shield: cc3000 Wi-Fi shield, WIFI101 shield, Ethernet shield. Among the three shields, we want to find the best one for our project depending on the following four parameters: 1. whether the shield is suitable for Arduino Mega or not. 2. the speed of data transmission. 3 the stability of the internet connection. 4 whether we need to use cat5 cable or not.

Task1: Test the cc3000 Wi-Fi shield
Procedure:
Step 1: install the cc3000 Wi-Fi shield on the Arduino Mega
Step 2: run our testing program on it.
Step 3: record the time how long the shield gets connected to internet
Step 4: record the time how long the shield is able to receive the request from the web browser
Step 5: record how often the shield gets unconnected from the internet

Task2: Test the WIFI shield 101
Procedure:
Step 1: install the WIFI shield 101 on the Arduino Mega
Step 2: run our testing program on it.
Step 3: record the time how long the shield gets connected to internet
Step 4: record the time how long the shield is able to receive the request from the web browser
Step 5: record how often the shield gets unconnected from the internet

Task3: Test the Ethernet shield
Procedure:
Step 1: install the Ethernet shield on the Arduino Mega
Step 2: connect the cat5 cable between the Ethernet shield and router
Step 3: run our testing program on it.
Step 4: record the time how long the shield gets connected to internet
Step 5: record the time how long the shield is able to receive the request from the web browser
Step 6: record how often the shield gets unconnected from the internet

Expected result: The Ethernet shield should have the fastest speed of data transmission, and strongest internet connection.

5 * High Level Design

5.1 Conceptual View

(Yancong and Yujie wrote this section, Figure 5.1.1 is created by Yujie and Yancong.)

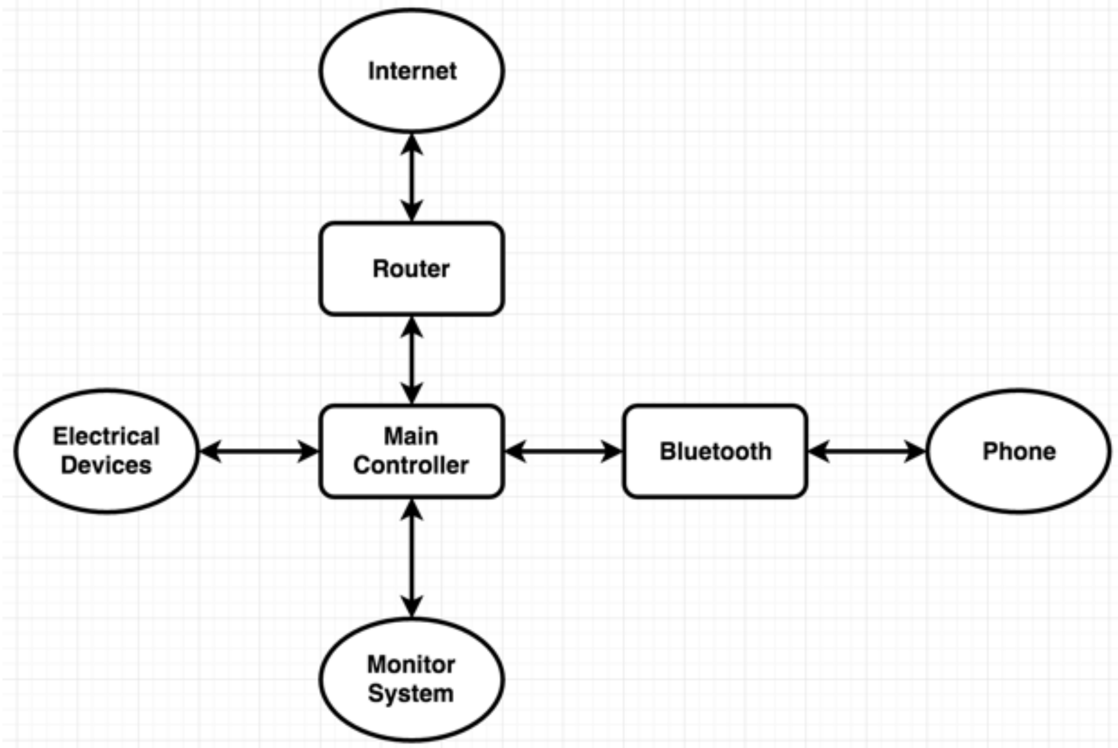


Figure 5.1.1: The diagram above shows the high level design of the Intelligent Home Control System. The Main Controller is connected to the Internet through WIFI router. The internet connected web server is able to receive sensor data and display real-time status online as the Monitor System. With Bluetooth compatibility, controller is able to communicate with phone application. The IHCS allows users to control varieties of devices in the house through both the Internet web server and phone application.

5.2 Hardware

(Yancong wrote this section. Figure 5.2.3 is created by Yancong and Yujie. Figure 5.2.2 is created by Qixin.)

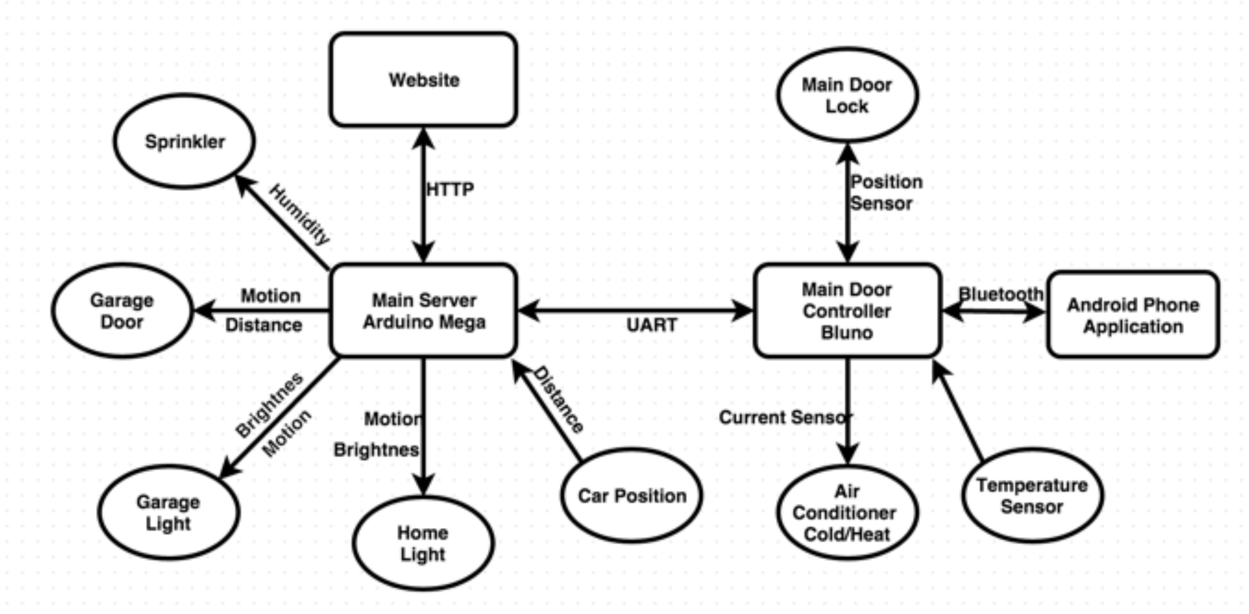


Figure 5.2.3: The diagram above shows the hardware diagram of the overall Intelligent Home Control System. IHCS uses two microcontrollers, the Arduino Mega 2560 and Bluno. Tow controllers communicate through UART. The controllers receive sensor information from different kinds of sensors, which include humidity, motion, ultrasonic, brightness, temperature, and current sensors. The main server controller is connected to the Internet using HTTP protocol. The Main Door Controller, Bluno, communicates with Android devices through Bluetooth protocol. Moreover, Bluno is able to control the main door lock automatically through Android application while the main controller controls all devices in the house through the Internet. IHCS is capable of controlling sprinkler, main electrical door, garage door, garage light, Home light, and air conditioner.

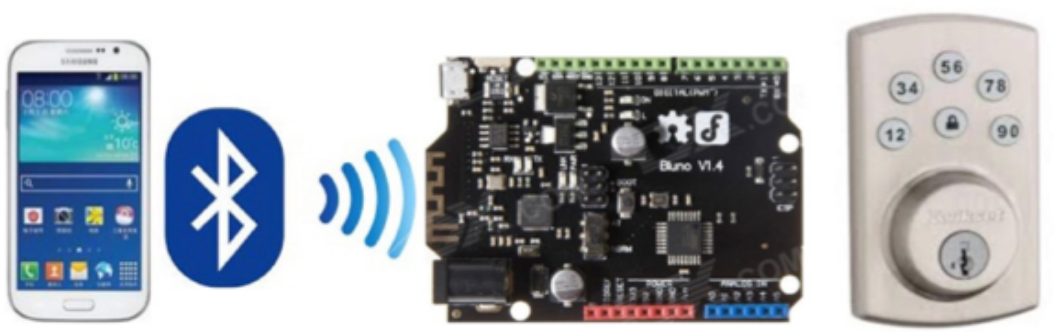


Figure 5.2.2: The diagram above shows the high level hardware diagram of Bluetooth Door Lock

Control System in IHCS. Android devices communicates with the Bluetooth module on Bluno to control the electrical door lock.

5.3 Software

(Yancong wrote this section. Figure 5.3.1 is created by Yancong.)

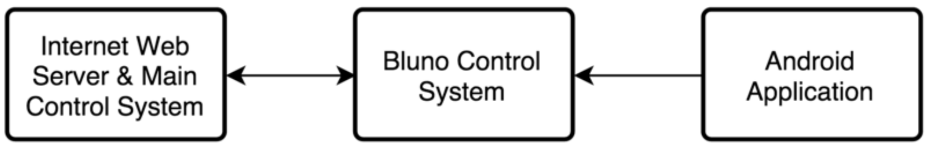


Figure 5.3.1: The diagram shows the high level software diagram of Intelligent Home Control System.

Our system contains three major programs, the Internet Web Server and Main Control System, Bluno Door Control System, and the Android Application. Both programs on Web Server and Bluno used C++/C in Arduino IDE while the Android Application used JAVA programming.

6 * Low Level Design

6.1 WIFI Web Server

(Yancong Deng is responsible for the WIFI Web Server.)

The WIFI Web Server is one of the major components in the Intelligent Home Control System. It is held on the Arduino Mega 2560 microcontroller. The Web Server is connected to the Internet with the Arduino WIFI Shield 101 module. The server is able to detect the status of the WIFI shield and the current connection with the Internet. When the module is connected to the Internet, Arduino is able to acquire data and requests when a client tries to access the web server via web browser. By analyzing the request from the client, the web server is able to output the HTTP website and execute the control command from the client.

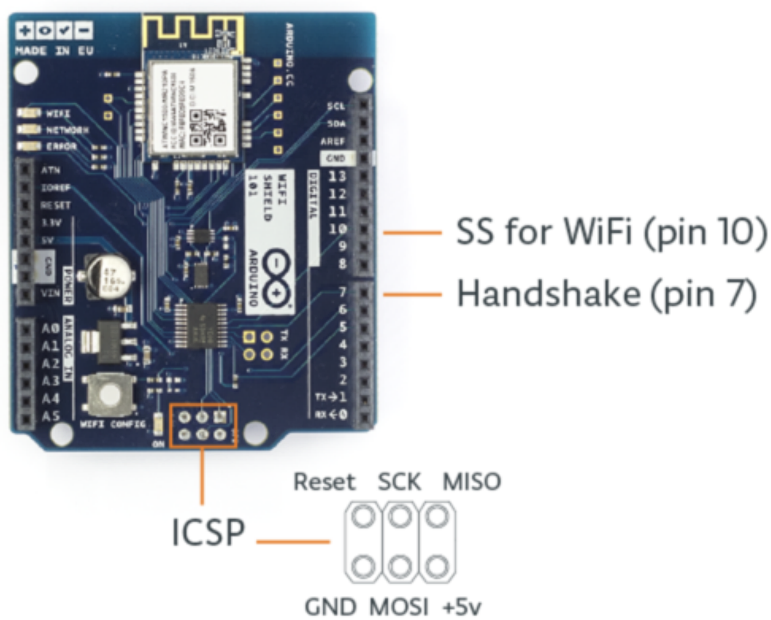


Figure 6.1 The above diagram shows the WIFI Shield 101 Module.

6.1.1 Processing narrative for WIFI Web Server

The WIFI Shield 101 Module wirelessly connects to the Internet using Service Set Identifier (SSID) and its password. The WIFI Web Server program can be divided into three major parts, which are initialization, recovery, and action.

In the first part of the program, it initializes the SSID, password, and port of the web server, which is the main port to hold the web server. In the initialization phase, the program has to ensure the present of the WIFI shield and the proper connection between the microcontroller and the Internet. After connected to the Internet, the program enters the main loop of the system, which contains the second part and the third part of the program.

The second part of the program is the recovery system. In this phase, the system constantly checks the WIFI connection in the main loop. This can prevent the system crashes when the Internet connection is lost. When the lost of WIFI signal is detected, the program enters into a loop for reconnection. The programs constantly check for the availability of the WIFI signal and attempt to reconnect. The program gets out of the loop when Stable WIFI connection is established.

The third part of the program is the main action. It constantly checks the server status for the availability of a client from web browser. When the system detected a request from a client, the program then tests if the client is connected and the availability of data. The system reads and records the request from the client until the HTTP webpage request, which is the Newline character '\n', is received,. Moreover, the system compares the received data with the predefined control command. If the the recorded request matches the control command, the system can execute the corresponding command.

WIFI Web Server Software Diagram

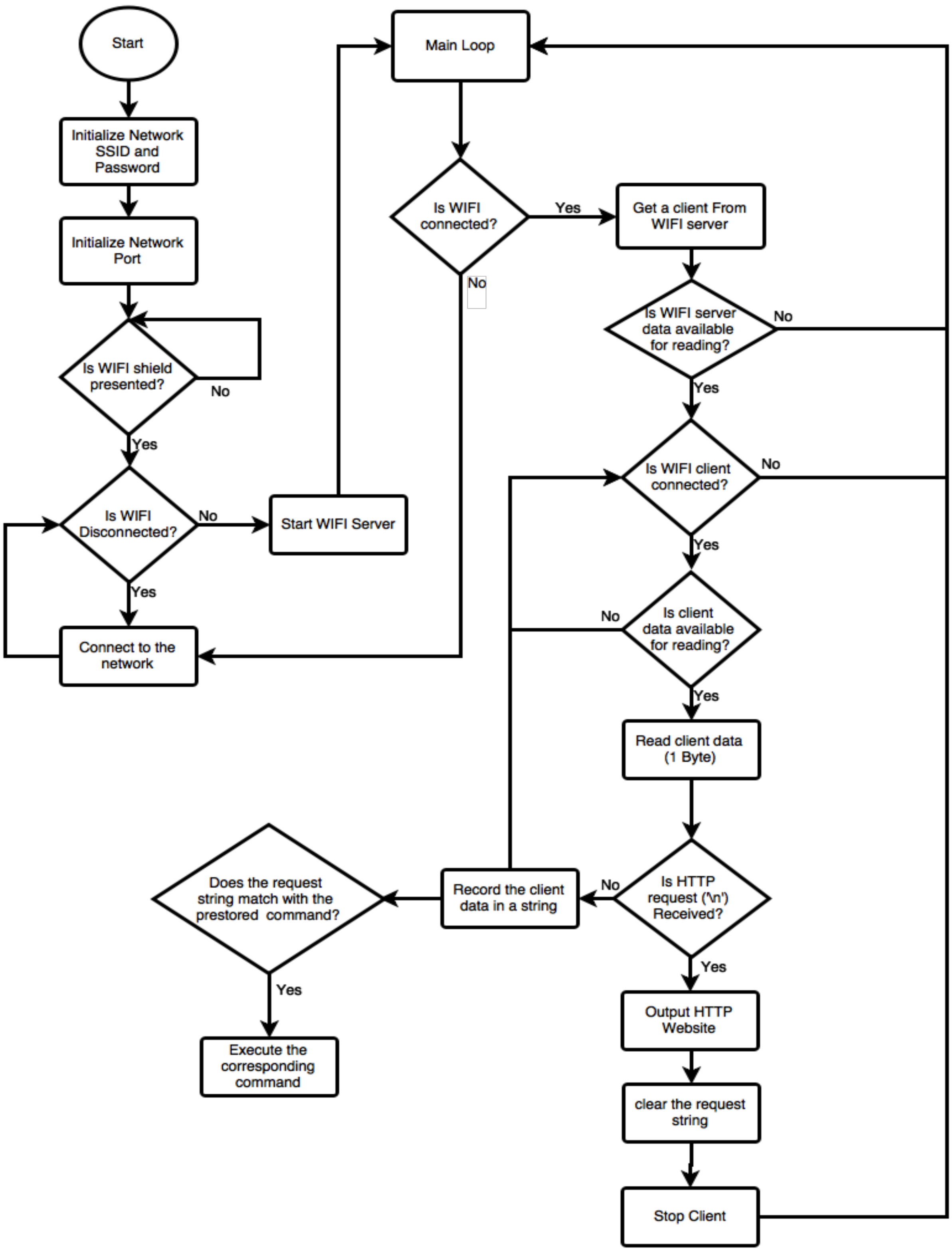


Figure 6.1.1 The above software diagram shows the basic structure of the Internet connected Web Server.

6.1.2 WIFI Web Server interface description.

The WIFI Web Server is the major control component for our system, which is able to display and control all the electrical devices through HTTP webpage. The Arduino Mega used HTTP protocol to deliver the real-time sensor data and the main control panel on the Internet website. The Internet website is created using HTML, CSS, and simple JavaScript programming. When the HTTP website is open on a web browser, the website will update its status every 20 seconds.

At the same time, the web server is able to receive requests from the users. Users can simply press the buttons on the HTTP website, and the Arduino Mega controller is able to read the corresponding request with the server program. Based on the command, the controller will take action accordingly.

The main controller also uses UART to communicate with the slave controller, Bluno. The system is able to monitor and control the main door and air conditioner.

6.1.3 WIFI Web Server processing details.

The WIFI Shield 101 is a IoT shield with crypto-authentication that can wirelessly connect Arduino with the Internet. The WIFI connection is used IEEE 802.11b/g/n standard, and it has up to 72 Mbps networks. The WIFI Web Server is able to connect to the Internet through WIFI router using the SSID and password of the router.

The web server website can easily design using HTML, CSS, and JavaScript. However, the Arduino web server has some design constraints and limitations. The web server requires Arduino Mega print out the HTML page to the client browser. Since the HTML program can be very large, the SRAM of Arduino Mega is limited to 8 K bytes, therefore, the memory limitation constraints the design of the website.

Moreover, the web server can only processing one client request at a time. If a client tries to continuously press the button and send request, the web server will sometime stop responding. For this case, we have added a Watchdog Timer functionto ensure the system can be properly restart.

6.2 Electrical Door Lock

(Yancong Deng is responsible for the Electrical Door Lock.)

One of the major components of our system is the Electrical Door Lock System. For the Electrical Door Lock, we have chosen the Kwikset Satin Nickel Powerbolt 2 Touchpad Electronic Deadbolt with Smart-Key. This Electrical Door Lock contains a password Touchpad and electrical motor circuitry.

Figure 6.2.1 The above picture shows the commercialized product of electrical door lock, the Kwikset Satin Nickel Powerbolt 2 Touchpad Electronic Deadbolt with SmartKey.

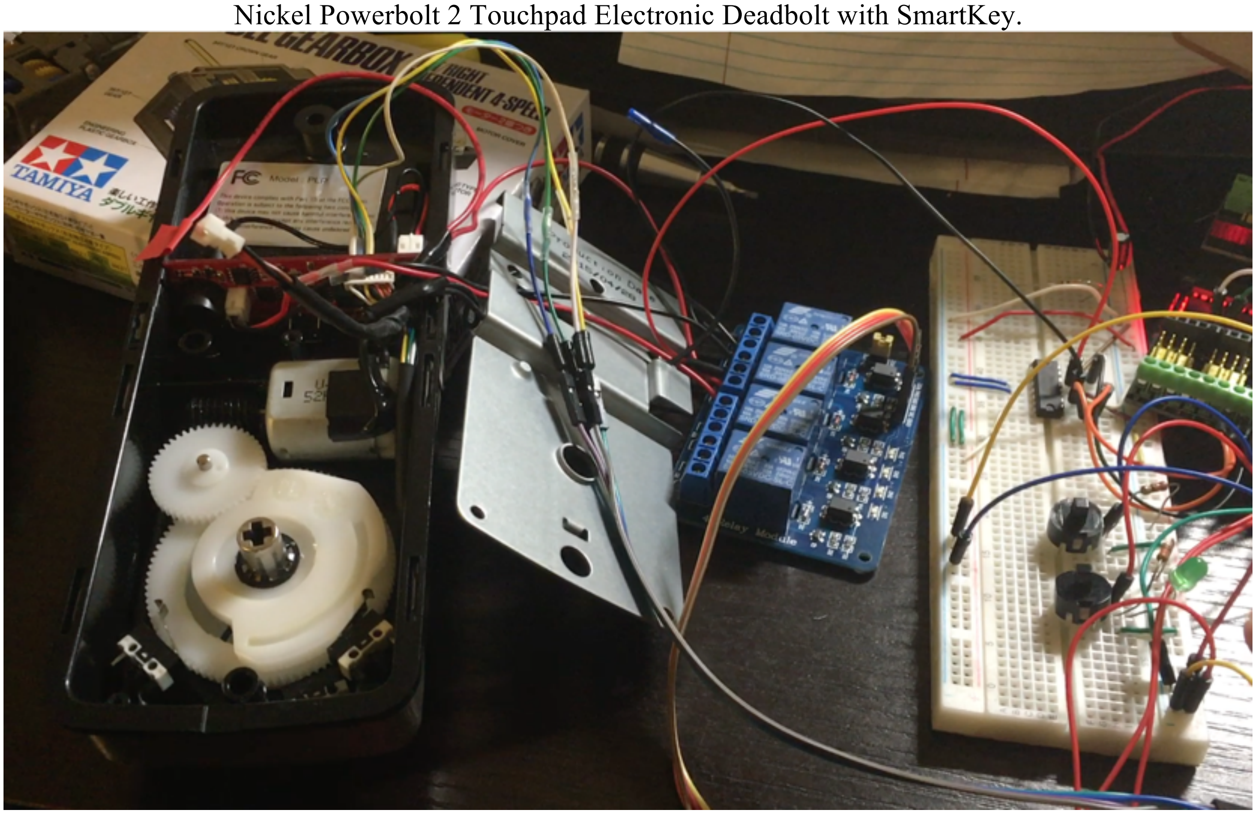


Figure 6.2.2 The above diagram shows the actual implementation of the door lock control circuit.

In order to control this electrical door lock for our system, we have to add a new control method into the existing control circuit. More importantly, our system should not disable the original control system. This makes sure the user can still use the door lock when the system stops working. For our system, we designed the bypass control method to introduce a new control system into the lock. The system uses H-bridge driver to control the DC motor of the electrical door lock. Our system also uses two electrical relays to separate the two control circuitries, which ensure only one control pathway is enable at a time. Our system is able to acquire the door lock position by reading the internal position sensors of the electrical door lock. We are able to test and understand the position sensor reading.

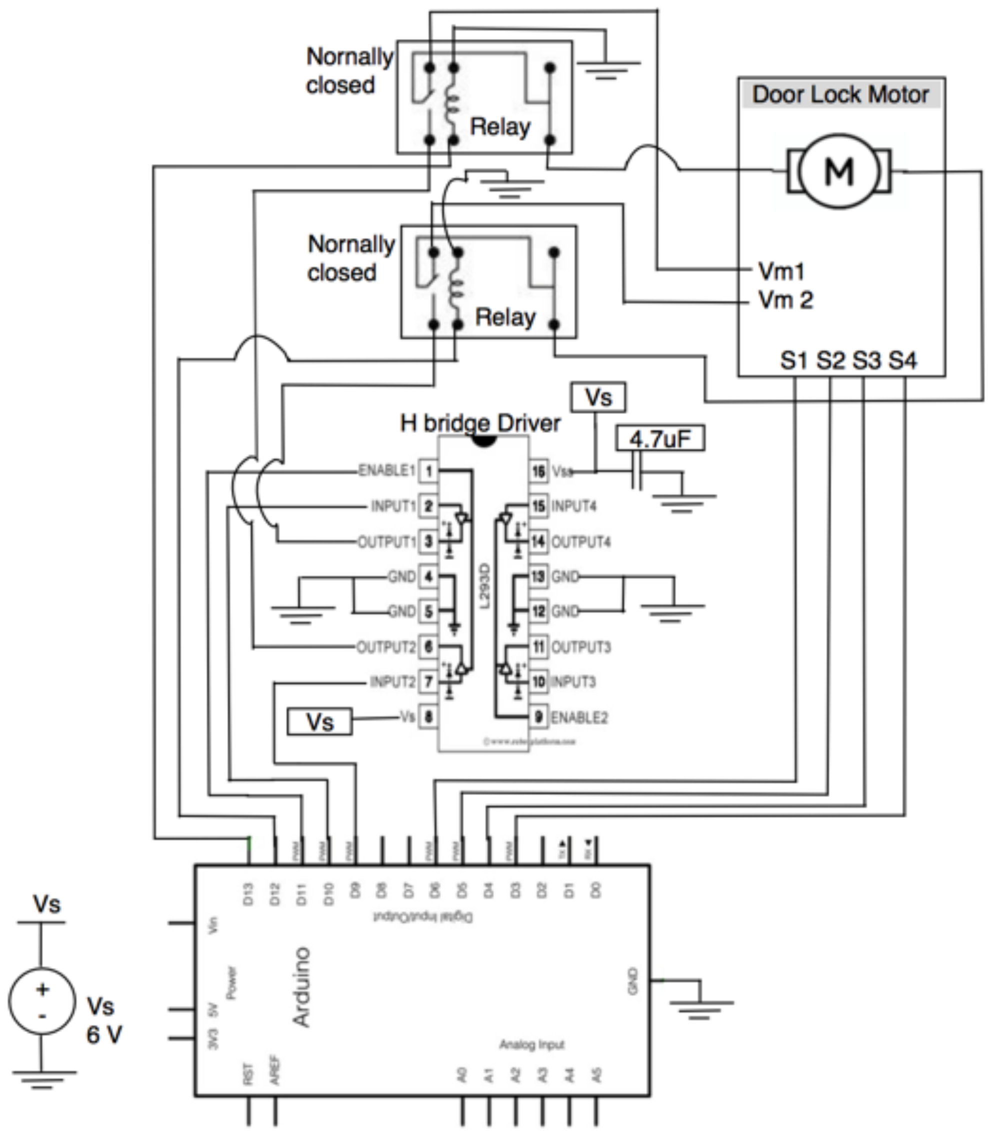
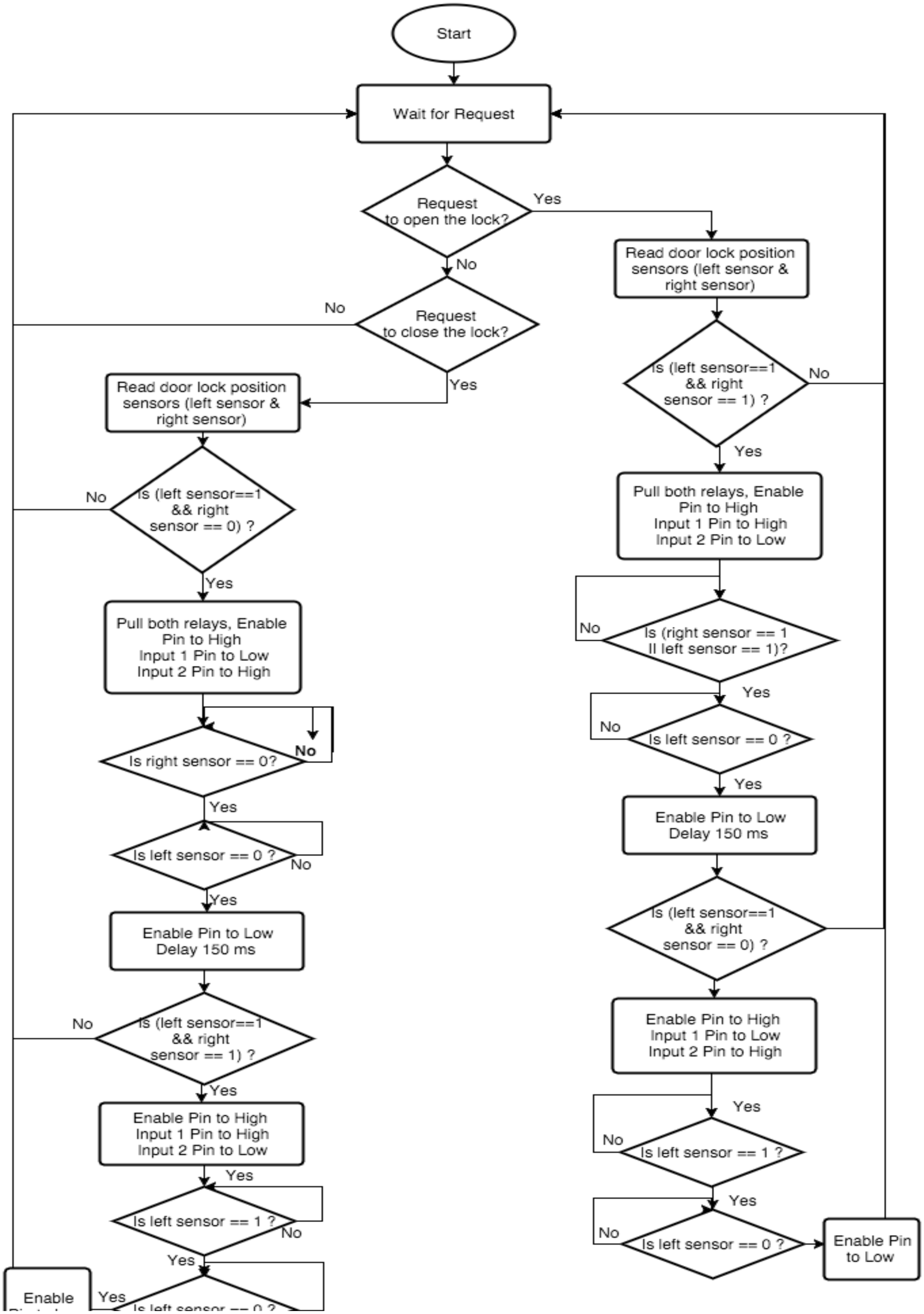


Figure 6.2.3 The above diagram shows the schematic of the door lock control circuit with H bridge driver and electrical relays.

6.2.1 Processing Analysis for Electrical Door Lock

As Figure 6.2.3, the controller is able to acquire door lock position by reading the S1 – S2 Pin on the Door Lock Motor, which are the door lock position sensors. By reading and analyzing the position sensors, Arduino program is able to determine whether the lock is open, close or neither.

More importantly, our system is able to control the Door Lock Motor using the H-bridge Driver and electrical relays. When the Arduino attempts to control the lock, the program has to apply a 3.3 to 5 V to both relay modules to cut off the connection of the original control circuit of the door lock. Moreover, in order to use the H-bridge driver, we have to pull the Enable Pin, Input 1 Pin, and Input 2 Pin of the H-bridge driver to HIGH. The main control software program is provided as the following diagram:



Start
Wait for Request
Request to open the lock?
Yes
No
Request to close the lock?
No
Yes
Read door lock position sensors (left sensor & right sensor)
Is (left sensor==1 && right sensor == 1) ?
No
Yes
Pull both relays, Enable Pin to High Input 1 Pin to High Input 2 Pin to Low
No
Is (right sensor == 1 ll left sensor == 1)?
Yes
No
Is left sensor == 0 ?
Yes
Enable Pin to Low Delay 150 ms
Is (left sensor==1 && right sensor == 0) ?
Enable Pin to High Input 1 Pin to Low Input 2 Pin to High
Yes
No
Is left sensor == 1 ?
Yes
No
Is left sensor == 0 ?
Enable Pin to Low
Read door lock position sensors (left sensor & right sensor)
No
Is (left sensor==1 && right sensor == 0) ?
Yes
Pull both relays, Enable Pin to High Input 1 Pin to Low Input 2 Pin to High
Is right sensor == 0?
No
Yes
Is left sensor == 0 ?
No
Yes
Enable Pin to Low Delay 150 ms
No
Is (left sensor==1 && right sensor == 1) ?
Yes
Enable Pin to High Input 1 Pin to High Input 2 Pin to Low
Yes
Is left sensor == 1 ?
No
Yes
Enable
Yes

Figure 6.2.3 The above Figure shows the software diagram of the Electrical Door Lock control system.

6.2.2 Electrical Door Lock interface description.

The main controller constantly acquires data from the door lock position sensors. Based on the sensor readings, the controller is able to update the state information of the door lock online through HTTP protocol. When the system receive a request from the webserver or Android application, Based on the state of the door lock, the Electrical Door Lock system is able to determine if any action is required due to the client requests.

6.2.3 Electrical Door Lock processing details

The PCB design control circuit for the door control system, the major components are one H-bridge and two normal relay. For the Door Lock Control system, we used the Dual H-Bridge Motor Driver from Texas Instruments, L293DNE chip.

In the circuit design, the system used electrical relay to separate two control systems, the Arduino control system and the original system. The original system was connected to the Normally Closed Pin on the electrical relay. In this case, users can still use normal operation of the door lock when the IHCS is not activate. When IHCS attempts to control the lock, it can simply provide a HIGH voltage to the relays and turn ON the Arduino control system.

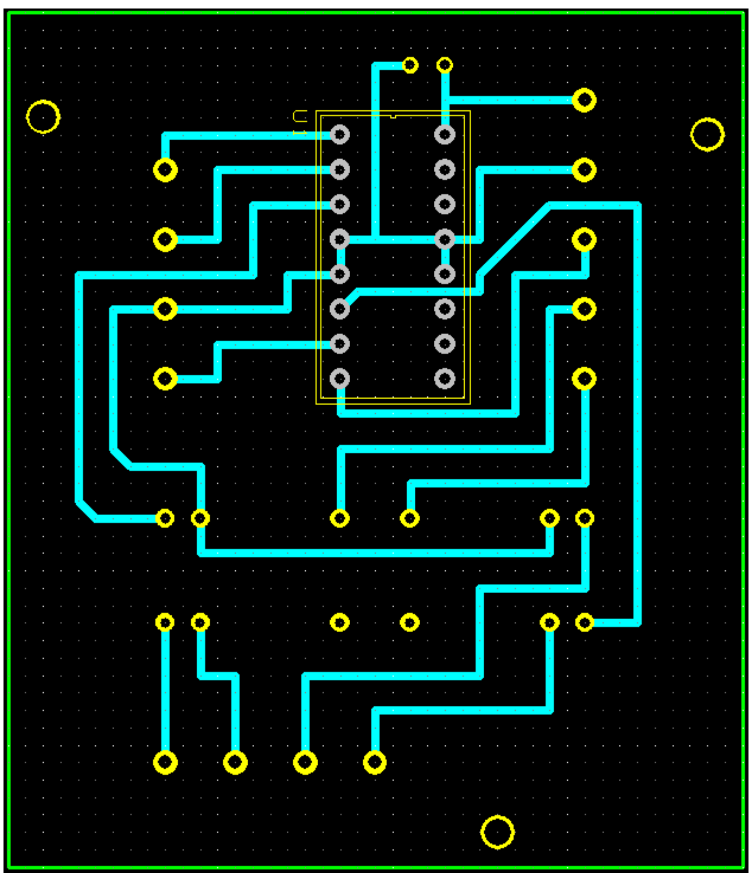


Figure 6.2.4 The above diagram shows the PCB design of the H-bridge circuit and relay circuit.

6.3 Bluetooth Android Application

(Qixin is responsible for this section.)
The Android application is a part of Door Lock Control System. In order to open the main door lock automatically, It requires Bluetooth communication and Android application, so that Android devices can search, connect, and communicate with the controller, Bluno, automatically.

Bluetooth Android Application
Software Diagram

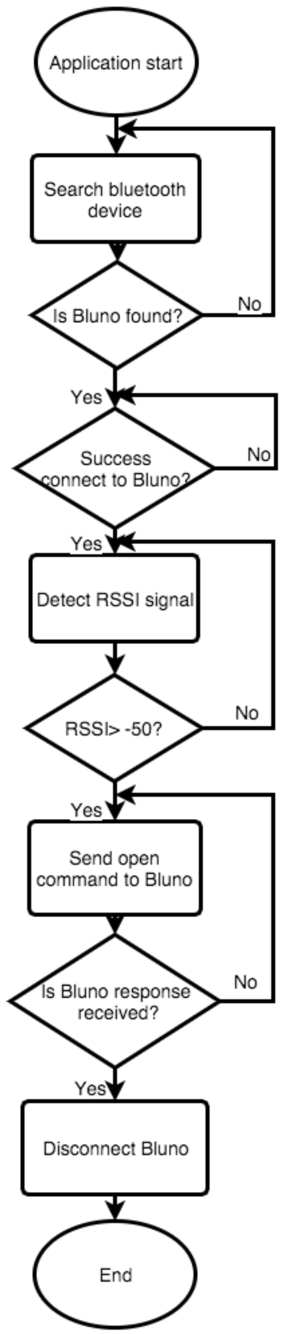


Figure 6.3.1 The above diagram shows the Bluetooth Application Software Diagram

6.3.1 Processing narrative for Bluetooth Android application

The Android application does not require to press any button to accomplish the Automatic Control function. It is able to detect any Bluetooth devices within a rang of 50 feet. By analyzing the RSSI of the Bluetooth signal, the program is able to estimate the distance of the devices. When the Android application detected the distance to Bluno is within 3 feet, it will attempt to connect to the controller. Once the connection is established, the Android application sends out the passcode for opening the door. When Bluno successfully received the passcode, it will open the door automatically. The entire process from detected the devices within range to send out passcode needs approximately 5 to 7 seconds.

6.3.2 Bluetooth Android Application interface description.

The Android application does not require any action for opening the main door lock as long as the application is opened and running on the Android devices. When the application is opened, it enters into an interface, which shows the detection, connection, and sent message of the device.

First of all, the application begins with searching the surrounding devices until it finds the specified Bluno. Then, the application keeps checking the Received Signal Strength Indication (RSSI) signal until it goes up to -50. When RSSI reached the desired value, the program will attempt to establish a connection. After the Bluetooth connection is established, the application sends a passcode to the Bluno for requesting to open the door. When the Android application receives a response from Bluno, said the door is already open, the application turns off the Bluetooth function.

6.3.3 Bluetooth Android Application processing details

The Android application is designed for Bluetooth 4.0 or higher version. It was written in Java language and installed into SAMSUNG Galaxy S4 mobile phone, which should have Android OS version 4.4.2 or higher. The application program can be divided into four main parts, which are searching devices, connection, sending code, and detecting RSSI signal.

In the first part of the application, it uses 'onLeScan' method to search the Bluetooth devices and uses the Bluetooth Adapter to receive the callback. In our application, the program checks if the detected Bluetooth UUID can be matched with the Bluno. If the UUID is successfully matched, the device will be put into the Bluetooth Adapter.

In the second part of the application, the program sets a 'GATT Server' to the Bluno. It uses connectGatt() function to establish a connection between Android device and Bluno. In order to disconnect the connection, the program calls disconnect() function to cut off the connection.

In the third part, after established the Bluetooth connection, the GATT Server has been set up. The application uses the mBluetoothLeService.writeCharacteristic() method to send the information to the Bluno.

In the last part, which is the most important part of the program. The application gets the RSSI value from connected device. The application uses mBluetoothGatt.readRemoteRssi() method to obtain the value in Bluetooth server. Then, the program puts the value into broadcast and allows the program to get the value through receiving broadcast.

6.4 Car Position Signal System

(Yujie is responsible for this section.)

Car Position Signal System has two major functions. The first function is able to detect whether the car parks properly inside the garage or not. The second function is able to notify the driver by blinking the garage light when the car arrived within the proper range. For proper range, it means that both the garage door and back door are able to open and close properly without hitting the car.

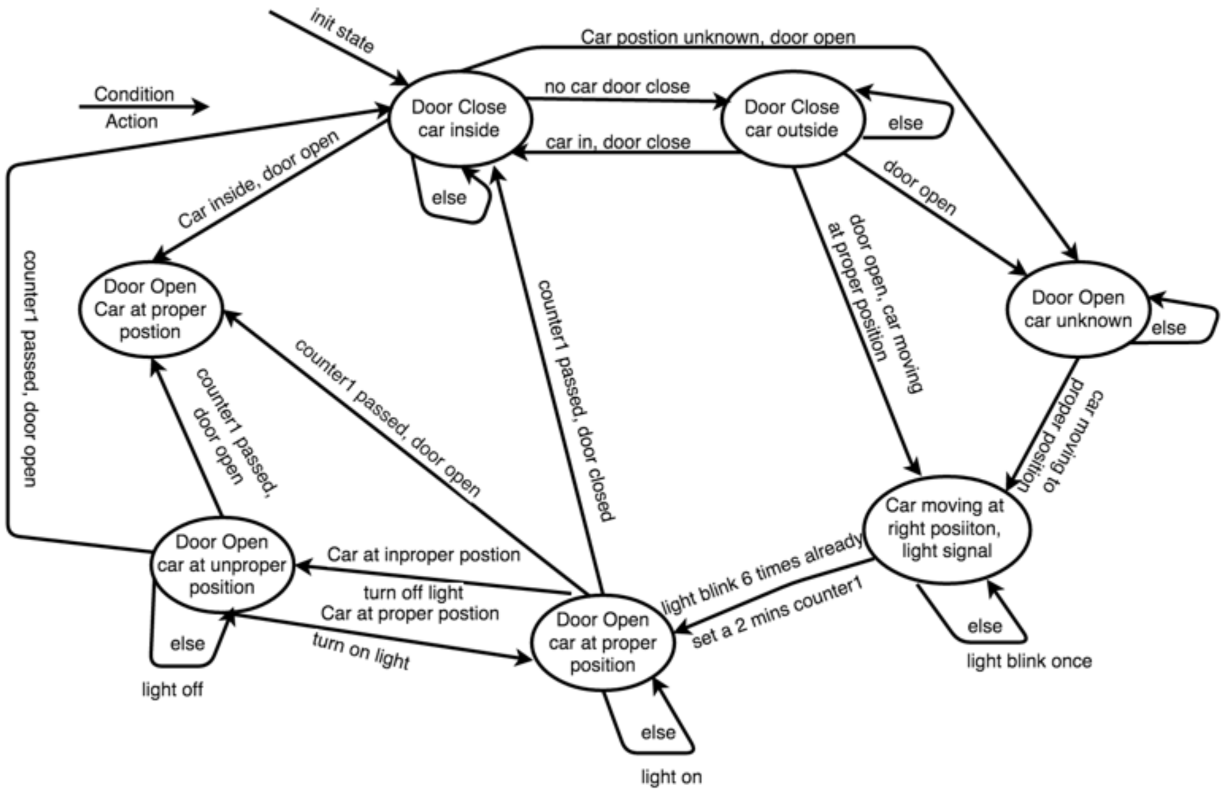


Figure 6.4.1: Car Position Signal System FSM

6.4.1 Processing narrative for Car Position Signal System

In the FSM of the Car Position Signal System, it requires two variables. The first variable is the garage door state, and the second variable is the Ultrasonic distance sensor reading. The distance sensor reading shows the distance between the car and the wall of the garage. If the garage is empty and the garage door is closed, the reading of the Ultrasonic distance sensor should be around 646 cm. The proper parking range for the car is between 80 cm to 115 cm.

When the car parks within the proper range, the Car Position Signal System will notify the user by blinking the garage light three times. If the car stays at the proper range, the light will stay on for 2 minutes. However, after the first notification, if the car moves and exceeds the proper range of the parking position, the Car Position Signal System will again notify the driver by turning off the light immediately. After the second notification, if the driver parks the car back to the proper position, then the Car Position Signal System will turn back on the light for 2 minutes.

6.4.2 Car Position Signal System interface description.

The main controller Arduino Mega constantly updates the state of the garage door and the car position by reading from two Ultrasonic sensors. The main controller also updates the state of the garage light by reading from brightness sensor and current sensor. In our system, all the sensor readings are received by main controller. All the sensor readings will be updated to the webpage by using HTTP protocol when a web client request is received. Meanwhile, the Car Position Signal System automatically runs as long as the main controller is ON.

6.4.3 Car Position Signal System processing details

The sensors used in the Car Position Signal system are the Ultrasonic Sensor and Photo Sensor. The test results and specification of these sensors are listed below.
Ultrasonic Sensor(HC-SR04):

Working Voltage	DC 5 V
Working Current	15mA
Max Range	3m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Photo Sensor:

Working Voltage	DC 3.3V-5 V
Measuring Angle	15 degree
DO:	digital output
AO:	analog output
Dimension	36*15*6mm
Darkness	As the room is darker, the photo sensor reading goes higher.

The limitation of these two sensors are that the readings are sometime fluctuated. For example, we reads the sensor reading about 20 times per second. Within the 20 times of readings, there will be at least one false reading. As a result, in our program, we have to write additional program to ensure the correct sensor reading to prevent unexpected behavior of the system.

6.5 Auto Garage Light System

(Yujie is responsible for this section.)
Auto Garage Light System is able to turn off the garage light when it detects there is nobody in the garage for certain amount of time. If the system does not detect any motion in the garage consecutively for 3 minutes, the light will be turned off. If the system detects motion in the garage, the light will stay on for another 3 minutes and check status again.

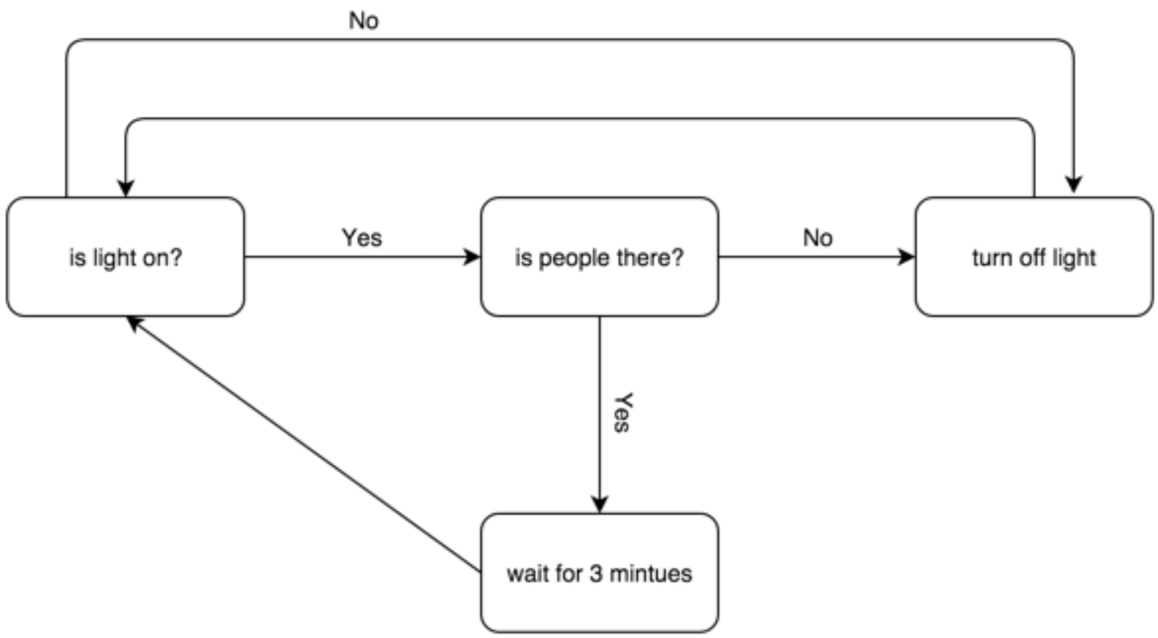


Figure 7.5.1 The above diagram shows the Finite State Machine of the Auto Garage Light System.

6.5.1 Processing narrative for Auto Garage Light System

In this Auto Garage Light System, it requires two sensor readings. The first sensor is the brightness sensor, which detects whether the light is on or off. The second sensor is the motion sensor, which detects whether there is people in the garage door.

The Auto Garage Light System first checks whether the light is ON or OFF. If the light is OFF, then the program does nothing. If the garage light is ON, then the program checks for motion in the garage. If there is motion detected and the light is ON, the garage light will be turned OFF. If motion is detected in the garage, the controller keeps the light to stay ON and checks for the garage's status three minutes after.

6.5.2 Auto Garage Light System Interface Description.

The main controller Arduino Mega constantly updates the state of the garage light by reading from the brightness sensor and current sensor. When the Auto Garage Light System is activated when the garage light is ON.

6.5.3 Auto Garage Light System Processing Details

The sensors used in the Auto Car Garage Light system are the Motion Sensor and Photo Sensor. The test results and specification of the photo sensor is Page. 39. The test results and specification of the Motion sensor is listed below.

Motion Sensor:

Working Voltage range	DC 4.5-20V
Quiescent Current	50uA
Max Range	7 m
Min Range	1 cm
Measuring Angle	110 degree
Signal Output - High	3.3V
Signal Output - Low	0V
Dimension	32 * 24 mm
trigger	repeated(default)
Time delay	0.1sec(default)

There are limitation for the motion sensor used in our system. If there is a person sits stilly in the garage door. The motion sensor sometimes can not detect this person, so that the system will think there is nobody in the garage and turn OFF the light after three minutes. Other than this problem, the motion detector works accurately for detecting any human motion.

6.6 Auto Home Light System

(Yujie is responsible for this section.)
Auto Home Light System is able to detect any motion in the kitchen and turn ON the Home Light when it is dark. The controller constantly checks the status of the Home Light and the motion in the kitchen. When the Home Light is ON, the controller will keep the light ON if there is people in the kitchen. When the Home Light is OFF, the controller will turn OFF the light if no motion is detected for three minutes. Moreover, the Auto Home Light System is able to open the light when the Kitchen is dark. When the Home Light is OFF and the environment of the kitchen is dark, the system turns ON the Home Light automatically if motion is detected.

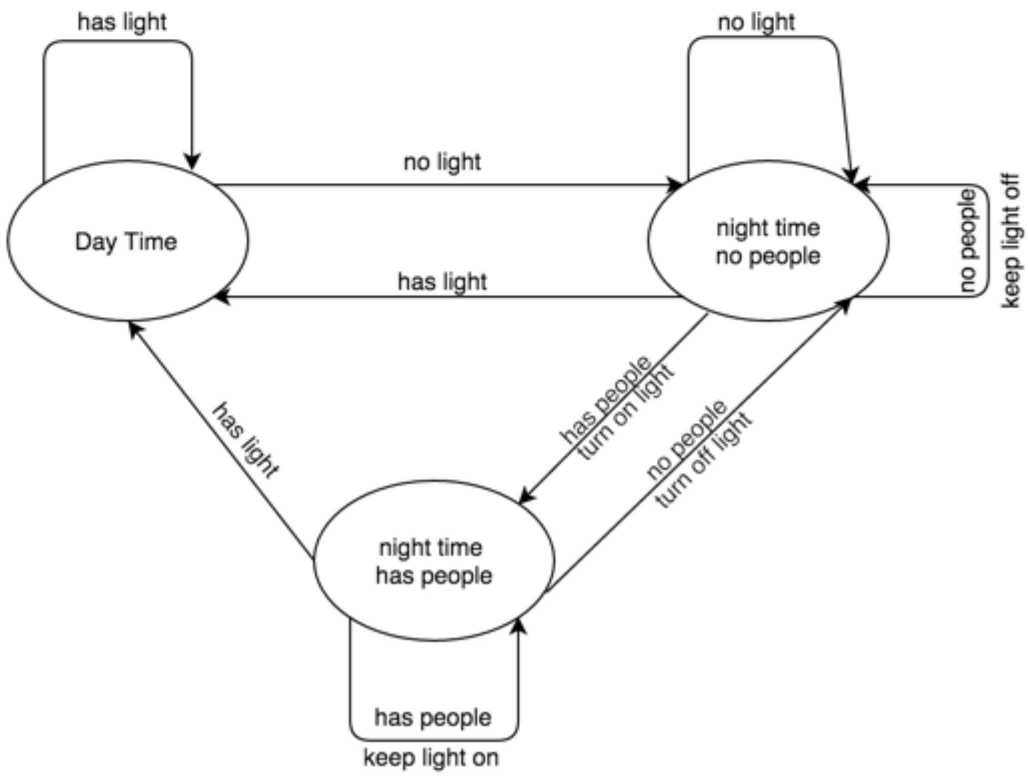


Figure 6.6.1 State Machine of the Auto Home Light System.

6.6.1 Processing narrative for Auto Home Light System

In the Auto Home Light System, it requires two sensor reading. The first sensor is the brightness sensor, which detects whether the light is on or off. The second sensor is the motion sensor, which detects whether there is people in the kitchen.

The Auto Home Light System first checks whether the light is ON or OFF. If the Home Light is OFF, then the program does nothing. If the Home Light is ON, then the program checks for motion in the kitchen. If there is motion detected and the Home Light is ON, the light will be turned OFF. If motion is detected in the kitchen, the controller keeps the light to stay ON.

6.6.2 **Auto Home Light System interface description.**

The main controller Arduino Mega constantly updates the state of the Home Light by reading from the brightness sensor and current sensor. When the Auto Home Light System is activated when the Home Light is ON. With the combination of the brightness and motion sensor, the Home Light will automatically take actions accordingly.

6.6.3 Auto Home Light System processing details

The sensors used in the Auto Car Garage Light system are the Motion Sensor and Photo Sensor. The test results and specification of the photo sensor is listed in Page. 39. The test results and specification of the Motion sensor is listed in Page. 42.

There are limitation for the motion sensor used in our system. If there is a person sits stilly in the kitchen area. The motion sensor sometimes can not detect this person, so that the system will think there is no motion and turn OFF the light after three minutes.

6.7 Auto Main Door Lock System

(Yujie is responsible for this section.)
Auto Main Door Lock System detect whether door is locked or not. This system will be activated only when the main door is unlocked. If the main door is unlocked, the Auto main door will automatically lock the main door after 5 minutes.

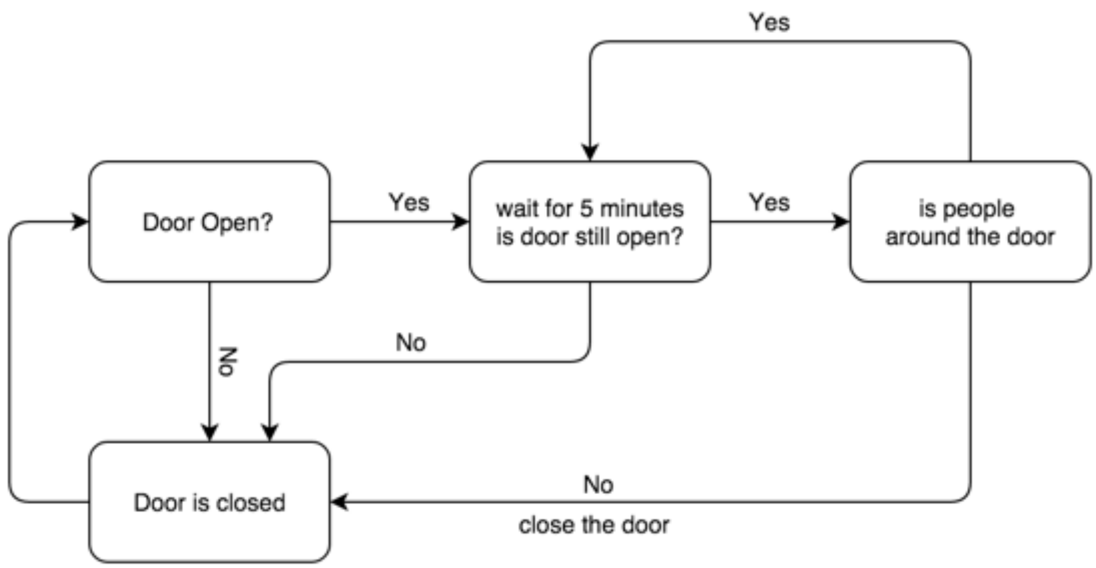


Figure. 6.7.1 The above Figure shows the software diagram of the Auto Main Door Lock Sysrem.

6.7.1 Processing narrative for Auto Main Door Lock System

Auto Main Door Lock System acquires the position status of main door by reading the position sensors. Based on the sensor reading, the system is able to know whether the main door lock is opened, closed, or neither. If the door is open, the Auto Main Door Lock System will check the state of the lock again after 5 minutes. If the lock is still opened, the system closes the lock automatically. If the door is closed after five minutes, the system does nothing.

6.7.2 Auto Main Door Lock System interface description.

The Bluno constantly checks the state of the main door lock and sends the status to the main controller, Arduino Mega, using UART communication. Based on the data provided from the Bluno, the system is able to send command to the Bluno by using UART.

6.7.3 Auto Main Door Lock System processing details

The limitation of UART communication is the speed of the data transmission is very slow.
It takes about 3 seconds for Bluno to correctly receive the command from the Arduino Mega and take action accordingly.

6.8 **Air Conditioner Control System**

(Yujie is responsible for this section.)
Air Conditioner Control system allows user to control their air conditioner through the Internet website. Due to the performance of the Air Conditioner, the system prevents users to turn on cooling and heating at the same time. In order to protect the Air Conditioner, the system also prevent the user from turn on and off the AC alternatively in a short period.

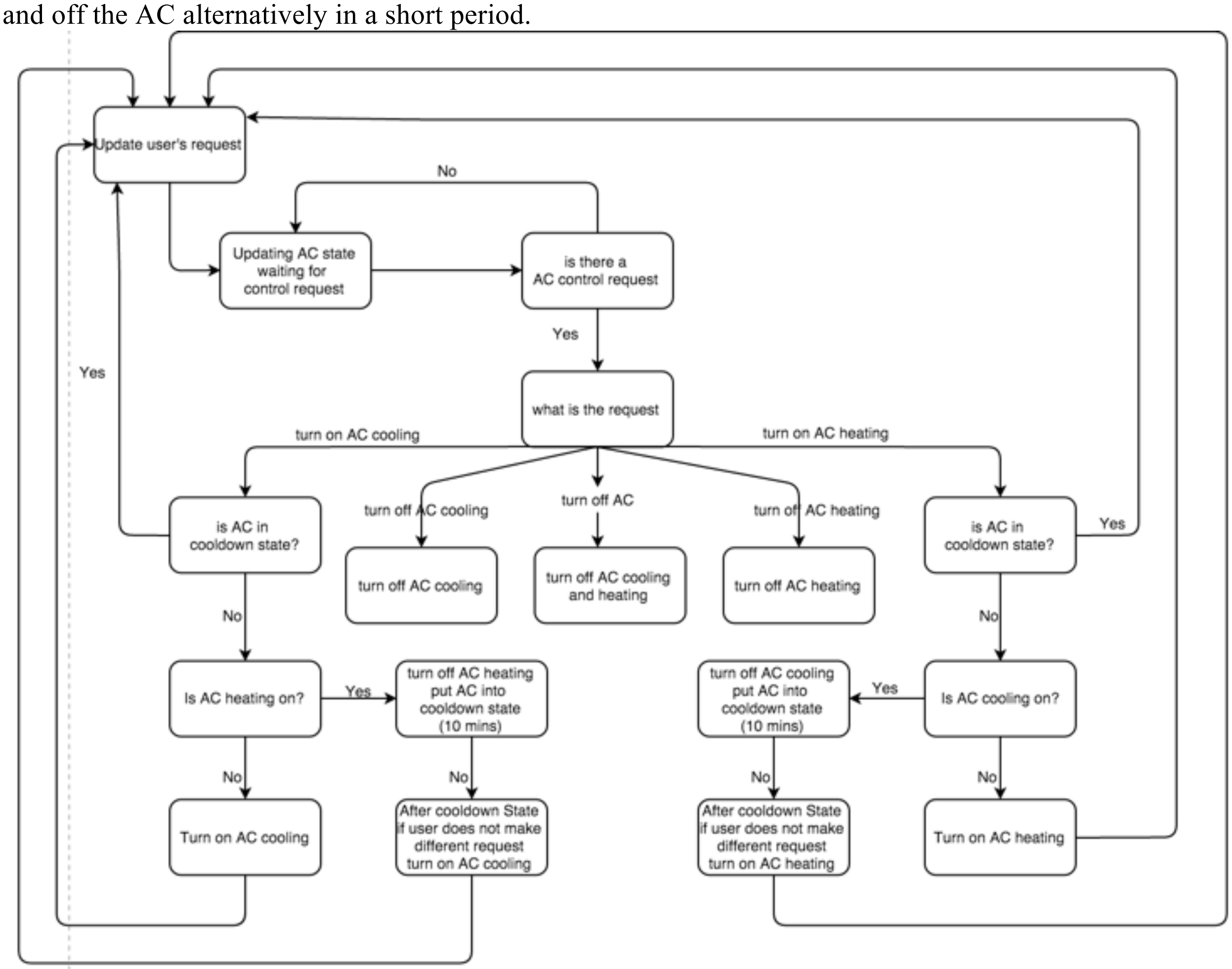


Figure 6.8.1 The above figure shows the software diagram of the Air Conditioner Control System.

6.8.1 Processing narrative for Air Conditioner Control System

The AC control system checks the state of the air conditioner when the web server received a client request. Based on the AC's state and user's request, the system determine the appropriate actions. For example, if a user request to turn on cooling while the heating is ON, the system will turn OFF the cooling and initiate a 10 minutes cool down before turn ON the heating. The purpose of the Cool Down state is to protect the air conditioner from over using the HVAC motor.

6.8.2 Air Conditioner Control System interface description.

The user sends his or her request from webpage to the main controller. After a request is received, the main controller communicates with the Bluno to obtain the AC's current state via UART communication. Based on the user's the request and current state of the AC, the main controller is able to provide the proper response to the Bluno. Based on the main controller's command, the Bluno take control action to the the AC thermostat.

6.8.3 Air Conditioner Control System processing details

1. In our system, we added a new control circuit into the thermostat of the air conditioner. The added control circuit is mainly the relay circuit, shown in following diagram.

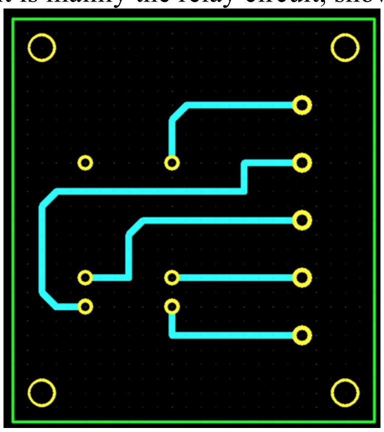


Figure. 6.8.2 The above diagram shows the PCB design of the relay circuit

7 User Interface Design

7.1 Application Control

(Qixin Huang designed and wrote the user Interface for Android application.)

7.1.1 Bluetooth Control

The Android application does not require any action to operate. The user only needs to click on the application logo for initiation. The application is then ran automatically. When the application stays ON, a message bar 'Open the lock now...' will be displayed on the screen. The application is also capable of showing any message that is received from the Bluno under the 'Received Data' area. The detail diagram or interface is shown in the following diagrams.

7.1.2 Web Page Control

The Website control panel can be divided into five parts. The first part of the webpage is the local weather, which display the weather condition of the household's area. The second part of the webpage is the heading of our Website, which also contains the website's slogan, logo, and a Google search bar. The third part is main control panel, which contains all the control buttons of our system. The fourth part of the website contains all the house condition of the system, which gets updated every 20 seconds. The fifth part of the website is a live IP camera.

7.2 Screen

7.2.1 Bluetooth Application Interface

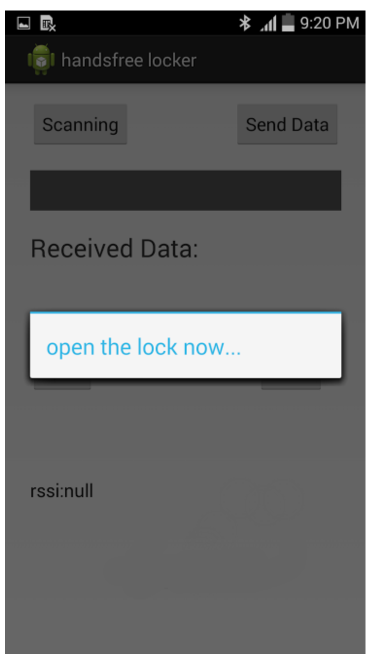


Figure 7.2.1: The Bluetooth function of mobile phone will be turned on automatically when the user opens the Android application. Then it will scan the Bluetooth device after opening.

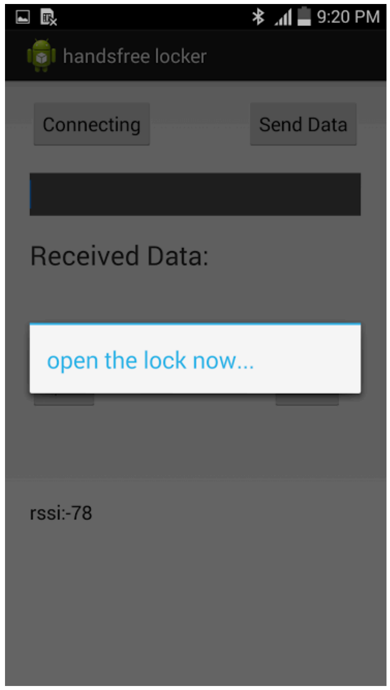


Figure 7.2.2: Once the application scanned Bluno, the connection status will go to "connecting", and the RSSI signal will be displayed on the screen.

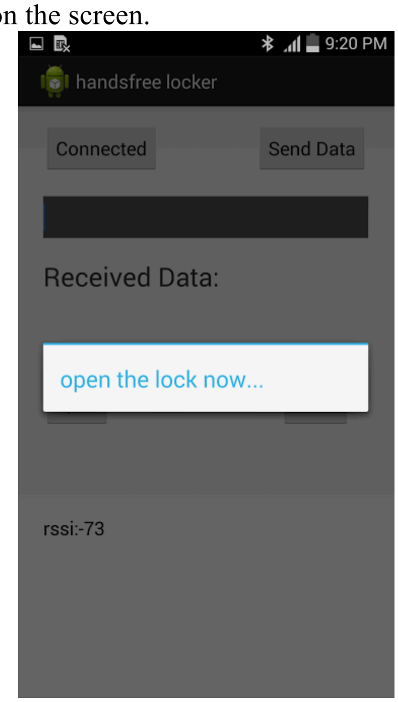


Figure 7.2.3: When Android device is connected to the Bluno, it will keep the connection and check for RSSI.

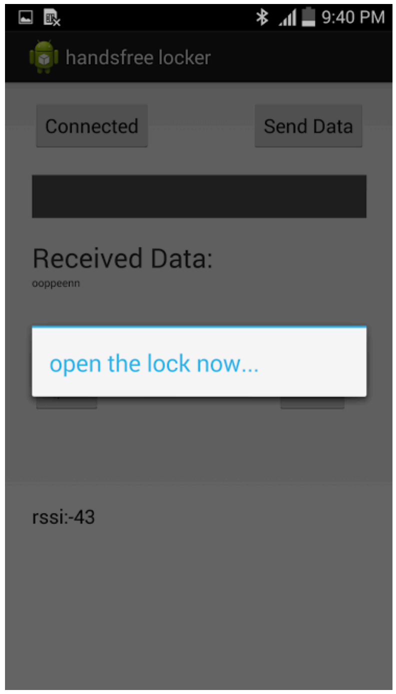


Figure 7.2.4: When the RSSI signal is larger than -50, the application will send passcode to Bluno to open the door lock.

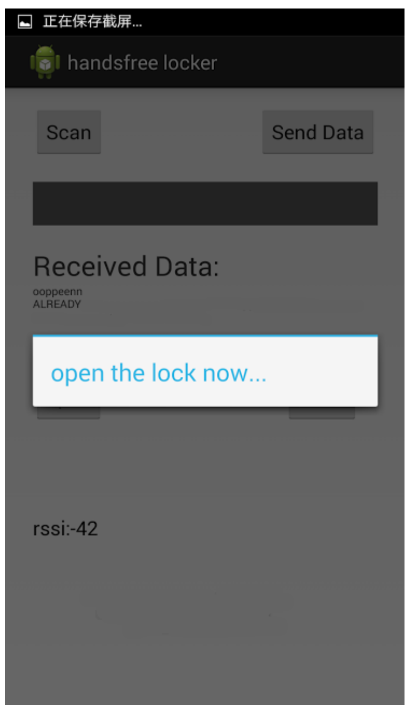


Figure 7.2.5: After opening the door, the Bluno will respond with ‘ALREADY’. Then, Android application will disconnect with the Bluno.

7.2.2 Web Page Interface

Figure 7.2.2.1: Web page control panel

7.3 Development system and components available

The Bluetooth Android application was created in the Eclipse software. For testing, we used the SAMSUNG Galaxy S4 to run the application. We also use “Logcat” to watch the application status.

8 * Experiment Design and Test Plan

8.1 * Design of Experiments

8.1.1 Internet Connection Test:

(Yancong Deng is responsible for the Internet Connection Test.)

1: Test Description:

The objective of this experiment is to ensure the WIFI module is able to connect to internet. Arduino Mega microcontroller connects to Internet through the WIFI Shield 101 Module. We add the network SSID and password of the WIFI router in the Arduino program. The WIFI module must be able to obtain connection to the Internet. we are able to verify the WIFI module's internet connection in the following two ways:1. checking the available connected internet devices in the router software or 2. Using Arduino Serial Monitor to Print WIFI status of this WIFI module using Arduino WIFI library. The WIFI connection test fails if we verify that the WIFI connection fails from either the router software or from the WIFI status printed from the Arduino Serial Monitor .

2: Setup

Prerequisite:

- WIFI Shield 101, or Ethernet shield connects to Arduino Mega
- Arduino is powered (5V)
- Home WIFI Router is available and working

Start Condition:

- The WIFI Shield is not connected with Internet

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1: Include Wi-Fi 101 AND SPI libraries in the program. 2: Setup Network SSID and password; and Setup Web server port on Port 80. 3: Used WIFI.begin() function with SSID and password 4: Start the server using server.begin() function. 5: Check the WIFI status using WIFI.status() function or check connection on router interface.	The Output WIFI status should be "Connected", or the device's Mac address and IP address should be shown on the router software.

Pass/Fail: 100%

8.1.2 Internet Recovery Test:

(Yancong Deng is responsible for the Internet Recovery Test.)

1: Test Description:
The objective of this experiment is to test the WIFI module is able to detect the lost connection of the WIFI module and reconnect to internet. The program continuously checks the WIFI connection status. If the connection is lost, the program calls reconnect function to attempt to reconnect to the Internet. The Internet Recovery test fails if it could not detect the loss of WIFI connection and could not reconnect to internet while the internet is available.

2: Setup

Prerequisite:

Prerequisite:

- WIFI Shield 101 connects to Arduino Mega
- Arduino is powered (5V)
- Home WIFI Router is available and working

Start Condition:

- The WIFI Shield is connected to the Internet

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Turn OFF the WIFI router 2. Print out current WIFI status on Serial monitor 3. Turn ON WIFI router 4. Wait for reconnection and print out WIFI status	The printed message should show that WIFI is disconnected After WIFI router is ON, the WIFI status should be connected.

Pass/Fail: 100%

8.1.3 Web Page Control Panel Test

(Yujie Cao, Yancong Deng, and Qixin Huang are responsible for the Web Page Control Panel Test.)

1: Description:
The web page control panel test ensures that by clicking the button on the webpage, the user is able to send the request to the main controller; in addition, the main controller is able to properly control the corresponding electrical devices based on the user's request.

2: Setup

Prerequisite:

- 10.1.1 pass
- 10.1.2 pass

Material:

- The main control system is powered(5V)
- The main controller's Internet connection is normal

Start Condition:
Step 1: Login into the webpage's home page by typing the IP address.

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Click Garage Door: open	1.Garage door open, 2.The "open" button in green shade 3. Garage door state: OPEN
Click Garage Door: close	1.Garage door close, 2.The "close" button in green shade 3. Garage door state: CLOSE
Click Garage Door: Small light	1.Garage small light on 2.The "Small light" button in green shade
Click Lights: Glight	1. Garage light is on 2. The "Glight" button in green shade
Click Lights: Glight in green shade	1. Garage light is off 2. The "Glight" button is not in green shade
Click Lights: Homelight	1. Homelight is on 2. The "Homelight" button in green shade 3. Home Brightness: value < 400
Click Lights: Homelight in green shade	1. Garage light is off 2. The "Homelight" button is not in green shade 3. Home Brightness: value > 400
Click Air-conditioner: Cold OFF	1. Turn on AC's cooling 2. The button become "Cold ON" in green shade 3 AC State: Cold On
Click Air-conditioner: Cold On(in green shade)	1. Turn off AC's cooling 2. The button become "Cold OFF" 3 AC State: Cooldown (in the first 10 minutes)

	off (after the 10 minutes)
Click Air-conditioner: Heat OFF	1. Turn on AC's heating 2. The button become "Heat ON" in green shade 3 AC State: Heat On
Click Air-conditioner: Heat On(in green shade)	1. Turn off AC's heating 2. The button become "Heat OFF" 3 AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Off	1. Turn off AC 2. AC button become "Off" in green shade 3.AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Home door: Open	1. open the home door 2. The button "Open" become "Open" in green shade 3.The button "Close" does not have green shade any more
Click Home door: Close	1. close the home door 2. The button "Close" become "Close" in green shade 3.The button "Open" does not have green shade any more
Click Sprinkler: OFF	1. turn on the sprinkler 2. the button become "ON" in green shade
Click Sprinkler: (On in green shade)	1. turn off the sprinkler 2. the button become "OFF"

Pass/Fail: 99%

8.1.4 Web Page Value Display Test

(Yancong Deng and Yujie Cao are responsible for the Web Page Value Display Test)

1: Description:

The web page value display test ensures that the webpage displays the current house's information by reading the sensors' value.

2: Setup

Prerequisite:

- 10.1.1 pass
- 10.1.2 pass

Material:

- the main control system is powered on
- Internet connection for the main controller work properly

Start Condition:

Step 1: Login into the webpage's home page by typing the IP address.

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Click Garage Door: open	1. Garage door state: OPEN
Click Garage Door: close	1. Garage door state: CLOSE
Home Temperature	display the home temperature in Celsius
Click Lights: Homelight	1. Home Brightness: value < 400
Click Lights: Homelight in green shade	1. Home Brightness: value > 400
Click Air-conditioner: Cold OFF	1. AC State: Cold On
Click Air-conditioner: Cold On(in green shade)	1. AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Heat OFF	1. AC State: Heat On
Click Air-conditioner: Heat On(in green shade)	1. AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Off	1.AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Home Motion	1. people moving in kitchen Motion reading > 600 2. no people moving kitchen motion reading = 0
Check Grass State	1.shows: Dry when humidity sensor reading

	< 200 2. shows: Good when humidity sensor reading > 200
Click Sprinkler: OFF	1. Sprinkler state: On
Click Sprinkler: (On in green shade)	1. Sprinkler state: OFF
Garage door distance: 1. < 55 cm 2. >55 cm and < 180 cm 3. >180 cm	Garage door state: 1. CLOSE 2. Error 3. Open
Car Position: 1. > 550 cm 2. >80cm and <115 cm 3. < 45cm	Car information: 1. no car in garage door 2. car in garage door 3. back door open

Pass/Fail: 99%

8.1.5 Timer Function Test

(Yujie Cao is responsible for the Timer Function Test)

1: Test Description:

The objective of this experiment is to test the internal timer function in Arduino Mega 2560. The timer 5 in Arduino Mega is used as the main timer of the program. We set up the timer with the prescaler of 256 and the preload timer value is 0. When the timer counts from 0 to 65536, then goes back to 0, the timer 5 interrupt will be called and a counter variable will increment by one in the interrupt.

The interrupt frequency (Hz) of the timer can be calculated by the formula:

Interrupt Frequency = (Arduino Bus clock) / (prescaler * (compare match register + 1))

Since the Arduino has Bus Clock speed at 16,000,000Hz, the interrupt frequency of the timer with prescaler at 256 should be: f = 16,000,000 / (256*65536+1) = 0.9536 Hz. Therefore, the period of the timer interrupt should be T = 1/f = 1.0485 s. In order word, each increment of the counter is approximately one second.

This test experiment is to measure the counter increment for each timer interrupt and verify that each increment should take about one second.

2: Setup

Prerequisite:

- Arduino Mega
- Timer 5 is available for use

Start Condition:

- Timer 5 register is properly set up
- Timer prescaler is set to 256

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Set TCNT5 = 0 TCCR5B |= (1 << CS12); TIMSK5 |= (1 << TOIE5); 2. Enable interrupt. 3. Print the increment counter in the interrupt function on the Serial Monitor 4. User physical timer to time each interrupt	Each output on the Serial Monitor should take about 1 second.

Pass/Fail: 100%

8.1.6 HTML Webpage Refresh Test:

(Yancong Deng is responsible for the HTML Webpage Refresh Test.)

1: Description:
The WIFI server is able to display a HTML webpage on browser when it receives request from a client. For our system, we have set up the refresh function to update the sensor value and clear any previous command code on the webpage. The purpose of this test is to ensure the refresh function works properly.

2: Setup

Prerequisite:

- Arduino server is on.

Start Condition:

- WIFI module is connected to the Internet

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. visit the web server using any Internet browser. 2. observe the webpage	The web page is refresh after the fixed period of time (depend on the setup of the HTML code, such as 20 second).

Pass/Fail: 90%

8.1.7 Garage Door Auto Close Test

(Yujie Cao is responsible for the Garage Door Auto Close Test.)

1: Description:

The garage door sensor detects whether the door is closed or not. When the garage door is open, the main controller will check the reading of the motion sensor. if the motion sensor's reading > 600+, We consider that it has people working in the garage. If there is no motion for 5 minutes continuously, the garage door will automatically close. The purpose of this test is to verify Garage door auto close function works

2: Setup

Prerequisite:

- The Arduino is able to control the garage door
- Arduino Timer is available

Material:

- Garage door work properly under manual control.
- Power source

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
No people in the garage. Open the garage door by using webpage.	The garage door will close in 5 minutes.
People walk in the garage. Open the garage door by using webpage.	The garage door will never close as long as there are people walking.
People walk in the garage. Open the garage door by using webpage. People leave later.	The garage door will close after the people left the garage in 5 minutes.

Pass/Fail: 100%

8.1.8 Garage Light Auto Off Test

(Yujie Cao is responsible for the Garage Light Auto Off Test.)

1: Description:

Garage light auto off function is used for turning off the light in the garage door in case people forget to turn off it. The main controller constantly checks whether the garage light is on or not. If the light is on, main controller will check whether the motion sensor detect people or not. if the main controller does not receive any people's motion information from motion sensor for 2 minutes, the main controller will turn off the garage light.

The purpose of this test is to verify garage light auto off function works

2: Setup

Material:

- Garage light works properly under manual control.
- Power source

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
No people in the garage. turn on the garage light through webpage	The garage light will be off in 2 minutes.
People move in the garage. turn on the garage light through webpage	The garage light will never be off as long as the people is moving
People move in the garage. Open the garage door by using webpage. people left	The garage light will be off after the people left the garage for 2 minutes.

Pass/Fail: 100%

8.1.9 Garage Light FSM Test

(Yujie Cao is responsible for the Garage Light FSM Test.)

1: Description:

Garage light FSM function's purpose is following: when the driver drives a car back to garage, when the car is moved into proper position range, the garage light will blink three times to notify the driver that he or she can park the car at the current position now.

The purpose of this test is to verify garage light auto off function works

2: Setup

Prerequisite:

- none

Material:

- Garage light
- main controller
- car position sensor (ultrasonic sensor)

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Step 1: car is not in garage Step 2: open the garage door Step 3: car stops at the proper range.	The garage light should blink three times and stay on for two minutes.

Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall(car distance < 80cm)	The garage light should blink when a car is in the proper range. and the light should be off when this car is too close to the wall(car distance < 80cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car stop too far to the wall(car distance > 115cm)	The garage light stay its current state without any action (car distance >115 cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall (car distance < 80cm) step4: driver move the car back to the proper range.	The garage light should blink when the car is in the proper range. and the garage light should be off when the car is too close to the wall(car distance < 80cm) at step4: the garage light should be on for two minutes

Pass/Fail: 95%

8.1.10 UART Communication Test

(Yancong Deng and Yujie Cao are responsible for the UART communication test.)

1: Test Description:
The objective of this experiment is to test the UART communication between Arduino Mega and Bluno at the range of 50 Feet. Both microcontrollers should be able to transmit and receive information. The purpose of this test is to verify the receive and transmit function on both microcontrollers work properly. The test fails when one of the microcontrollers does not receive a message or receive an incorrect message.

2: Setup

Prerequisite:
- Arduino Mega and Bluno
- Receive Pin, Transmission Pin, and Ground should be properly connected

Start Condition:
- Both microcontroller should be set up at the same Baud Rate

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Use Arduino Mega send a message to Bluno. For example, it sends a string "Hello\n" 2. When the Bluno detects or receives any data from the RX Pin, Bluno stores the data until '\n' is detect-	The Bluno should receive and output the message "Hello" on the Serial Monitor.

ed. 3. Bluno outputs the received message	
1. Use Bluno send a message to Arduino Mega. For example, it sends a string “Hello\n” 2. When the Arduino Mega detects or receives any data from the RX Pin, Bluno stores the data until ‘\n’ is detected. 3. Arduino Mega outputs the received message	The Arduino Mega should receive and output the message “Hello” on the Serial Monitor.

Pass/Fail: 85%

8.1.11 Main Door Auto Lock Function Test

(Yujie Cao and Yancong Deng are responsible for the the Main Door Auto Lock Function Test.)

1: Description:
The main door auto lock function detects whether the main door is closed or not. if the main controller detect that the main door lock is open, the main controller will check the main door lock state again after 5 minutes. after 5 minutes, if the main door lock is still open, the controller will close the main door lock automatically. The purpose of this test is to verify main door auto lock function works

2: Setup

Prerequisite:

- The Arduino is able to talk to Bluno through UART
- the main door lock is powered.
- Bluno is able to control the main door lock

Material:

- main door lock works properly under manual control.
- Power source for main door lock.
- power source for two controllers.

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Open the main door by using key	the main door lock will be closed in 5 minutes.
Open the main door by using passcode	the main door lock will be closed in 5 minutes.
Open the main door by using webpage	the main door lock will be closed in 5 minutes.

Pass/Fail: 100%

8.1.12 AC 10 Minutes Cool Down Test

(Yancong Deng and Yujie Cao are responsible for the AC 10 Minutes Cool Down test.)

1: Description:
The AC 10 minutes cool down function aims to protect the Air Conditioner's motor. To be specific, user is not allowed to turn back on AC if the AC is just turned off within 10 minutes. The purpose of this test is to verify AC 10 minutes cool down function works.

2: Setup

Prerequisite:
- The Arduino is able to talk to Bluno through UART communication
- the main door lock is powered.
- Bluno is able to control the main door lock

Material:
- main door lock works properly under manual control.
- Power source for main door lock.
- power source for two controllers.

Initial Condition: the AC is off

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Step 1: turn on AC cooling, Step 2: turn off AC cooling Step 3: turn on AC cooling again	The AC cooling can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC cooling, Step 2: turn off AC cooling Step 3: turn on AC heating again	The AC heating can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC heating, Step 2: turn off AC heating, Step 3: turn on AC heating again	The AC heating can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC heating, Step 2: turn off AC heating, Step 3: turn on AC cooling again	The AC cooling can not be turned on at step3 for 10 minutes after step2

Step 1: turn on AC heating, Step 2: turn on AC cooling again	The AC heating will be off. The AC cooling can not be turned on until 10 minutes cool down period passes
Step 1: turn on AC cooling, Step 2: turn on AC heating again	The AC cooling will be off The AC heating can not be turned on for 10 minutes cool down period passes

Pass/Fail: 100%

8.1.13 Bluetooth Android Application Connection Test

(Qixin Huang is responsible for the Bluetooth Android Application Connection Test.)

1: Description:

The connection between the Android phone and Bluno is the fundamental function of main door subsystem. The application can only connect to the verified Bluno. We want to know the Bluetooth connection distance and the success rate. The purpose of this test is to verify the distance that can keep the connection, and the connection success rate.

2: Setup

Prerequisite:

- The application has no bug to effect the connection process.
- The baud rates of application and Bluno are the same.
- The mobile phone must support Bluetooth 4.0.

Material:

- Bluno
- Android mobile phone which has installed application

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Click the connect button	The status changes from “Scanning” to “Connecting” when RSSI value is larger than -50, The status changes to “Connected” within 7 seconds.
After established the connection, move the Android devices away from the Bluno.	When the distance reached to some specific value, the status changes from “Connected” to “Disconnected”.

Pass/Fail: 90%

8.1.14 Bluetooth Android Application RSSI Detection

(Qixin Huang is responsible for the Bluetooth Android Application RSSI Detection)

1: Description:
The detection of RSSI value is a important part of this subsystem. The RSSI value has proportional relationship with the distance. The RSSI value gets larger when the mobile phone gets approaching to the Bluno. The application decides to send code to Bluno according to the RSSI value.

2: Setup

Prerequisite:

- The application has no bug to effect the RSSI detection process.
- he baud rates of application and Bluno are the same.
- The mobile phone must support Bluetooth 4.0.

Material:

- Bluno
- Android mobile phone which has installed application

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
Click the connect button	When the connection has been set up, the RSSI value will be displayed on the screen.
Keep the connection and increase the distance between the mobile phone and Bluno.	The RSSI value will be smaller with longer distance.

Pass/Fail: 95%

8.1.15 10.1.15 Data Exchange Between Android Application and Bluno

(Qixin Huang is responsible for the Data Exchange Between Android Application and Bluno)

1: Description:
The data exchange between Android application and Bluno is the last part of main door lock control system. What we concern about is the precision of transferring data.

2: Setup

Prerequisite:

- The baud rates of application and Bluno are the same.
- The mobile phone must support Bluetooth 4.0.

Material:

- Bluno

- Android mobile phone which has installed application

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1.Click the connect button	The mobile phone will set a connection to the Blu-no.
2.Write some information in the sending box, and click the send button.	The information can be displayed on the serial monitor of Bluno at Arduino IDE software.
3.Write some information at the serial monitor in Arduino IDE.	The information can be displayed on the screen of mobile phone.

Pass/Fail: 95%

8.1.16 Watchdog Timer Test:

(Yancong Deng is responsible for the Watchdog Timer Test.)

1: Test Description:
The objective of this experiment is to test Watchdog Timer function in the program. Watchdog Timer function is able to reset the entire program when the Watchdog timer is not properly reset within a period of time. The purpose of this test is to verify the reset function and the accuracy of the watchdog. This test fails if the function is not reset or re-set in unexpected time.

2: Setup

Prerequisite:

- Arduino Mega

Start Condition:

- watchdog timer is enable

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Print a string message on the setup function, such as “Start” 2. Set Watchdog timer to “WDTO_8S”, which require to reset the Watchdog timer every 8 seconds 3. Do not reset the Watchdog timer on the Main loop 4. Observe the Serial Monitor, and count the used time	The Serial Monitor on Arduino Mega should show message “Start” for every 8 seconds.

5. Reset the Watchdog timer on the Main loop for every 5 seconds 6. Observe the Serial Monitor, and count the used time	The Serial Monitor on Arduino Mega should not show message "Start" other than the initial message.

Pass/Fail: 100%

8.1.17 I2C Communication Test:

(Yancong Deng and Yujie Cao are responsible for the I2C communication test.)

1: Test Description:
The objective of this experiment is to test the I2C communication between Arduino Mega and Bluno within the range of 50 Feet. Both microcontrollers should be able to transmit and receive information. The purpose of this test is to verify the Receive function and Transmit function on both microcontrollers. The test fails when anyone of the microcontroller does not receive a message or receives an incorrect message.
2: Setup

Prerequisite:

- Arduino Mega and Bluno
- SDA Pin, SCL Pin, and Ground should be properly connected
- Include Wire.h library

Start Condition:

- Both microcontroller should be set up at the same channel

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Use Arduino Mega send a message to Bluno. For example, it sends a string "Hello\n" Using Function: Wire.beginTransmission(5); Wire.write("Hello\n"); Wire.endTransmission(); 2. When the Bluno detects or receives any data from Wire.onReceive() function 3. Bluno outputs the received message	The Bluno should receive and output the message "Hello\n" on the Serial Monitor.
1. Use Bluno send a message to Arduino Mega. For example, it sends a string "Hello\n" Using Function: Wire.beginTransmission(5); Wire.write("Hello\n"); Wire.endTransmission(); 2. When the Arduino Mega detects or receives any	The Arduino Mega should receive and output the message "Hello\n" on the Serial Monitor.

data from Wire.onReceive() function 3. Arduino Mega outputs the received message	

Pass/Fail: 85%

8.1.18 Main Door Lock Position Test:

(Yancong Deng is responsible for the Main Door Lock Position Test.)

1: Test Description:
The objective of this experiment is to test the sensor reading from the electrical door lock. The system is able to determine the current position of the door lock. The electrical door lock has two major sensors (Left and Right sensors), and they give High/Low digital output. The purpose of this test is to verify and match the current position of the lock.

2: Setup

Prerequisite:

- Arduino
- Addition wiring from the sensor to the Arduino

Start Condition:

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Close the door 2. Check the sensor readings 3. If the right sensor == 1 && left sensor == 1, the Door is on Closed position 4. Print out current position	The program should show the door is currently closed.
1. Close the door 2. Check the sensor readings 3. If the right sensor == 0 && left sensor == 1, the Door is on Closed position 4. Print out current position	The program should show the door is currently open.

Pass/Fail: 100%

8.1.19 Main Door Lock Control Test:

(Yancong Deng is responsible for the Main Door Lock Control Test.)

1: Test Description:
The objective of this experiment is to test the control system of the electrical door lock. After receiving a request to the controller, Open or Close, the controller should be able to determine the current position of the door lock and take actions accordingly.

2: Setup

Prerequisite:

- Arduino
- H-bridge control circuit to the Door Lock
- The Arduino is able to control the door lock

Start Condition:

- The Door lock and Arduino share common ground

3.Stimulus (procedure**) / 4. Response:**

Stimulus	Response
1. Close the door 2. Send "Close Door" request to the controller	The system does nothing
1. Close the door 2. Send "Open Door" request to the controller	The system controller closes the Lock
1. Open the door 2. Send "Close Door" request to the controller	The system controller opens the Lock
1. Open the door 2. Send "Open Door" request to the controller	The system does nothing

Pass/Fail: 100%

8.2 * Bug Tracking

(All team members contributed on this section.)

Date	Defect	Discovered by	Fix/ Investigate
11/23/15	Controller cannot receive correct Sensor values from Electrical Door Lock	Yancong Deng	Yancong Deng
11/23/15	The Door lock stops on the incorrect position when using Arduino to control	Yancong Deng	Yancong Deng
12/02/15	The original control system goes wrong for the Electrical Door Lock	Yancong Deng	Yancong Deng
12/03/15	The application has only one RSSI value for all the found devices.	Qixin Huang	Qixin Huang
12/13/15	The application cannot dismiss a dialog.	Qixin Huang	Qixin Huang

12/15/15	The WIFI server does not respond properly due to the limit of the flash memory on Arduino Uno.	Yancong Deng	Yujie Cao Yancong Deng
1/07/16	The light bulb is slightly flashed when it is turned OFF	Yujie Cao	Yujie Cao
1/11/16	The WIFI server stops response when too many requests are received.	Yancong Deng	Yancong Deng
11/12/16	The Website stays with control command, when the website is manually refreshed. The website resend the command to the server.	Yancong Deng	Yancong Deng
12/02/15	Sometimes the application gets some abnormal RSSI value.	Qixin Huang	Qixin Huang
1/15/16	The I2C communication cannot receive proper data over the range of 30 feet	Yancong Deng Yujie Cao	Yancong Deng Yujie Cao
1/20/16	The FSM of the Auto garage light cannot distinguish the coming object is whether a car or a person. In other words, this FSM is unable to recognize the human's behavior	Yujie Cao	Yujie Cao
1/26/16	The Bluetooth application connecting process is unstable.	Qixin Huang	Qixin Huang
2/05/16	The garage door height detection hardware setting disturbs the original sensor and causes the garage door malfunctions.	Yujie Cao	Yancong Deng
2/05/16	The website displays different formats at different browsers.	Qixin Huang	Qixin Huang
2/07/16	The AC current sensors' reading gets influenced by other electrical cables	Qixin Huang	Qixin Huang Yancong Deng Yujie Cao
2/10/16	The humidity sensor's reading fluctuates a lot between 200--500 when the soil is half dry or wet. (sensor reading range 0--650)	Yujie Cao	Yujie Cao
2/13/16	The motion sensor does not work properly as we tested in the lab since the detection angle needs to be adjusted.	Yujie Cao	Yujie Cao

8.3 * Quality Control

(All team members contributed on this section.)

Date	Time	Test Case Number	Test Track Number	Passed/ Failed	Reason for Failure	Retest?
11/22/15	10:00 AM	18: Main Door Lock Position	18– 1	Failed	The sensors are not receiving the correct amount of voltage and current	No
11/22/15	12:00 PM	18: Main Door Lock Position	18 – 2	Passed	N/A	Yes
11/24/15	8:00 PM	13.Bluetooth Application Connection Test	13 - 1	Passed	N/A	No
11/24/15	4:00 PM	1: internet connection Test	1-1	Passed	N/A	No
11/26/15	1:00 PM	1: internet connection Test	1-2	Failed	Improper WIFI initialization in the program for WIFI Shield 101	Yes
11/26/15	5:00PM	1: internet connection Test	1-3	Passed	N/A	Yes
11/28/15	5:00 PM	19: Main Door Lock Control Test:	2 - 1	Failed	Improper control signal in the program	No
11/28/15	22:00PM	14.Bluetooth Application RSSI Detection Test	14-1	Failed	Only one value for all the found devices	Yes
11/29/15	8:00 PM	19.Main Door Lock Control Test:	2 - 2	Failed	Wrong Programming setting for sensor reading	Yes
12/5/15	8:00 PM	19.Main Door Lock Control Test:	2 - 3	Passed	N/A	Yes
12/02/15	7:00 PM	2: Internet Recovery Test:	2 – 1	Passed	N/A	No
12/05/15	23:00PM	14.Bluetooth Application RSSI Detection Test	14-2	Passed	N/A	Yes
12/07/15	21:00PM	15.Data Exchange	15-1	Passed	N/A	No

		Between Android Application and Blu-no				
12/23/15	7:00 PM	4. Web Page Value Display	4 – 1	Passed	N/A	No
1/07/16	4:00 PM	17. I2C communica-tion	17-1	Failed	Communication dis-tance is too long for I2C	No
1/08/16	10:00 AM	10. UART communi-cation	10-1	Passed	N/A	No
1/08/16	3:00 PM	5. Timer Function	5 - 1	Passed	N/A	No
1/10/16	10:00 PM	6. HTML Web page Refresh test	6-1	Failed	Refreshing too fast, can not go back to home page	No
1/11/16	1:00 AM	6. HTML Web page Refresh test	6-2	Passed	N/A	Yes
1/12/16	10:00AM	8: Garage Light Auto Off Test	8-1	Failed	Motion Sensor is not sensitive enough	No
1/13/16	11:00AM	6.New HTML Webpage Test	6-3	Passed	N/A	No
1/14/16	1:30 PM	7: Garage Door Auto Close Test	7 – 1	Failed	Forget to add delay for webpage ‘operation	No
1/14/16	3:00 PM	7: Garage Door Auto Close Test	7-2	Passed	N/A	Yes
1/15/16	8:00PM	8: Garage Light Auto Off Test	8-2	Passed	N/A	Yes
1/16/16	2:00 PM	9: Garage Light FSM Test	9 – 1	Failed	sensor unable to recog-nize people walk across	No
1/16/16	5:00 PM	9: Garage Light FSM Test	9-2	Failed	take action when the sensor’s readings are wrong	Yes
1/17/16	10:00 AM	9: Garage Light FSM Test	9-3	Passed	N/A	Yes
1/20/16	8:00 PM	16: Watchdog Timer	16-1	Passed	N/A	No
1/28/16	11:00	11: Main Door Auto	11-1	Passed	N/A	No

	AM	Lock Function				
2/3/16	6:00 PM	12: AC 10 Minutes cool Down Test	12-1	Passed	N/A	No

8.4 * Identification of critical components

(All team members contributed in this section.)

1. WIFI Shield 101
2. Garage door: distance sensor
3. Car position: distance sensor
4. Bluno: Bluetooth module/ UUID
5. UART Communication between Arduino mega and Bluno
6. Android Application
7. Electrical Door Lock

8.5 * Items Not Tested by the Experiments

(All team members contributed in this section.)

We did not accurately test or measure the working distance and connection time for the WIFI Shield 101module. The test is not being made has the following reasons. The connection time for the WIFI module is not invariant, and it is highly dependent on the signal strength of the WIFI signal. In general, the WIFI module can obtain stable connection over time. Therefore, we decided that it is not necessary to test the connection time of the WIFI module.

We also did not perform a test or measure for the Bluetooth detection range. Since our system only require to detect Bluetooth devices within a close range, such as 20 feet, therefore, we did not design an accurate experiment to test the Bluetooth detection range. We have approximately measure the range, which is about 100 feet.

9 * Experimental Results and Test Report

9.1 Internet Connection Test:

9.1.1 Experiment 1

(Yujie and Yancong performed this experiment. Yancong wrote this section.)

1. The experiment 1is tested on the Ethernet Module. We use the Ethernet library and properly set up the Network SSID and password. The experiment result showed that the Ethernet can properly connect to the Internet.
2. We expect the Arduino can properly connect to the router, and we can view the connection status through either router interface or Arduino function. This experiment matches our expectation of this test.
3. Our test results showed that Ethernet Module works properly to connect to the Internet.
4. Since the Ethernet can properly connect to the Internet through router, there are not additional corrective action is needed.

9.1.2 Experiment 2

(Yujie and Yancong performed this experiment. Yancong wrote this section.)

1. This test is the retest for the Internet connection on the WIFI Shield 101 Module. We use the Arduino WIFI 101 library and set up the Network SSID and password as Experiment 1. However, the experiment result showed that the WIFI shield could not properly connect to the Internet. We are unable to verify the connection on either router interface or WIFI status on the controller.
2. We expect the Arduino can properly connect to the router, and we can view the connection status through either router interface or Arduino function. This experiment does not match our expectation of this test.
3. Our test results showed that WIFI Shield 101 Module did not work properly to connect to the Internet. Since the WIFI library is different than the Ethernet library. We have to take two things into consideration in the debugging process, which is the program for the WIFI connection and the actual Hardware of the WIFI shield.
4. Since the WIFI Shield 101 Module cannot properly connect to the Internet through router, we had reviewed our Arduino program and performed researches on the WIFI module. We had found out the improper configuration of the server port.

9.1.3 Experiment 3

(Yujie and Yancong performed this experiment. Yancong wrote this section.)

1. This test is the retest for the Internet connection on the WIFI Shield 101 Module. We had fixed some minor error in our program after research for the mistake. However, the experiment result showed that the WIFI shield is able to properly connect to the Internet.

2. We expect the Arduino can properly connect to the router, and we can will the connection status through either router interface or Arduino function. This experiment matches our expectation of this test.
3. Our test results showed that WIFI Shield 101 Module works properly to connect to the Internet after the fixed of Server port error.
4. Since the WIFI Shield 101 Module can properly connect to the Internet through router, there are not additional corrective action is needed.

9.2 Internet Recovery Test:

9.2.1 Experiment 1

(Yancong Deng performed and wrote this experiment.)

1. This experiment is to verify the performance of the Internet Connection Recovery System. For this test, the system correctly detected the disconnection of the Internet. It was able to enter the recovery function in the main loop and tried to reconnect to the Internet. After turning on the WIFI router, the function is able to reconnect to the Internet when the WIFI is available.

2. The test result perfectly matched with the expected result of this experiment.

3. Our test results showed the recovery program works as expected. It is able to quickly notice the disconnection and re-connect to the Internet when it is available.

4. No additional corrective actions required.

9.3 Web Page Control Panel Test:

9.3.1 Experiment 1

(Yancong, Qixin, and Yujie performed the experiment. Yujie wrote this section.)

1.

Stimulus	Result
Click Garage Door: open	1.Garage door open, 2.The “open” button in green shade 3. Garage door state: OPEN
Click Garage Door: close	1.Garage door close, 2.The “close” button in green shade 3. Garage door state: CLOSE
Click Garage Door: Small light	1.Garage small light on 2.The “Small light” button in green shade

Click Lights: Glight	1. Garage light is on 2. The “Glight” button is in green shade
Click Lights: Glight in green shade	1. Garage light is off 2. The “Glight” button is not in green shade
Click Lights: Homelight	1. homelight is on 2. The “Homelight” button is in green shade 3. Home Brightness: value < 400
Click Lights: Homelight in green shade	1. Garage light is off 2. The “Homelight” button is not in green shade 3. Home Brightness: value > 400
Click Air-conditioner: Cold OFF	1. Turn on AC’s cooling 2. The button becomes “Cold ON” in green shade 3 AC State: Cold On
Click Air-conditioner: Cold On(in green shade)	1. Turn off AC’s cooling 2. The button becomes “Cold OFF” 3 AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Heat OFF	1. Turn on AC’s heating 2. The button becomes “Heat ON” in green shade 3 AC State: Heat On
Click Air-conditioner: Heat On(in green shade)	1. Turn off AC’s heating 2. The button becomes “Heat OFF” 3 AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Off	1. Turn off AC 2. AC button becomes “Off” in green shade 3.AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Home door: Open	1. Open the home door 2. The button “Open” becomes “Open” in green shade 3.The button “Close” does not have green shade any more
Click Home door: CLose	1. Close the home door 2. The button “Close” becomes “Close” in green shade 3.The button “Open” does not have green shade any more

Click Sprinkler: OFF	1. Turn on the sprinkler 2. The button become “ON” in green shade
Click Sprinkler: (On in green shade)	1. Turn off the sprinkler 2. The button become “OFF”

2. The experiment result matches the expected results.
3. Our system works perfectly and does what the user want it to do
4. No Corrective action required.

9.4 Web Page Value Display Test:

9.4.1 Experiment 1

(Yancong, Qixin, and Yujie performed the experiment. Yujie wrote this section.)

1.

Stimulus	Result
Click Garage Door: open	1. Garage door state: OPEN
Click Garage Door: close	1. Garage door state: CLOSE
Home Temperature	Display the home temperature in Celsius
Click Lights: Homelight	1. Home Brightness: value < 400
Click Lights: Homelight in green shade	1. Home Brightness: value > 400
Click Air-conditioner: Cold OFF	1. AC State: Cold On
Click Air-conditioner: Cold On(in green shade)	1. AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Heat OFF	1. AC State: Heat On
Click Air-conditioner: Heat On(in green shade)	1. AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Click Air-conditioner: Off	1.AC State: Cooldown (in the first 10 minutes) off (after the 10 minutes)
Home Motion	1. People walk in kitchen: Motion reading > 600 2. Nobody in the kitchen: motion reading = 0
Check Grass State	1.Shows: Dry when humidity sensor reading < 200 2. Shows: Good when humidity sensor reading > 200

Click Sprinkler: OFF	1. Sprinkler state: On
click Sprinkler: (On in green shade)	1. Sprinkler state: OFF
Garage door distance: 1. < 55 cm 2. >55 cm and < 180 cm 3. >180 cm	Garage door state: 1. CLOSE 2. Error 3. Open
Car Position: 1. > 550 cm 2. >80 cm <115 cm 3. < 45cm	Car information: 1. no car in garage door 2. car in garage door 3. back door open

2. The experiment results match the expected results.
3. The webpage is able to display housing information correctly.
4. No corrective actions required.

9.5 Timer Function Test:

9.5.1 Experiment 1

(Yujie performed and wrote the experiment.)

1.

Stimulus	Result
1. Set TCNT5 = 0 TCCR5B |= (1 << CS12); TIMSK5 |= (1 << TOIE5); 2. Enable interrupt. 3. Print the increment counter in the interrupt function on the Serial Monitor 4. User physical timer to time each interrupt	Each output on the Serial Monitor should take about 1 second.

2. The result of the Timer Function matches the expected result of the test.
3. After the proper setup of the Timer, it is able to tick approximately by one second.
4. No additional corrective action is required

9.6 HTML Web Page Refresh Test:

9.6.1 Experiment 1

(Yancong Deng performed and wrote this experiment.)

1. This experiment is to test the performance of the Refresh function on HTML website. The HTML website is held on the Arduino Web server and set up to refresh the webpage every 3 seconds. The result of this test showed that the Website can be open in the initial visit. The website is unable to properly refreshed.
2. We expect the Arduino server can properly refresh to the website for every 3 seconds.
3. Our test results showed that Arduino Server did not work properly as expected. The reason of this error can be cause by the HTML program or the main Arduino server program.
4. Since the HTML Refresh function cannot properly perform. We had reviewed our main Arduino program by using Serial Monitor to debug. We realized the Arduino program has stop responding after the first client request. The system is stuck on the second client request. We have found out that the system stops functioning because the WIFI server received new request while it was still processing the previous task.

9.6.2 Experiment 2

(Yancong Deng performed and wrote this experiment.)

1. This test is to retest the performance of the Refresh function on HTML after the debugging change. The HTML website is now set up to refresh every 15 seconds. The result of this test showed that the Website can be properly refreshed.
2. We expect the Arduino can properly refresh for every 15 seconds, and it is able clear the Website address to /Home.
3. Our test results showed that the HTML refresh function completely match the expected performance.
4. Since the HTML refresh function can properly refresh the website in a fixed of time, there are not additional corrective action is needed.

9.7 Garage Door Auto Close Test:

9.7.1 Experiment 1

(Yujie and Yancong performed on this test. Yujie wrote this section.)

1.

Stimulus	Result
No people in the garage. Open the garage door by using webpage.	The garage door will close in 5 minutes.
People moving in the garage. Open the garage door by using webpage.	The garage door will close as soon as the garage door just open for 10 seconds
People moving in the garage. Open the garage door by using webpage. People are gone later.	The garage door will close as soon as the garage door just open for 10 seconds

2. The second and third part of this test fails.

3. In our program, we forget to add the time delay for the Garage Door Auto close function from the webpage operation.
4. Corrective actions taken: adding the 5 minutes Time delay for the Garage Door Auto close function from the webpage operation.

9.7.2 Experiment 2

(Yujie and Yancong performed on this test. Yujie wrote this section.)

1.

Stimulus	Result
No people in the garage. Open the garage door by using webpage.	The garage door will close in 5 minutes.
People walk in the garage. Open the garage door by using webpage.	The garage door will never close as long as the people is working
People walk in the garage. Open the garage door by using webpage. People leave later.	The garage door will close after the people left the garage in 5 minutes

2. Experiment result exactly matches expected results
3. Garage Door Auto Close Function works properly.
4. No Corrective actions required

9.8 Garage Light Auto Off Test:

9.8.1 Experiment 1

Yujie and Yancong Perform this test

1.

Stimulus	Result
No People in the garage. Turn on the garage light through webpage	The garage light is off after 2 minutes.
People come into the garage. turn on the garage light through webpage	The garage light is off sometimes even when the people is moving
People come into the garage. Open the garage door using webpage. people left	The garage light will be off after the people left the garage for 2 minutes.

2. The second part of the test fails
3. The reason for the failing part is that the motion sensor is not sensitive enough.
4. Adjust the angle of the motion sensor located, and increase the motion sensor's sensitivity to maximum, decrease the motion sensor's reading cycle to minimum.

9.8.2 Experiment 2

Yujie and Yancong perform this test

1.

Stimulus	Result
No people in the garage. turn on the garage light through webpage	The garage light is off in 2 minutes.
People move in the garage. turn on the garage light through webpage	The garage light is never off as long as the people is moving
People move in the garage. Open the garage door by using webpage. people left	The garage light is off after the people left the garage for 2 minutes.

2. Experiment results exactly match expected results
3. Motion sensor's adjustment works
4. No Corrective actions taken

9.9 Garage Light FSM Test:

9.9.1 Experiment 1

Yujie and Yancong performing the experiment

1.

Stimulus	Result
Step 1: car is not in garage Step 2: open the garage door Step 3: car stopping at the proper range.	The garage light should blink three times and stay on for two minutes.
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall(car distance < 80cm)	The garage light should blink when the car is in the proper range. and the light should be off when the car is too close to the wall(car distance < 80cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car stop too far to the wall(car distance > 115cm)	The garage light stay its current state without any action (car distance >115 cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall (car distance <	The garage light should blink when the car is in the proper range. and the garage light should be off when the car is too close to the wall(car distance < 80cm) at step4: the garage light should be on for two minutes

80cm) Step 4: driver move the car back to the proper range.	
when people walk by	The light is shut off or sometimes blink

2. The experiment results match the expected results, However, we meet new scenario:
When the people walks across the sensor, the sensor treat the human's movement as car's movement that leads to unexpected result.
3. The controller is unable to notice it is wrong data so that it run the Garage Light FSM function.
4. Adding the new function to the Garage Light FSM to distinguish the difference between the car's movement and people's movement.

9.9.2 Experiment 2

(Yujie and Yancong performing the experiment)
1.

Stimulus	Result
Step 1: car is not in garage Step 2: open the garage door Step 3: car stopping at the proper range.	The garage light should blink three times and stay on for two minutes.
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall(car distance < 80cm)	The garage light should blink when the car is in the proper range. and the light should be off when the car is too close to the wall(car distance < 80cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car stop too far to the wall(car distance > 115cm)	The garage light stay its current state without any action (car distance >115 cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall(car distance < 80cm) Step 4: driver move the car back to the proper range.	The garage light should blink when the car is in the proper range. and the garage light should be off when the car is too close to the wall(car distance < 80cm) at step4: the garage light should be on for two minutes
When people walk by	No action required

2. The experiment results match the expected results, However, we meet a new scenario:
due to the sensor's false reading is about 5%. The Garage Light FSM can not work properly.
3. The controller is unable to notice it is wrong data so that it run the Garage Light FSM function.
4. Adding the new function to the Garage Light FSM to ignore the sensor's false reading.

9.9.3 Experiment 3

(Qixin, Yujie and Yancong performing the experiment)
1.

Stimulus	Result
Step 1: car is not in garage Step 2: open the garage door Step 3: car stopping at the proper range.	The garage light should blink three times and stay on for two minutes.
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall(car distance < 80cm)	The garage light should blink when the car is in the proper range. and the light should be off when the car is too close to the wall(car distance < 80cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car stop too far to the wall(car distance > 115cm)	The garage light stay its current state without any action (car distance >115 cm)
Step 1: car is not in garage Step 2: open the garage door Step 3: car moving too close to the wall (car distance < 80cm) Step 4: driver move the car back to the proper range.	The garage light should blink when the car is in the proper range. and the garage light should be off when the car is too close to the wall(car distance < 80cm) at step4: the garage light should be on for two minutes
When people walk by	No action from FSM
Wrong reading from the distance sensor	No action from FSM

2. The experiment results match the expected results. No exception happens
3. FSM is working robust for the real world.
4. No action taken.

9.10 UART Communication Test:

9.10.1 Experiment 1

(Yujie and Yancong performed this test.)
1.

Stimulus	Result

1. Use Arduino Mega send a message to Bluno. For example, it sends a string "Hello\n" 2. When the Bluno detects or receives any data from the RX Pin, Bluno stores the data until '\n' is detected. 3. Bluno outputs the received message	The Bluno should receive and output the message "Hello" on the Serial Monitor.
1. Use Bluno send a message to Arduino Mega. For example, it sends a string "Hello\n" 2. When the Arduino Mega detects or receives any data from the RX Pin, Bluno stores the data until '\n' is detected. 3. Arduino Mega outputs the received message	The Arduino Mega should receive and output the message "Hello" on the Serial Monitor.

2. Experiment result exactly matches expected results. the Pass rate is 100%.
3. The Passing rate is 100% out of twenty communications. the communication works fine for our project that requires the transmission range is about 50 feet.
4. No Corrective actions taken

9.11 Main Door Auto Lock Test:

9.11.1 Experiment 1

(Yujie and Yancong performed this test.)

1.

Stimulus	Result
Open the main door by using key	The main door lock will be closed in 5 minutes.
Open the main door by using passcode	The main door lock will be closed in 5 minutes.
Open the main door by using webpage	The main door lock will be closed in 5 minutes.

2. Experiment results match the expected results
3. This function works good confirms that the UART communication is stable between two Arduinos
4. No Corrective actions taken

9.12 AC 10 Minutes Cool Down Test:

9.12.1 Experiment 1

(Qixin, Yancong, Yujie perform this test.)

1.

Stimulus	Result
Step 1: turn on AC cooling, Step 2: turn off AC cooling Step 3: turn on AC cooling again	The AC cooling can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC cooling, Step 2: turn off AC cooling Step 3: turn on AC heating again	The AC heating can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC heating, Step 2: turn off AC heating, Step 3: turn on AC heating again	The AC heating can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC heating, Step 2: turn off AC heating, Step 3: turn on AC cooling again	The AC cooling can not be turned on at step3 for 10 minutes after step2
Step 1: turn on AC heating, Step 2: turn on AC cooling again	The AC heating is off. The AC cooling can not be turned on until 10 minutes cool down period passes
Step 1: turn on AC cooling, Step 3: turn on AC heating again	The AC cooling is off The AC heating can not be turned on for 10 minutes cool down period passes

2. Experiment results match the expected result
3. It takes too much time to run the complete test
4. No corrective actions taken

9.13 Bluetooth Android Application Connection Test:

9.13.1 Experiment 1

(Qixin Performed this test.)

1.

Click the connect button	The status flashed many times between scanning and connecting.
Keep the connection and extend the distance between the mobile phone and Bluno.	The connection kept on at the beginning, when the distance reached to nearly 100 feet, the status changed from connected to disconnected.

2. The results are not match what the expected.

3. The Bluetooth application may has not found the device yet, but the application try to connect the device.
4. Extend the searching time.

9.13.2 Experiment 2

(Qixin Performed this test.)
1.

Click the connect button	The status started to change from scanning to connecting, then changed to connected.
Keep the connection and extend the distance between the mobile phone and Bluno.	The connection kept on at the beginning, when the distance reached to nearly 100 feet, the status changed from connected to disconnected.

2. Experiment results match the expected results
3. Sometimes it takes a long time to scan the device.
4. No corrective actions taken

9.14 Bluetooth Android Application RSSI Detection:

9.14.1 Experiment 1

(Qixin Performed this test.)
1.

Click the connect button	When the connection has been set up, only one RSSI value for all the devices.
Keep the connection and extend the distance between the mobile phone and Bluno.	The RSSI value was smaller with the extending distance.

2. The results are not match what the expected.
3. The RSSI may calculate for all the devices.
4. Change another way to get the RSSI.

9.14.2 Experiment 2

(Qixin Performed this test.)
1.

Click the connect button	When the connection has been set up, the RSSI value was displayed on the screen.
Keep the connection and extend the distance between the mobile phone and Bluno.	The RSSI value was smaller with the extending distance.

2. Experiment results match the expected results

3. When the connection took long time, the RSSI value may be stuck.
4. No corrective actions taken

9.15 Data Exchange Between Android Application and Bluno:

9.15.1 Experiment 1

(Qixin Performed this test.)

1.

Click the connect button	The mobile phone set a connection to the Bluno.
Write some information in the sending box, and click the send button.	The information has be displayed on the serial monitor of Bluno at Arduino IDE software.
Write some information at the serial monitor in Arduino IDE.	The information was displayed on the screen of mobile phone.

2. Experiment results match the expected results
3. The information transmission is instant and correct.
4. No corrective actions taken

9.16 Watchdog Timer Test:

9.16.1 11.16.1 Experiment 1

(Yancong performed the experiment.)

1. This experiment tests the performance of the Watchdog Timer function. The Watchdog Timer function is set to about 8 seconds, which meant if the Watchdog Timer was not being reset, it re-set the entire program. The test result of this experiment showed the Watchdog can properly reset the program if the Timer is not being reset.

2. The Watchdog Timer function works properly and perfectly match our expected result.

3. Our test results showed that Watchdog Timer function works properly to reset program if the program does not reset the Watchdog Timer.

4. No additional corrective actions are required

9.17 I2C Communication Test:

9.17.1 Experiment 1

(Yancong and Yujie perform this test.)

1.

Stimulus	Result

1. Use Arduino Mega send a message to Bluno. For example, it sends a string "Hello\n" Using Function: Wire.beginTransmission(5); Wire.write("Hello\n"); Wire.endTransmission(); 2. When the Bluno detects or receives any data from Wire.onReceive() function 3. Bluno outputs the received message	N/A
1. Use Bluno send a message to Arduino Mega. For example, it sends a string "Hello\n" Using Function: Wire.beginTransmission(5); Wire.write("Hello\n"); Wire.endTransmission(); 2. When the Arduino Mega detects or receives any data from Wire.onReceive() function 3. Arduino Mega outputs the received message	N/A

2. Experiment results do not match expected results
3. The communication range is 50 feet, I2C does not work for our project
4. No Corrective actions taken

9.18 Main Door Lock Position Test:

9.18.1 Experiment 1

(Yancong performing the experiment.)

1.

Stimulus	Result
1. Close the door 2. Check the sensor readings 3. If the right sensor == 1 && left sensor == 1, the Door is on Closed position 4. Print out current position	N/A
1. Close the door 2. Check the sensor readings 3. If the right sensor == 0 && left sensor == 1, the Door is on Closed position 4. Print out current position	N/A

2. Experiment results do not match expected results
3. Sensors are not receiving the correct amount of voltage and current
4. Modify the control circuit

9.18.2 Experiment 2

(Yancong performing the experiment.)

1.

Stimulus	Result
1. Close the door 2. Check the sensor readings 3. If the right sensor == 1 && left sensor == 1, the Door is on Closed position 4. Print out current position	The program should show the door is currently Closed.
1. Close the door 2. Check the sensor readings 3. If the right sensor == 0 && left sensor == 1, the Door is on Closed position 4. Print out current position	The program should show the door is currently Open.

2. Experiment results match expected results
3. N/A
4. No corrective actions taken

9.19 Main Door Lock Control Test:

9.19.1 Experiment 1

(Yancong performing the experiment.)

1.

Stimulus	Result
1. Close the door 2. Send "Close Door" request to the controller	the lock's motor break the mechanical system
1. Close the door 2. Send "Open Door" request to the controller	the lock's motor break the mechanical system
1. Open the door 2. Send "Close Door" request to the controller	the lock's motor break the mechanical system
1. Open the door 2. Send "Open Door" request to the controller	the lock's motor break the mechanical system

2. Experiments result does not match expected results
3. Does not get the position signal from the lock in the program.
4. We need to check the lock's position signal and add these readings to the program

9.19.2 Experiment 2

(Yancong performing the experiment.)

1.

Stimulus	Result
1. Close the door 2. Send “Close Door” request to the controller	The system controller opens the Lock
1. Close the door 2. Send “Open Door” request to the controller	The system does nothing
1. Open the door 2. Send “Close Door” request to the controller	The system does nothing
1. Open the door 2. Send “Open Door” request to the controller	The system controller closes the Lock

2. Experiments result does not match expected results
3. Improper control signal usage in the program.
4. Check and correct the signal reading circuit.

9.19.3 Experiment 3

(Yancong performing the experiment.)

1.

Stimulus	Result
1. Close the door 2. Send “Close Door” request to the controller	The system does nothing
1. Close the door 2. Send “Open Door” request to the controller	The system controller closes the Lock
1. Open the door 2. Send “Close Door” request to the controller	The system controller opens the Lock
1. Open the door 2. Send “Open Door” request to the controller	The system does nothing

2. Experiment results match expected results
3. The control function finally works.
4. No corrective actions taken

10 * Conclusion and Future Work

10.1 * Conclusion

(All team members contributed to this section.)

Our senior design project is the Intelligent Home Control System. Our group have three members, who are Yancong Deng, Qixin Huang, and Yujie Cao. After Eighteen weeks' dedicated work, we have successfully implemented a fully functional and robust Internet-connected Home Control System. We are proud of our senior design project. We believe the idea of home control system can be one of the most advanced and popular technology in the future. It would definitely have a great impact in the future living style and provide convenience for human.

Our Intelligent Home Control System is capable of controlling multiple devices in the house environment through the Internet webpage. So far, our system is able to control electrical door lock, garage door, sprinkler, light, and air conditioner. Our system controller collects sensor data from varieties of sensors, such as temperature, motion, brightness, soil humidity, and current sensors.

By analyzing the sensor data, we are able to design and implement home automation system. Yujie is responsible for all the automation systems. The automation systems in IHCS include Auto Garage Door System, Auto Garage Light System, Car Position Detection, Home Light System, and Auto Closing Main Door function. For example, The Auto Garage Door System is able to detect many motion within the garage area. If user forgets to close his or her door on the way out, our system can be able to help. The system will continuously seek motion in the garage. If no motion is detected consecutively for five minutes, our system will close the garage door automatically.

Our system not only controls all the electronic devices in the house, but also displays real-time house information on the website. Arduino microcontroller hosts a online web server through HTTP protocol. It is capable of updating real-time acquired data onto the webpage. User is able to acquire any status of electronic devices or house condition through the webpage. For our Electrical Door Lock Control System, we add a new control system into a commercial electrical lock. We used the bypass control method for this system, which makes sure only one control pathway is enable at a time. In addition, Yancong is s responsible for the Web server program and the Electrical Door Lock System.

Moreover, our Electrical Door Lock can be controlled by Android devices automatically. Qixin is responsible for the Bluetooth Application and the Automatic Main Door System. In our system, the slave microcontroller, Bluno, has embedded with Bluetooth 4.0 Module, which is capable of connecting to Android devices through Bluetooth communication. When the Android devices approach near the main door, the Android application will detect the Bluetooth RSSI of the Bluno. By analyzing the detected Bluetooth UUID and Mac Address, the system is able to automatically open the door for the authorized users.

However, due to the time limitation, comparing IHCS with our original design objectives, some desired objectives or original goals were not achieved. Our project has not yet achieved the Power Monitor System, Database, Auto Garage System with car identification, Email Alarm System, Android application in background mode.

Yujie Cao is not able to implement the home power monitor system due to the following technical reasons. "The only way to know that know the power usage for home electricity meter is that there is an infrared light shining when a certain amount of energy has been used" was Said by the Riverside Public Utility Electrical Engineer. Yujie was under the impression that the implementation of the Power Monitoring System would be too difficult for a limited of time. There are several reasons of this impression. First, it is expensive to buy a very precise and stable infrared light receiver. This system requires to install a small circuit of the infrared light receiver outside of the house environment. It would be difficult to protect the circuit outside due to the changing weather.

Due to the time limitation, Yujie is not able to implement a database service for our system. Since all the team members were not familiar to the database or SQL programming. It would require excessive amount of time to develop a suitable database for our system. Therefore, we designed to respite the implementation of this objective.

Moreover, IHCS was not able to automatically open the garage door when a specific car present in front of the door. In fact, we have come up with two approach to solve this problem, which are using wireless RFID transceiver or Computer Vision to detect the specific license plate. However, due to the budget limitation, we have to give up on the RFID method. The long distance RFID receiver is quite expensive, and the cheap RFID card's detection range is about 5 cm, which is too short for our project. On the other hand, due to the time limitation, Yancong did not have enough time to finish the image processing programming for the license plate recognition.

Yancong also tried to set up an Email service to send out notifications when certain events were triggered. Our original plan is to send a trigger flag to a third party database, such as Carriots. Then, the third party database would automatically send an Email to user. However, our controller serves used as a web server, which is not a client, but the Carriots requires our controller to send request as a web client. Due to this conflict, we are unable to setup the Email service.

Furthermore, Qixin is unable to do enough research on making the application running at the background. Qixin also tried to set up the Bluno in master mode. Unfortunately, since the Bluno is unable to acquire any Bluetooth UUID of surrounding devices. Bluno is unable to identify the correct devices for connection. As a result, we decide to set up the Bluno in slave mode. In this case, Android Phone actively searches for Bluno.

In the development of our Senior Design Project, we have learnt a lot of new knowledge and skills while implementing the IHCS. First of all, we enhanced our self-learning abilities and problem solving skills when we met and solved so many new encountered problems. Secondly, we realized that building a real commercial product has a huge different than building a demo prototype because the real product requires all the components and programs work stably and reliably. In comparison with the laboratory in the school, we encountered a lot of practical challenges that we have never met before. Such as rebuilding the wire according to our need; using UART instead of I2C due to 50 feet transmission range. We also learnt a lot new programming skills such as PCB design, Java, HTML, CSS, and JavaScript.

After finished the design of our project, we realize that building a prototype is very different from building an actual product. In order to mass produce our system, the actual product requires to modelize and standardize every part of the system. We also realize a good product needs a group of people working closely together. However, during our senior project, we have not only improve our technical skills and knowledge as engineers, but also develop a profound friendship between team members.

10.2 Future Work

In the future, we would like to add database and more robust Internet server into our system. IHCS should be capable of saving all the house information in the database. User is able to analyze house power usage from the database. In addition, multiple control systems can be added into the system, such as Car detection and Auto Brightness adjustment. With a larger online server, multiple users should be able to independently communicate with their unique system at the same time through our server.

10.3 * Acknowledgement

We would like to thank Dr. Ping Liang for advising and mentoring us throughout the entire design process. He gave us so many valuable ideas and advices, which enlightened and guided us along the way. Dr. Liang not only gave advices and suggestion for our project, but also helped us to improve our professional presentation skill, technical writing, and many electrical related concepts.

We would also like to thank Dr. Roman Chomko and Dr. Nissim Amos for their lectures, which provides lots of engineering related knowledge. We have learned many knowledge that is not only beneficial for us as students, but also beneficial to our future careers.

We would also like to thank Dear Pavle Kirilov for helping us to solder the Electrical Door Lock and implement the PCB for our circuit.

11 * References

[1] "Android bluetooth 4.0 LTE", *Blog.csdn.net*, 2016. [Online]. Available: http://blog.csdn.net/chaoyue0071/article/details/43450183. [Accessed: 18- Mar- 2016].
[2] "Arduino - ArduinoBoardMega2560", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega2560. [Accessed: 18- Mar- 2016].
[3] "Arduino - ArduinoWiFiShield101", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoWiFiShield101. [Accessed: 18- Mar- 2016].
[4] "Arduino - EEPROM", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Reference/EEPROM. [Accessed: 18- Mar- 2016].
[5] "Arduino - Interrupts", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Reference/Interrupts. [Accessed: 18- Mar- 2016].
[6] "Arduino - SPI", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Reference/SPI. [Accessed: 18- Mar- 2016].
[7] "Arduino - WiFi", *Arduino.cc*, 2016. [Online]. Available: https://www.arduino.cc/en/Reference/WiFi. [Accessed: 18- Mar- 2016].
[8] "Arduino Playground - ArduinoReset", *Playground.arduino.cc*, 2016. [Online]. Available: http://playground.arduino.cc/Main/ArduinoReset. [Accessed: 18- Mar- 2016].
[9] "Arduino Playground - OneWire", *Playground.arduino.cc*, 2016. [Online]. Available: http://playground.arduino.cc/Learning/OneWire. [Accessed: 18- Mar- 2016].
[10] "arduino-info - Timers-Arduino", *Arduino-info.wikispaces.com*, 2016. [Online]. Available: https://arduino-info.wikispaces.com/Timers-Arduino. [Accessed: 18- Mar- 2016].
[11] "Bluetooth | Android Developers", *Developer.android.com*, 2016. [Online]. Available: http://developer.android.com/intl/zh-cn/guide/topics/connectivity/bluetooth.html. [Accessed: 18- Mar- 2016].
[12] "Bluetooth 4.0", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Bluetooth#Bluetooth_v4.0. [Accessed: 18- Mar- 2016].
[13] "Bluno SKU:DFR0267 - Robot Wiki", *Dfrobot.com*, 2016. [Online]. Available: http://www.dfrobot.com/wiki/index.php/Bluno_SKU:DFR0267. [Accessed: 18- Mar- 2016].
[14] "CSS Tutorial", *W3schools.com*, 2016. [Online]. Available: http://www.w3schools.com/css/default.asp. [Accessed: 18- Mar- 2016].
[15] "DFRobot/BlunoBasicDemo", *GitHub*, 2016. [Online]. Available: https://github.com/DFRobot/BlunoBasicDemo/tree/master/Android/BlunoBasicDemo. [Accessed: 18- Mar- 2016].
[16] "Google Search Source Code", *Dreamdu.com*, 2016. [Online]. Available: http://www.dreamdu.com/webbuild/google_search/. [Accessed: 18- Mar- 2016].
[17] "How to constantly check bluetooth RSSI", *Blog.csdn.net*, 2016. [Online]. Available: http://blog.csdn.net/dull_boy2/article/details/8978576. [Accessed: 18- Mar- 2016].
[18] "How To Embed Foscam FI9821W Video Feed Into Your Web Page", *YouTube*, 2016. [Online]. Available: https://www.youtube.com/watch?v=wFwRnZESNi0. [Accessed: 18- Mar- 2016].
[19] "HTML Tutorial", *W3schools.com*, 2016. [Online]. Available: http://www.w3schools.com/html/. [Accessed: 18- Mar- 2016].
[20] "HTML weather code", *Blog.forecast.io*, 2016. [Online]. Available: http://blog.forecast.io/forecast-embeds/. [Accessed: 18- Mar- 2016].

[21] "JavaScript Tutorial", *W3schools.com*, 2016. [Online]. Available: http://www.w3schools.com/js/default.asp. [Accessed: 18- Mar- 2016].
[22] "Microcontroller - AVR ATmega32 - Introduction to UART - First of Many UART Tutorials", *YouTube*, 2016. [Online]. Available: https://www.youtube.com/watch?v=e8uEKWpVf10. [Accessed: 18- Mar- 2016].
[23] "PIR Motion Sensor", *PIR Motion Sensor*, 2016. [Online]. Available: https://learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf. [Accessed: 18- Mar- 2016].
[24] "Powerbolt 2 | Kwikset, Maker of SmartKey & Kevo Door Hardware | Locks, Handlesets, Deadbolts & Door Knobs", *Kwikset.com*, 2016. [Online]. Available: http://www.kwikset.com/products/styles/powerbolt-deadbolt-2.aspx. [Accessed: 18- Mar- 2016].
[25] "RS-232", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/RS-232. [Accessed: 18- Mar- 2016].
[26] "SainSmart Photosensitive Resistance Sensor Module for Arduino UNO Mega2560 Robot 3D Printing, Arduino, Robotics | Sainsmart", *Sainsmart.com*, 2016. [Online]. Available: http://www.sainsmart.com/sainsmart-photosensitive-resistance-sensor-module-for-arduino-uno-mega2560-robot.html. [Accessed: 18- Mar- 2016].
[27] A. second, "Android: Update Bluetooth RSSI every second", *Stackoverflow.com*, 2016. [Online]. Available: http://stackoverflow.com/questions/16252242/android-update-bluetooth-rssi-every-second. [Accessed: 18- Mar- 2016].
[28] "Serial Peripheral Interface Bus", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus. [Accessed: 18- Mar- 2016].
[29] D. Team, "Bluetooth Android Application", *Arduino.cn*, 2016. [Online]. Available: http://www.arduino.cn/thread-16311-1-1.html. [Accessed: 18- Mar- 2016].
[30] "Temperature Sensor DS18B20", *Temperature Sensor DS18B20:*, 2016. [Online]. Available: http://henrysbench.capnfatz.com/henrys-bench/ds18b20-arduino-user-manual-introduction-and-contents/ds18b20-arduino-users-manual-part-3-reading-a-single-temperature/. [Accessed: 18- Mar- 2016].
[31] D. TheMan, "Arduino Watchdog Timer (WDT) Example Code", *Big Dan the Blogging Man*, 2014. [Online]. Available: https://bigdanzblog.wordpress.com/2014/10/24/arduino-watchdog-timer-wdt-example-code/. [Accessed: 18- Mar- 2016].
[32] "Ultrasonic Ranging Module HC - SR04", *Ultrasonic Ranging Module HC - SR04*, 2016. [Online]. Available: http://www.micropik.com/PDF/HCSR04.pdf. [Accessed: 18- Mar- 2016].
[33] "Universal asynchronous receiver/transmitter", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter. [Accessed: 18- Mar- 2016].
[34] "Universal asynchronous receiver/transmitter", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter. [Accessed: 18- Mar- 2016].

12 * Appendices

Appendix A: Parts List

Module	Part	Link	Price
Main Server Control-ler & Sensors	Arduino MEGA 2560 R3	http://www.amazon.com/ATmega2560-16AU-ATMEGA16U2-Arduino-MEGA2560-Duemi-lanove/dp/B00OH21CRM/ref=sr_1_1?ie=UTF8&qid=1456990799&sr=8-1&keywords=aRDUINO+mega	$13.19
	8 Channel DC 5 V Relay Module	http://www.amazon.com/JBtek-Channel-Relay-Arduino-Raspber-ry/dp/B00KTELP3I/ref=sr_1_2?ie=UTF8&qid=1456990319&sr=8-2&keywords=aRDUINO+RELAY	$8.98
	Electrical Box	Home depot	$22.99
	Brightness Sensor	http://www.ebay.com/itm/Photosensitive-brightness-resistance-sensor-module-Light-intensity-detect-SN-/262073406743?hash=item3d04cad517:g:DiAAAOSwVL1WDOHM	$0.99
	Current Sensor	http://www.ebay.com/itm/5A-Range-AC-Current-transformer-module-Current-sensor-module-/151505189487?hash=item234669ba6f:g:hVQAAOxyXHpSQQ5~	$2.89
	Motion sensor	http://www.ebay.com/itm/HC-SR501-Infrared-PIR-Motion-Sensor-Module-for-HOT-Arduino-Raspberry-pi-Best-SM-/321988035310?hash=item4af7fb82ee:g:5UMAAOxyP4dTcvnN	$1.17
	Soil Mois-ture Sen-sor	http://www.ebay.com/itm/New-Useful-Soil-Hygrometer-Moisture-Water-Sensor-Detection-Module-for-Arduino-/151740596265?hash=item235471c029:g:f~EAAOSwMmBVnfFW	$0.99
	Ultrasonic Sensor	http://www.ebay.com/itm/1pcs-Ultrasonic-Module-HC-SR04-Distance-Measuring-Transducer-Sensor-for-Arduino-/400985326881?hash=item5d5c968521:g:kLQAAOxyNyFS-xFw	$1.03

	Temperature Sensor	http://www.ebay.com/itm/Waterproof-DS18B20-Temperature-Temp-Sensor-Digital-Thermal-Probe-for-Arduino-EA-/251886483181?hash=item3aa59ab6ed:g:VwoAAOSwBahVC82e	$5.88
	Wire	http://www.ebay.com/itm/Primary-Wire-5LLK7-Automtv-AWG-14-100-Ft-Blue-EC9-3-/400407316410?hash=item5d3a22c7ba:g:s5cAAMXQk-FREDzq	$33

Module	Part	Link	Price
Electrical Door Lock	Bluno	http://www.dfrobot.com/index.php?route=product/product&product_id=1044#.Vtf0qJMrLdQ	$34.90
	H-Bridge Driver	http://www.ebay.com/itm/4-pcs-UCN5804B-UCN5804-Stepper-Motor-Translator-Driver-IC-Chip-/151743865182?hash=item2354a3a15e:g:leQAAOSwjVVVodwg	$4.92
	Relay Module	http://www.ebay.com/itm/Kootek-8-Channel-DC-5V-Relay-Module-for-Arduino-Raspberry-Pi-DSP-AVR-PIC-ARM-BH-/111822234526?hash=item1a091fe39e:g:ZiQAAOSwiwVWREWU	$5.97
	Electrical Box	Home depot	$12.99

Appendix B: Software List

- Eclipse IDE
- Arduino IDE
- Sublime Text 2

Appendix C: Link to Project Code

https://drive.google.com/folderview?id=0B2VjPyYzfuZkelF2SWZfU2dnTzA&usp=sharing