VOICE CONTROLLED HOME AUTOMATION SYSTEM

By

Zhe Gong

Hongchaun Li

Final Report for ECE 445, Senior Design, Fall 2014

TA: Haoyu Wang

10 December 2014

Project No. 13

Abstract

This project builds a system that can remotely control on and off of multiple power sockets in different rooms, each with corresponding voice command, thus conveniently manage different electric equipment by voice. The project verifications are met and design goal is successfully achieved, however noise and distance handling may need future development.

Contents

1. Introduction 1
   1.1 Core Unit 1
      1.1.1 Microphone 1
      1.1.2 Microprocessor 1
      1.1.3 Power Electronics 2
      1.1.4 Transmitter: 2
   1.2 Remote Unit 2
      1.2.1 Receiver 2
      1.2.2 Microcontroller 2
      1.2.3 Relay 2
2. Design 3
   2.1 Design procedure 3
      2.1.1 Power electronics 3
      2.1.2 Microprocessor 3
      2.1.3 Transmitter and receiver 3
      2.1.4 Microcontroller 3
      2.1.5 Microphone 3
      2.1.6 Relay 3
   2.2 Design Details 4
      2.2.1 Power electronics 4
      2.2.2 Microprocessor 5
      2.2.3 Transmitter and receiver 6
      2.2.4 Microcontroller 7
3. Design Verification 8
   3.1 Power electronics 8
   3.2 Transmitter and receiver 8
   3.3 Microcontroller 9
   3.4 Microprocessor 10
   3.5 Completed project 10

4. Costs 11
4.1 Parts 11
4.2 Labor 11
5. Conclusion 12
5.1 Accomplishments 12
5.2 Uncertainties 12
5.3 Ethical considerations 12
5.4 Future work 13
References 14
Appendix A Requirement and Verification Table 15
Appendix B Schematic of Remote Unit 17
Appendix C Microprocessor software flow chart 18
Appendix D Microcontroller software flow chart 19

1. Introduction

In the home environment nowadays there's a demand of voice controlled switch system, which can conveniently manage different electric equipment by voice. Currently there are voice controlled products limited to a single power socket. As a result we are motivated to build a system that can remotely control on and off of multiple power sockets in different rooms, each with corresponding voice command, to create an automated and comfortable home environment. The report will describe the blocks the project divided into, discuss the design procedure and details, requirement and verification, cost and draw conclusions. The following are the top-level block diagrams and descriptions.

1.1 Core Unit

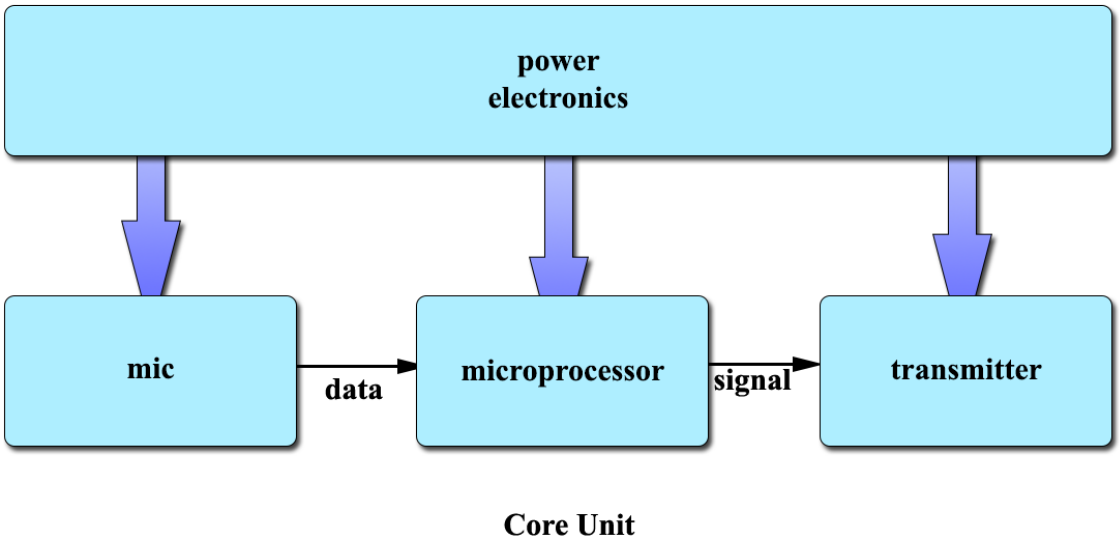


Figure 1 Core unit block diagram

Figure 1 shows the block diagram of Core Unit. In proposal we have a block of microcontroller, relay and power socket connected to microprocessor in Core Unit, which is similar to remote unit but with local wire connection. In the following design process, we decide to remove it because with the same function as remote unit, the wire controlled power socket is redundant.

1.1.1 Microphone

Microphone captures users' voice commands.

1.1.2 Microprocessor

Microprocessor (Raspberry pi) collect the data and do the first phase analyze. It sends data to online engine for further analyze and receive plain text. It analyzes the text to send command to different unit. The microprocessor should also be able to communicate via the Ethernet port (which is already provided by Raspberry pi). Raspberry pi has HDMI & USB port that could be connect to a keyboard & monitor for user to change setting.

1.1.3 Power Electronics

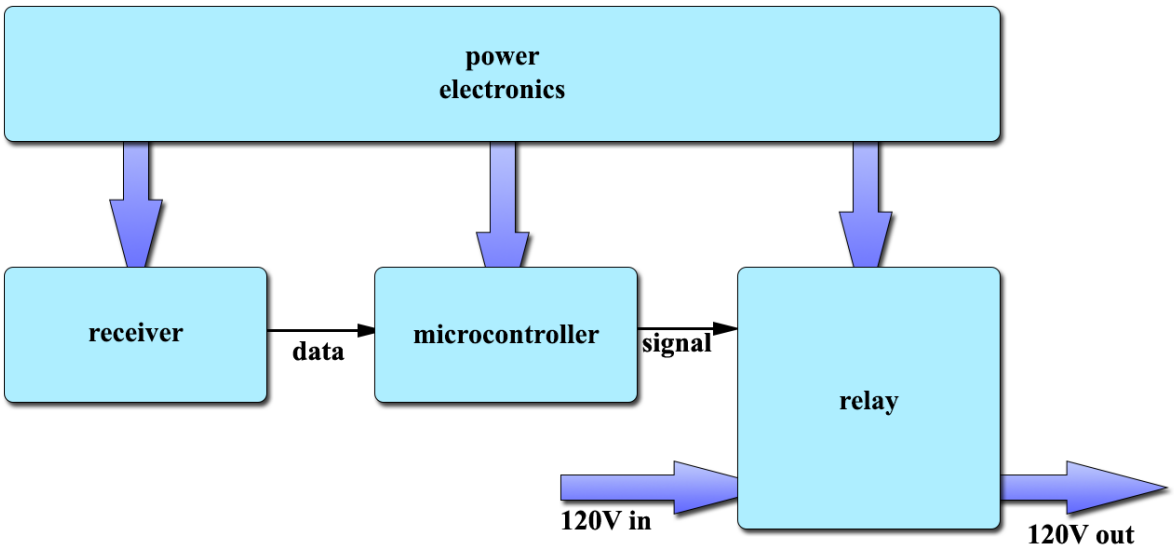
The power electronics circuit consists of an AC-DC converter with R and Cs to convert wall outlet AC to a steady and clean power supply of 5V.

1.1.4 Transmitter:

Transmitter receives the signal from microprocessor and send to remote unit wirelessly.

1.2 Remote Unit

Figure 2 shows the block diagram of Remote Unit.



Remote Unit

Figure 2 Remote unit block diagram

1.2.1 Receiver

Receiver wirelessly receives the signal send by the core unit.

1.2.2 Microcontroller

An inexpensive microcontroller ATmega32 is used to process the signal sent from core unit and send command to relay. Libraries are used to receive the data or implement the sampling, converting function by ourselves. After converting the data back to command, it sends signal to control the relay. A switch is added to enable and disable the voice control.

1.2.3 Relay

Relay receives the commands from microcontroller and switch the socket on and off.

2. Design

This section describes the design decisions, alternatives and details of each block.

2.1 Design procedure

2.1.1 Power electronics

The first design of power circuit is to build the whole power electronics system from scratch, including the transformer (inductors), rectifier, filter and regulator. From HSPICE simulation the ripple of the generated 5V DC is small and satisfactory. However the self-built inductor based transformer are more vulnerable to production defects and unexpected unreliable performance. Considering the priority of guarantee the highly steady power supply for sensitive parts such as transmitter and receiver, we finally decided to use an AC-DC converter and developed the power circuit based on this chip.

2.1.2 Microprocessor

Among the different microprocessors we choose Raspberry Pi because it has all the ports we need, including the connection with microphone, speaker, keyboard, Internet, and screen. In addition, with its Rasbian OS it can run the voice recognition software Jasper that we use. Besides Jasper, we write our own control program, together with Jasper to generate control signal based on voice recognition.

The software part could be helpful if the hardware design has any flaw. In our design we have two plans. The first one is to establish dual-direction communication. The core device sends commands to the remote and when received, the remote device send acknowledgement back to core. The second one is to send a command from core to remote repeatedly. In this way, the type of the command is limited to really basic ON and OFF but it guarantees that the remote will not miss the command in a noisy environment.

2.1.3 Transmitter and receiver

TXM-418-LC which allows dual-directional wireless communication is used for both transmitter and receiver. Alternative design is to use single direction transmitter and receiver chip correspondingly. However, considering the possibility of future developing of feedback from remote unit to core unit, we use the bi-directional chip.

2.1.4 Microcontroller

We choose ATmega32 to process the control signal from microprocessor because it is inexpensive and easy to program and test with Arduino shell.

2.1.5 Microphone

Performance of microphone affects the voice recognition performance, so we test several microphones after the voice recognition was finished and choose the microphone on the Logitech Webcam, which has the longest voice pick-up distance and clearness.

2.1.6 Relay

A relay board is used so that it can turn on and off multiple power sockets.

2.2 Design Details

2.2.1 Power electronics

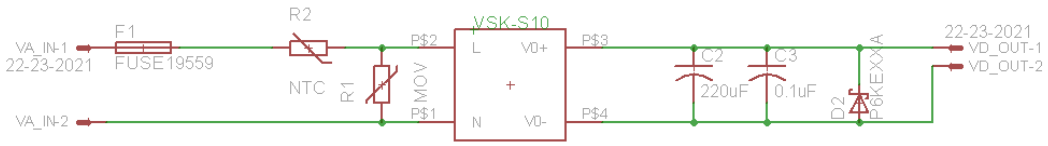


Figure 3 Power electronics schematic

Figure 3 shows the schematic of power electronics. As it mentioned in the datasheet of trm-418-lr [1], "It can be operated from a power supply as long as noise is less than 20mV. Power supply noise can affect the transmitter modulation; therefore, providing a clean power supply for the module should be a high priority during design." Our power supply has a noise level of 50 mv-p2p, so the input Vcc must be filtered. The datasheet of trm-418-lr presents us a simple filter and following is a simple calculation to see this filter satisfies the circuit.

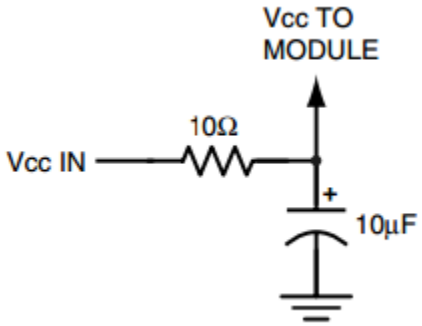


Figure 4 simple LPF circuit

Figure 4 is a simple LPF implemented a resistor (10 Ohm) and capacitor (10 µF, tantalum capacitor). So the frequency respond is

$$H(w) = \frac{1/\mathrm{jwc}}{\mathrm{R} + 1/\mathrm{jwc}}$$

Simplify the equation

$$H(w) = \frac{1}{1 + \mathrm{jw} \times 10^{-4}}$$

$$H(s) = \frac{1}{1 + \mathrm{s} \times 10^{-4}}$$

Since the datasheet doesn't mention the frequency range that may affect the modulation, it expect the noise far from the operating frequency (i.e. by degree of 100) should be within the limit. So what we want to achieve is, at $\mathrm{f} = 4\mathrm{MHz}$, the output of this circuit should have noise less than 20 mV.

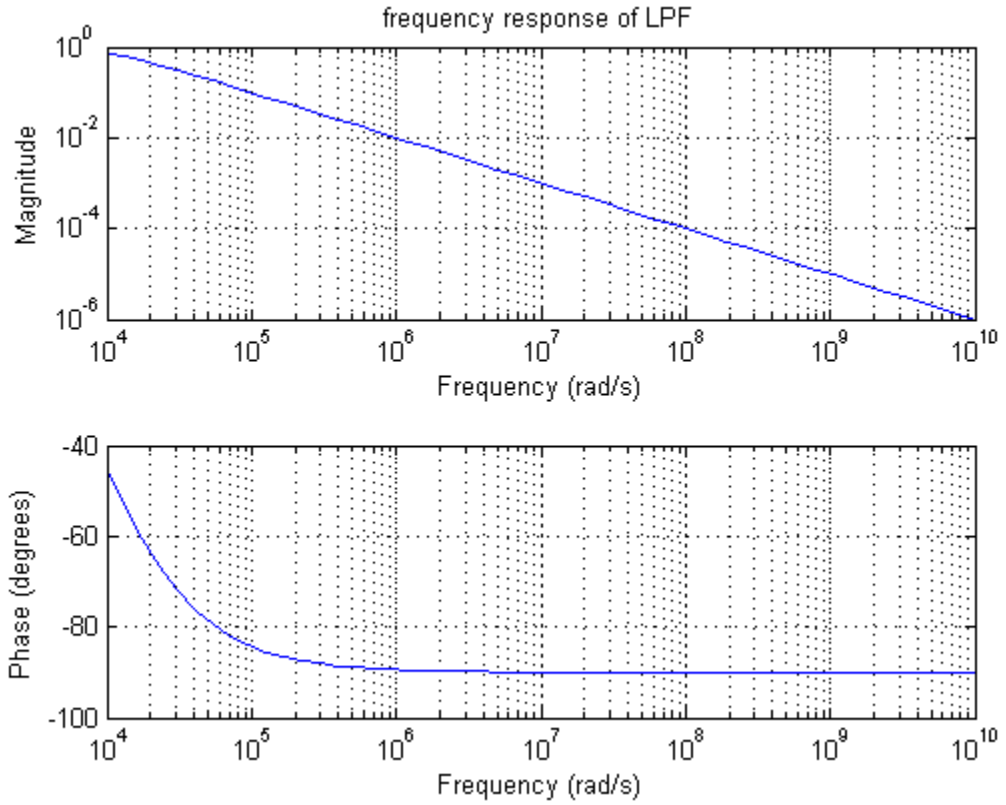


Figure 5 MATLAB simulated result

As we can see from figure 5, the magnitude of frequencies is around 10^{-2} which is far better than what is required.

To find the accurate result here is the equation

$$\frac{1}{1+\frac{1\mathrm{i}\times 400\times 10^{6}}{\pi\times 10^{-4}}}\times 50 = 0.003\ \mathrm{mV}$$

which is far better than the 20mV requirement.

2.2.2 Microprocessor

The Jasper library analyzes the voice input locally to see if it contains the keyword "jasper". If it contains "jasper", the voice input is sent to the online voice recognition engine and the corresponding text will be returned. My function searches the text to see if it contains the keywords like "Switch", the action verb like "turn on", "turn off", "toggle" and the remote ID that match the local records. If all requirement is

satisfied, my function will generate a 12 bits data that contains the pair information and specific command, encode it with hamming 11-15, insert the encoded data into a shell and send through the transmitter.

On the remote side, the program constantly checks the receiver's output and shifts the output into a buffer. If the buffer matches the pattern of the shell, the data inside the shell will be decoded. The program will check the pair information first. If the pair info match the records, the program will turn on/off specific switch base on the command.

From the implementation level, since our transmitter/receiver chip doesn't support USART (Universal Synchronous/Asynchronous Receiver/Transmitter), we design our own serial communication protocol. We use the PIGPIO library to set the GPIO pin 23 to high and low alternative to create digital signal output. The baud rate was set to 500 because of several test we made. With the baud rate set to a higher value, the signal become inaccurate, to a lower value, our chip seems to have trouble handle it. For encoding, we chose the hamming 11-15 so that we could correct 1 bit error and detect 2 bits error. The message we send to the remote unit contains the core id, remote id and command. The core id is a 4 bits data that generated from the serial number of CPU. The remote ID is also a 4 bits data that we assign to each microcontroller. The 3 bits left are command. With the bit distance set to 2, we have at most 4 different command available.

Check Appendix C for microprocessor software flow chart.

2.2.3 Transmitter and receiver

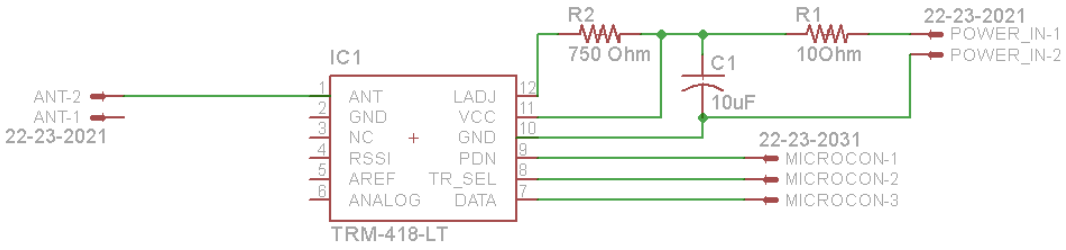


Figure 6 Transmitter and receiver schematic

Figure 6 shows the schematic of transmitter and receiver circuit. The TRM-418-LT chip we choose allow bi-directional wireless transfer of serial data. According to the datasheet, there are 3 pins that need to be connect to the microcontroller to achieve the functionalities we required. The pin PDN, TR_SEL and DATA. We are going to use Arduino & raspberry pi for the first stage implementation.

Depending on the software implementation, DATA might need to connect tx/rx pin of Arduino (and raspberry pi) or other digital pins for sending and receiving the digital data.

Pin TR_SEL can switch the transceiver between receiving and transmitting mode. When high, TRM-418-LT works in transmitting mode. This should be controlled by the microcontroller so that a hand shake could be established between the transmitter and receivers.

Pin PDN could turn the device into low power mode (basically turn the device off). A function could be implemented to turn the transmitter off in the core unit when not using to save some power.

2.2.4 Microcontroller

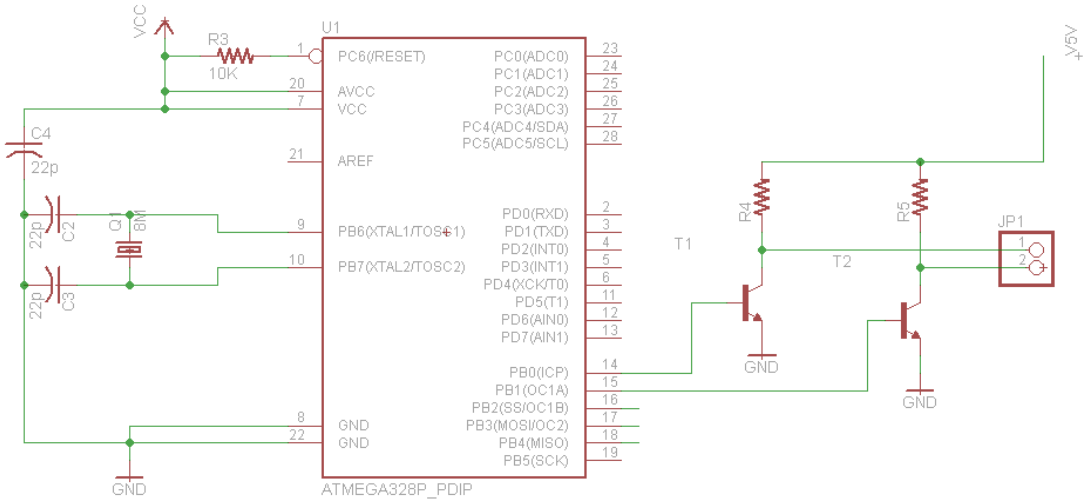


Figure 7 Microcontroller schematic

Figures 7 shows the schematic of Microcontroller circuit. Since the microcontroller ATmega32 is commonly used in Arduino, we took the circuit of Arduino [2] as reference and design the circuit of ATmega32. Due to the voltage difference of relay and microcontroller, transistors are used at output of microcontroller to control the relay.

Check Appendix B for full circuit of remote unit, including power electronics, receiver and microcontroller.

Check Appendix D for microcontroller software flow chart.

3. Design Verification

This section discuss the original requirement and verification plan and the actual testing of main blocks and completed project. Check Appendix A for full requirement and verification table from design review.

3.1 Power electronics

The requirement of power electronics is to transform Wall outlet 110V AC to 5 V DC ± 0.02V. The verification is to connect the power electronics circuit to wall outlet and use volt-meter to measure the output voltage. We got the result of a steady 4.99V for both the breadboard test and finished PCB test.

3.2 Transmitter and receiver

The requirement of transmitter and receiver is stable data transmission from transmitter of core unit to receiver of remote unit. To verify the data transmission, we input a signal to transmitter using function generator and make transmitter transmits the signal. Connect oscilloscope to output of receiver to verify the received signal. We passed the test in both breadboard and completed PCB stage. From figure 8 we can see that the received signal voltage is lower than the transmitted one, but the test still passed since the voltage is high enough for logical high.

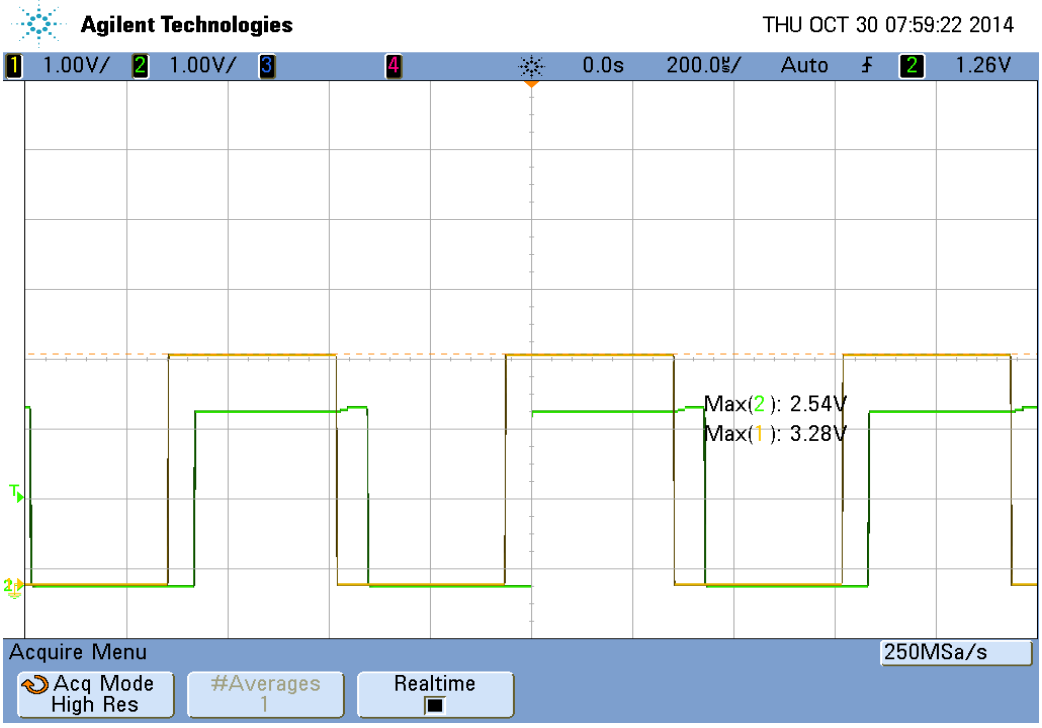


Figure 8 Transmitter and receiver test result. Yellow line is the signal generated by function generator; green line is the output of the transmitter

In design review, we designed lower level test for transmitter and receiver for debugging purpose, including Measurement of the antenna voltage to verify the transmitting and receiving of signal being, and measurement of frequency and gain of the signal, which should match the corresponding channels. Since the module passed the test at higher level, there was no need to perform the lower level tests.

3.3 Microcontroller

The requirement of microcontroller is to converts data received back to command and sends high or low signals to relay based on command. To verify that, we connected microprocessor and microcontroller and configured the microprocessor to send commands to microcontroller, verify the output of microcontroller to be high or low correspondingly. At breadboard stage, since the transmitter and receiver had been verified, we did the test with microprocessor and microcontroller linked by wireless communication and the test passed. Figure 9 is the test result.

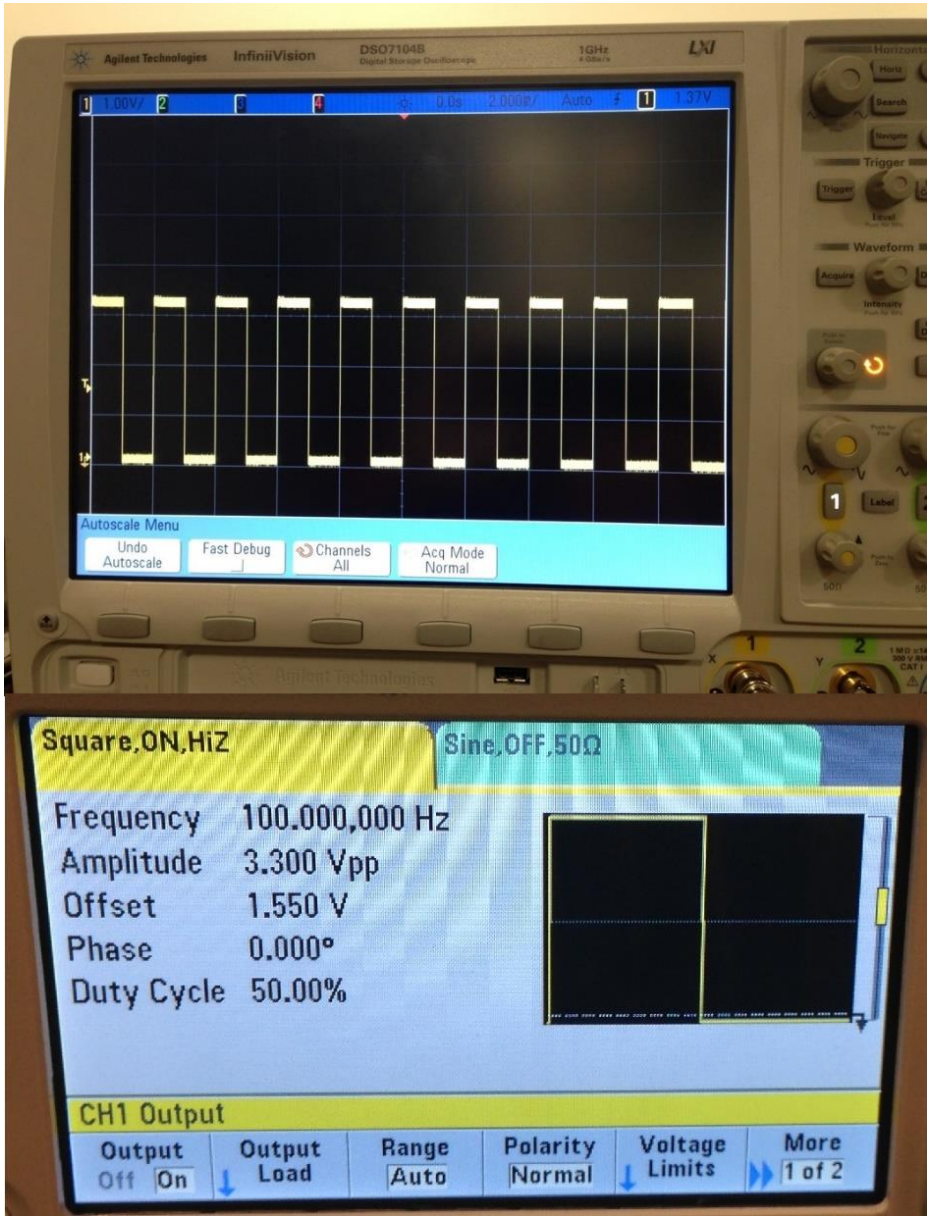


Figure 9 Test result of microcontroller. Photo below is generated 3.3V square wave signal; photo above is the output of the microcontroller.

3.4 Microprocessor

The first requirement of microprocessor is to recognize the voice command. To verify that microprocessor can recognize voice commands, set up the microprocessor with microphone, screen and Internet. Speak a sentence and make the recognized text be displayed on the screen. Verify the displayed text returned by online voice recognition engine is identical to what we said. We did 10 tests and 7of them passed. Factors that lead to failed tests include unclear voice pronunciation, too far away from microphone and noise.

The second requirement of microprocessor is to send data to transmitter, thus communicate with remote strip unit. To verify that, we connect the microprocessor directly to Arduino to test if the microprocessor can send correct data. We sent 10 sets of different data, and all the tests passed.

3.5 Completed project

To goal of the completed project is to turn on and off power socket based on voice command. To verify the completed project, we plugged in a lamp to each of the power socket. Then we said commands such as “Jasper, turn on switch 1.” “Jasper, turn off switch 2” and see if the system responds correctly.

We did 10 tests at home environment, 6 of them passed. Factors leaded to failed tests include voice pronunciation which affects command recognition, speaker’s distance to device which affect the performance of microphone, speaking pace which must fit the set-up voice capture timing, and distance of core and remote unit which affects the wireless communication.

We also had user test at demo in lab, which is a noisy environment. 4 of 10 test passed. Besides user’s several tries to manage the speaking pace, the noise decreases the recognition successful rate and response speed of the system. Speaking directly to the microphone can solve the issue and achieve a much higher successful rate.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Quantity	Retail Cost ($)	Actual Cost ($)
MMUN2211LT3GOSCT-ND	ON Semiconductor	4	0.14	0.56
102-2392-ND	CUI Inc	2	20.13	40.26
MOV-14D361K-ND	Bourns Inc.	4	0.21	0.84
F3297-ND	Littelfuse Inc	4	1.42	5.68
317-1158-ND	Cantherm	4	0.4	1.6
BC2665CT-ND	Vishay BC Components	4	0.37	1.48
P6KE6.8CA-TPMSCT-ND	Micro Commercial Co	2	0.47	0.94
1189-1518-1-ND	Rubycon	4	0.3	1.2
MCP1700T3302ETTCT-ND	Microchip Technology	2	0.44	0.88
MCP1703T-3302E/CBCT-ND	Microchip Technology	2	0.65	1.3
445-8517-ND	TDK Corporation	10	0.278	2.78
CanaKit Raspberry Pi B+ Comlete Starter Kit with WiFi	Cana Kit	1	62.95	62.95
Starter Kit for Newsite Uno R3	ARDUINO	1	33.99	33.99
SunFounder 2 Channel 5V Relay Shield Module	SunFounder	1	15.78	15.78
Gold plated Hi-Speed USB 2.0 Cable 3 Feet	Cable Matters	1	12.98	12.98
Total				183.22

Table 1 show the Parts used in the project and the costs.

4.2 Labor

Table 2 Labor Costs

Employee	Hourly Rate ($)	Worked Hour	Multiplier	Cost ($)
Zhe Gong	35	150	2.5	13125
Hongchuan Li	35	150	2.5	13125
Total				26250

Table 2 show the labor cost of the project.

5. Conclusion

5.1 Accomplishments

We built a voice controlled power socket system, which can control multiple power sockets by corresponding voice commands. The original goal is achieved. The product has the strength that it can control multiple sockets, and it is possible to expand new sockets.

5.2 Uncertainties

Uncertainties exist in the voice recognition and wireless communication aspects of our product.

Noise affects the voice recognition accuracy, response distance, and response speed. Using of more powerful microphone can decrease the susceptibility due to noise. Developing noise filter or noise handling algorithm may further improve the performance.

Wireless communication is limited by the effective communication distance of the transmitter and receiver, especially in the situation of a large house. Using more powerful wireless module can improve the performance.

5.3 Ethical considerations

As a product of power system, we designed the insulation of high power carefully to guarantee the safety of both developer and users. In addition our product will reduce the time people need to touch power socket, thus reduces the possibility of electric shock accidents. As a result, the project is consistent with the first code of the IEEE Code of Ethics [3]:

> *1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;*

Through the developing of the project, we learn and cite all the technology works we used, and keep documentation of our project for future development and reference, which is consistent with the fifth and sixth code of the IEEE Code of Ethics:

> *5. To improve the understanding of technology; its appropriate application, and potential consequences;*
>
> *6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;*

For various peer reviews, design review and presentation events of this projects, we accepted peers and instructors' advices and criticism of technical work and used the advice to improve our project, which is consistent with the seventh code of the IEEE Code of Ethics:

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others

5.4 Future work

Besides the development of noise handling and testing of more powerful microphone and wireless module mentioned above, another important future work is to add microphone in remote unit. As a result, when user and socket are both in a room far away from the core unit, the remote microphone can capture user's voice command and send it back to core unit. Core unit can then send control signal to remote unit to turn on and off the remote socket. Now our product only allows user to speak voice command in the same room of the core unit.

References

[1] *TXM-418-LC*, datasheet, Linx Technologies, 2014. Available at:
http://www.linxtechnologies.com/resources/data-guides/txm-xxx-lc.pdf

[2] *Arduino Uno R3 DIP Edition (Revision 3)*, schematic, Arduino, Available at:
http://www.jameco.com/Jameco/Products/ProdDS/2151486%20schematic.pdf

[3] IEEE Code of Ethics, web page. Available at:
http://www.ieee.org/about/corporate/governance/p7-_8.html/.Accessed October 2014.

Appendix A Requirement and Verification Table

Table 3 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Power Electronics 1. Wall outlet transform to 5 VDC ± 0.02V	1. Connect the power electronics circuit to wall outlet. Use volt-meter and oscilloscope to measure the output voltage. Verify the voltage is 5 VDC ± 0.02V.	1. Y
Microprocessor 1. Microprocessor can recognize the voice command. 2. Microprocessor can send data to transmitter, thus communicate with remote strip unit.	1. Set up the voice-recognizing program, internet and microphone connection with microprocessor. Say a voice command to microphone and verify that the text printed out by microprocessor is correct. Do the test 10 times. It should pass at least 9 of them. 2. Set up the transmitter and receiver with microprocessor. Write a test program that sends a 30 bits binary signal. Record the received signals from receiver and verify the signal is identical. Do the test 10 times. It should pass all the tests	1. N 2. Y
Microphone 1. Microphone captures and send human voice signal to microprocessor correctly.	1. Connect two microphones to the microprocessor. Say a voice command and check the two signals microprocessor get. Signals from the two microphone should be identical to the voice command and to each other. Do the test 10 times. It should all pass.	1. Y
Transmitter 1. Transmitter can send data on various channels.	1. Set transmitter to send signals on different channels. Measure the antenna voltage to verify the signal being transmitting. Measure the frequency and gain of the signal, which should match the corresponding channels. Test 5 channels. All of them should pass.	1. Y
Receiver 1. Receiver can receive data from transmitter	1. Make transmitter transmits a signal, set the receiver on the corresponding channel. Use volt-meter to measure the signal pin voltage to verify the voltage being outputted. Connect oscilloscope to receiver to show the signal. The displayed signal and the transmitted signal should be identical. Test 10 different signals,	1. Y

	should all pass.	
Microcontroller 1. Microcontroller converts data received back to command 2. Based on command, microcontroller sends high or low signals to relay	1. Connect microprocessor directly with microcontroller, skipping the wireless communication. Configure the microprocessor to send commands to microcontroller, verify the output of microcontroller to be high or low correspondingly. Test 10 commands, should all pass.	1. Y
Relay 1. Turn on and off based on input signal high or low.	1. Connect relay with function generator. Check the connectivity (on and off) when the input from function generator is high or low.	1. Y

Table 3 shows the requirement and verification plan from the design review and whether the verification was met in the end.

Appendix B Schematic of Remote Unit

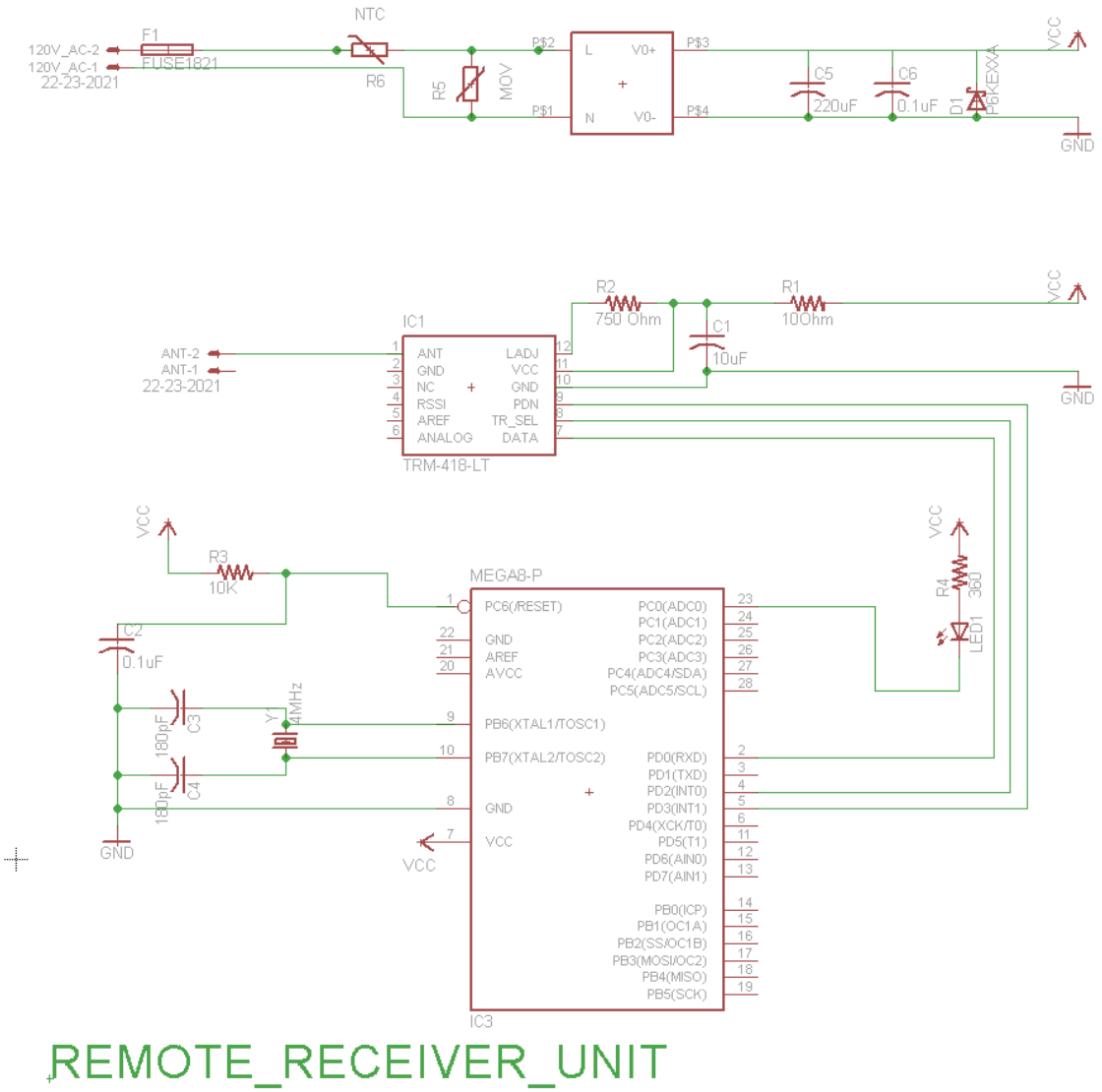


Figure 10 Remote unit schematic

Figure 10 shows the schematic of the whole remote unit.

Appendix C Microprocessor software flow chart

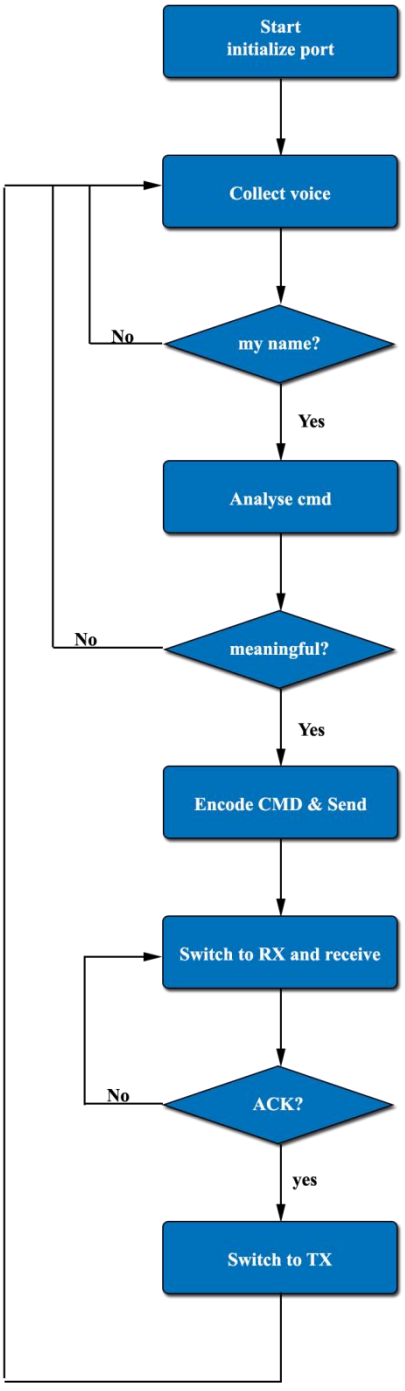


Figure 11 Microcontroller software flow chart

Figure 11 shows flow chart of our program of microcontroller.

Appendix D Microcontroller software flow chart

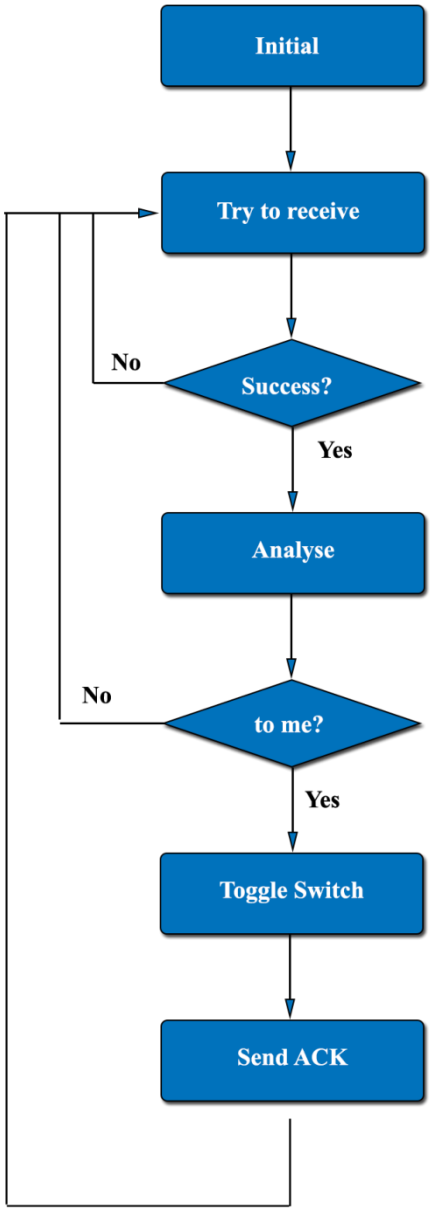


Figure 12 Microcontroller software flow chart

Figure 12 show flow chart of our program of microcontroller.