# Hand-Motion-Controlled Car

ECE 445 Design Document - Spring 2018

Team 52 - Tianyang Zheng, Bohan Hu, Jingyu Li
TA: Hershel Rege

# Contents

# 1. Introduction

## 1.1 Motivation

Nowadays, remote controlled moving devices are widely used in a number of applications. Remote controlled electrical toy cars/ships create lots of fun for kids and enthusiasts. Drones are able to capture amazing aerial photos as well as to assist in scientific explorations. Remote controlled mine detection vehicles help reduce casualties caused by land mines. Disaster rescue robots help save people's lives in areas where it's too dangerous for rescue teams.

Currently there are two common methods for human operators to control the movement of mobile vehicles/devices. The first method is to use a handheld remote controller with buttons and/or joysticks. The second method is to develop an app that communicates with the mobile vehicles and apply controls via the user interface such as touchscreen buttons and bars. While these two methods are effective and accurate, there are also some limitations. First, both methods are indirect and not very intuitive. Users insert input controls to the remote controllers, so they lack a sense of directly controlling their mobile devices. Second, both methods require users to carry a control device around. Users' arms and hands are likely to become tired after some time. This may reduce the efficiency as well as control accuracy. Moreover, normally both hands of a user are occupied with the controller during the control process. This makes the user unable to simultaneously perform other tasks which may assist the control process. Our team came up with a different control method that can improve upon those limitations while having the similar control performance.

## 1.2 Background

There are a large number of commercial remote controllers in the market. Some top brands of remotes are Logitech, Sideclick, and Vizio. Their latest remotes consist of buttons, small touchscreens, and joysticks in cases of toy cars and drones. With those remotes, users can conduct a large variety of control patterns. However, there's still no control mechanism that utilizes part of the user's body to perform direct controls in the market. As a result, even though we will only design a glove that does a very limited range of control patterns, this control mechanism is innovative and helpful. Moreover, it's not hard to add in more control functionalities once the whole system works.

## 1.3 Functionality

We aim to create a special glove attached with electrical components. Once a user wears the glove on one hand, he/she can control the movement of the mobile vehicle by different hand gestures. We create the following gestures:

1. Rotation of the hand around its vertical position to make the mobile vehicle turn left or right. In addition, the sharpness of the turn is proportional to the angle of rotation of the hand.
2. A flattened hand to make the mobile vehicle drive at full speed.

3. Curl of the fingers to decrease the speed of the vehicle proportionally.
4. A fist to make the vehicle stop completely.

Wireless communication is used between the glove and the mobile vehicle. We will also implement an additional feature for the mobile vehicle. When the vehicle gets too close to any object, it will automatically stop driving until the obstacle is no longer at its proximity. This feature helps avoid collisions between the vehicle and other objects.

## 1.4 High-Level Requirements

- Car must response fastly and accurately to every given gesture. When the user gives command through the glove, the car will be able to respond within 1 second.
- Glove must be relatively comfortable to wear and make gestures in it. The size of the glove will be between 8.5~9 inches. With all the electrical components installed on the glove, it should still fit the size of most people's hand.
- Communication between car and glove must be easy to set up while using.

## 1.5 Block Diagram

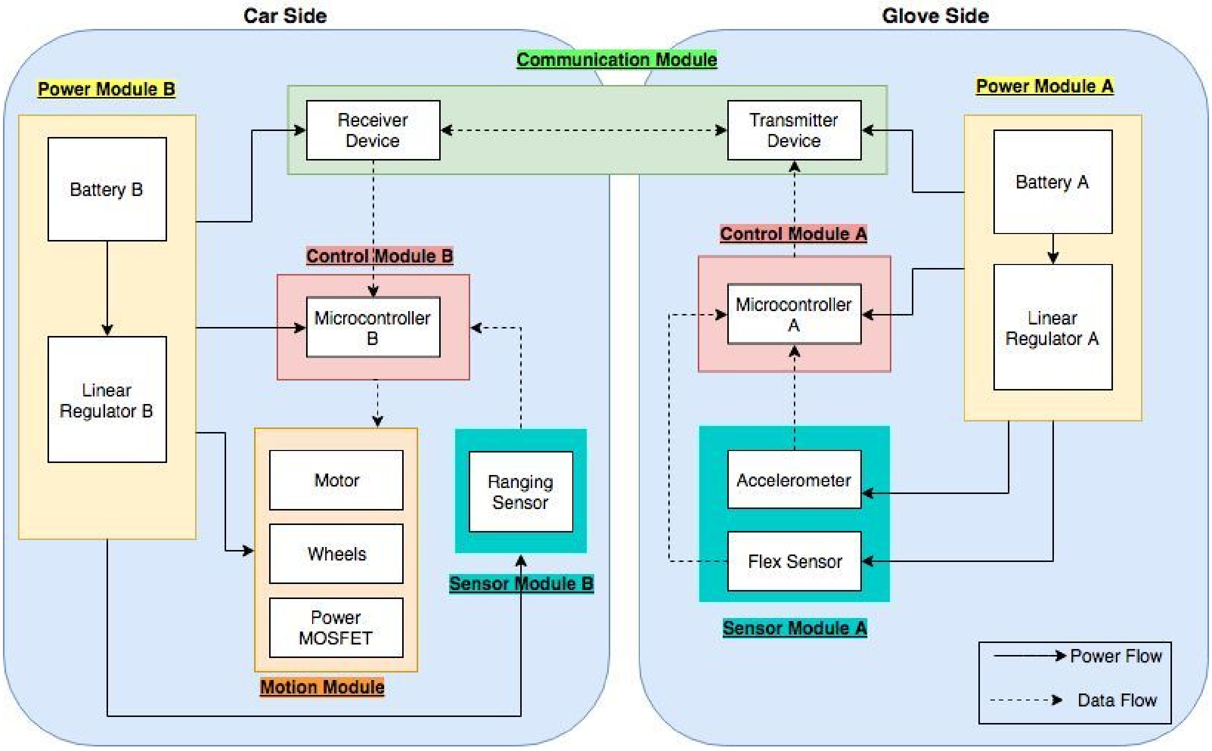


Figure 1. Block Diagram

The block diagram is divided into two main parts, the glove side and the car side since they are relatively isolated besides the wireless data communication between them. Moreover, the blocks are grouped into several modules in the sense that all components in a module work together to achieve the same general functionalities. These modules reflect the division of work for this entire project.

# 2. Design

## 2.1 Design Procedure

On the glove side, first is the power module A. There are two blocks in it: battery and linear regulators. For the battery block, we chose the AmazonBasics 9V Alkaline Batteries. There are many other voltage levels available, but 9V is the lowest level that satisfies the input voltage requirements of the linear regulators we chose. Moreover, 9V batteries are able to provide enough power for a reasonable amount of time after thorough calculation. The portable size of 9V batteries is another advantage. Only one such battery is needed for the glove side. For the linear regulator block, both 5V and 3.3V DC voltage are needed by certain components on the glove. We chose L7805CV 5V regulator and TC1262-3.3VAB 3.3V regulator. They are able to convert 9V battery voltage to the desired level.

Second is the control module A. This module consists of only one block, which is microcontroller A. A microcontroller is used to process all the data, coordinate different components, and thus control the components on the glove. We chose ATMega328P microcontroller since it can be easily programmed by Arduino codes and its pins are able to provide all the functionalities our project needs. In order to properly set up the pin functionalities of an ATMega328P chip, a circuit is needed. Figure 2 below is the circuit schematic of our microcontroller circuit.

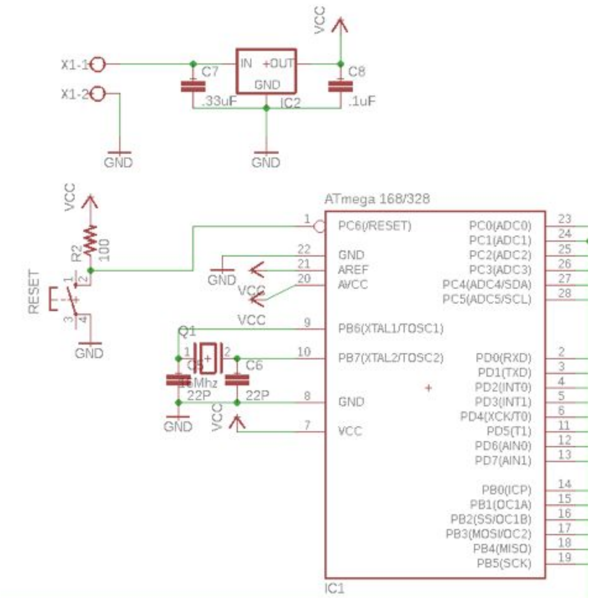


Figure 2. Microcontroller Circuit Schematic

Third is the sensor module A. It includes two blocks: flex sensor block and accelerometer block. For the flex sensor block, we used flex sensors to monitor the curls of fingers by the user. Three flex sensors are attached to the three fingers of the glove so that the sensors are bent proportionally to the user's fingers. We chose Sparkfun Flex Sensor since it is the only one available that is manufactured by a big-scale saler. The way this block sends data to the microcontroller block is as following: a flex sensor is connected in series with a 47kΩ resistor between a 5V voltage level and ground. The voltage at the point between the sensor and the resistor is fed into the ADC pin of the microcontroller. For the accelerometer block, a 3-axis accelerometer module is used to monitor the inclination of the user's hand. The accelerometer is fixed on the glove so it tilts the same way as the hand. We chose ADXL335 3-axis accelerometer since its 3-axis sensing enables us to attach the sensor any way we want on the glove. The variable analog voltage output of the sensor is fed into the microcontroller ADC pins for further processing.

Fourth is the transmitter device block. We chose a pair of NRF24L01 2.4GHz transceiver modules as the wireless data transmission devices. They can provide reliable data transmission over 100 meters. Moreover, they are easy to set up with a customized Arduino library. Another advantage is the small size which saves space on the glove. One transceiver acts as the transmitter on the glove side. It is controlled by the microcontroller to send data packages to the receiver on the car side.

On the car side, first is the power module B. It consists of a battery block and a voltage regulator block. For the battery block, the same AmazonBasics 9V batteries are used. Here, two batteries are used. One battery provides power only to the two motors, and the other powers all other components. We came up with this design since according to calculation, two motors need a large current draw which can not be sufficiently provided by a single battery together with all other components. A battery dedicated to the two motors is the solution. For the voltage regulator block, it is the same as the one on the glove side. Same parts are used to create 5V and 3.3V DC voltage.

Second is the control module B. It consists of an ATMega328P chip. It serves the same purpose as the control module A on the glove side, which is to control and coordinate the functioning of all the components on the car.

Third is the sensor module B. It includes just a single ranging sensor block. A ranging sensor is used to detect if there's any obstacle within a certain range in front of the car. An automatic protective stop will happen if the sensor detects anything that's too close to the car. We chose a HC-SR04 Ultrasonic Sensor since it is widely used by similar projects. It sends its sensing data to the microcontroller for distance calculation.

Fourth is the transceiver block. It uses the same transceiver module as the one on the glove. Here, the transceiver acts as the receiver that receives the data from the glove side and sends it to the microcontroller for processing.

Finally, there is a motion module on the car side. It is in charge of making the car move in a desired way commanded by the microcontroller. We used PWM control to control the spinning speed of the motors. Both the turning and speed change of the car are achieved by the proper control of the speed of each rear wheel. This module consists of a motor block, a power MOSFET block, and a wheel block. For the wheel block, it represents the mechanical components of the car. We purchased the Sparkfun Circular Robotics Chassis Kit (Three-Layer) and assembled the car ourselves. It has three layers of small-sized aluminum plates and drives on two rear wheels and one caster wheel at front. The main reason we chose this robotic car is that it has small size and light weight so that it provides better controllability for the user. For the motor block, we chose a pair of 200RPM Hobby Gearmotor. They fit perfectly with the car we chose and can be powered by 9V DC voltage. Moreover, they are small in size while still providing enough maximum spinning speed for our project. Finally, for the power MOSFET block, we designed a MOSFET circuit (see Figure 3) to control the 9V motor circuit using the 5V PWM control signal from the microcontroller. There are two main reasons for this design. First, after some testings we found out that 5V PWM signal from the microcontroller PWM pin can not be directly fed into the motor. Second, it is safer to control a higher voltage circuit using a lower voltage one.

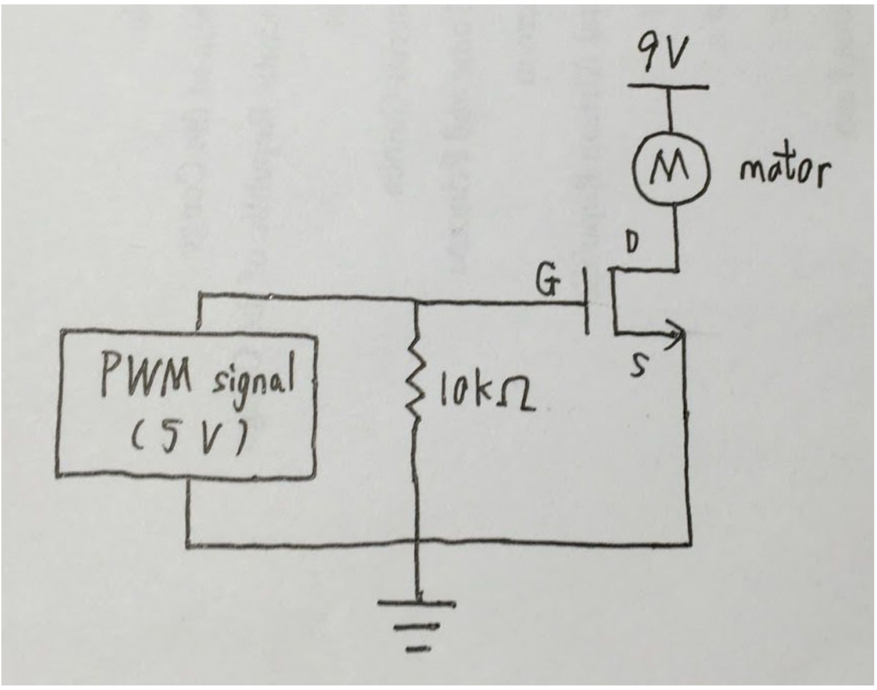


Figure 3. Power MOSFET Circuit

## 2.2 Design Details

### 2.2.1 Battery A

One AmazonBasics 9V battery only weighs 0.8 ounces, with a dimension of 2.9x1.8x0.5 inches. Adding up the power consumption and current flow of each individual component on the glove, the maximum total current will be 18.97mA and the maximum power will be 67.43mW. Therefore, one battery will be sufficient to provide enough power for 3-4 hours.

### 2.2.2 Linear Voltage Regulator A

The STMicroelectronics L7805CV 5V voltage regulator takes a wide range of voltage input from 7V to 35V. Its output current is 1A. It can operate normally under up to 125C environment. The TC1262-3.3VAB 3.3V regulator has an input voltage range of 2.7-6V. In our design, it takes in the output of the 5V regulator and produces a 3.3V output. On the glove, the microcontroller, the accelerometer, and the flex sensor circuit need a 5V voltage. The transceiver module needs a 3.3V voltage.
Power Consumption for both voltage regulators:
The operating voltage is 4.8-5.2V, and the current is 5mA-1A. Therefore, the power will be 0.024-5.2W.

### 2.2.3 Flex sensor

A Sparkfun Flex Sensor, essentially a variable resistance from 30kOhm to 130kOhm, must be able to control the range of voltage going into the microcontroller ADC pin to be between 1.389V to 3.125V. Below is a circuit diagram of feeding the voltage related to the bend of a flex sensor into a microcontroller (denoted as MP in the diagram).

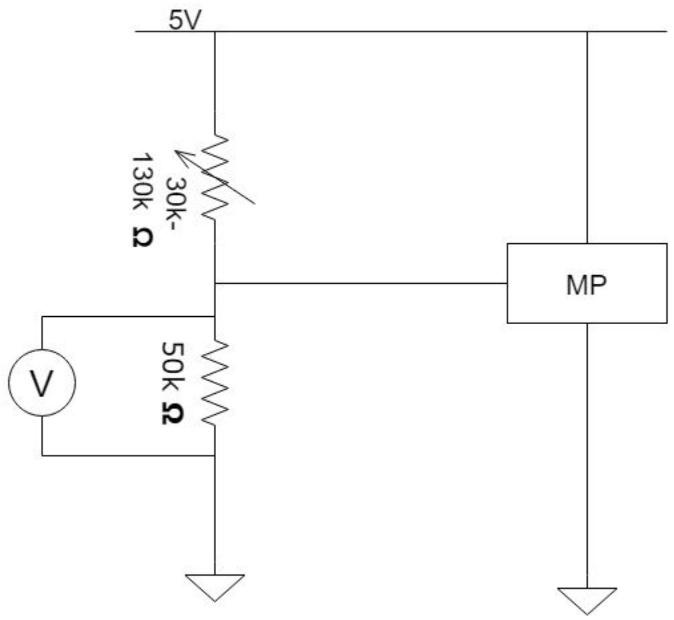


Figure 4. Circuit for Feeding Flex Sensor Measurement into a Microcontroller

### 2.2.4 Accelerometer

ADXL 335 3-Axis Accelerometer has ultra low power consumption (350 µA typically) and very small size. The range of power supply is 1.8-3.6V. The accelerometer has three analog output pins that will be connected to the analog input pins of the ATmega328 chip. This is how the sensor data will be transferred to the microcontroller for further processing. The three analog outputs of the accelerometer-denoted by Ax,out , Ay,out , Az,out-can be converted to three tilt angles (angles are with respect to the three sensing axes) using the formulas below:

$$\theta = \tan^{-1}\left(\frac{A_{X,OUT}}{\sqrt{A^2{}_{Y,OUT} + A^2{}_{Z,OUT}}}\right)$$ (Equation 1)

$$\psi = \tan^{-1}\left(\frac{A_{Y,OUT}}{\sqrt{A^2{}_{X,OUT} + A^2{}_{Z,OUT}}}\right)$$ (Equation 2)

$$\phi = \tan^{-1}\left(\frac{\sqrt{A^2{}_{X,OUT} + A^2{}_{Y,OUT}}}{A_{Z,OUT}}\right)$$ (Equation 3)

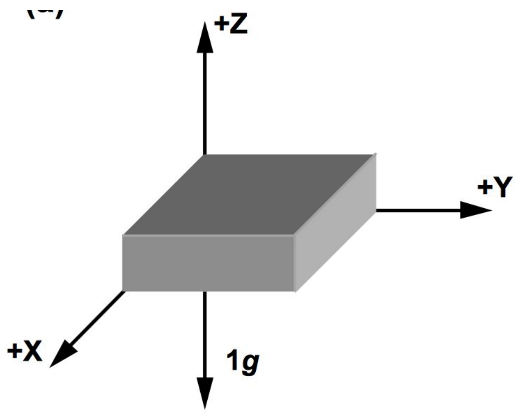


Figure 5. Accelerometer on a flat surface

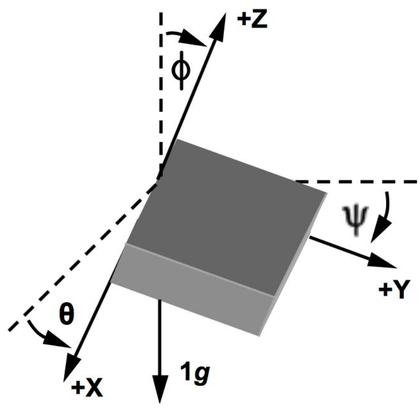


Figure 6. Accelerometer at anTilted Orientation

Power consumption:
typical operating power: 350 uA at 3V power supply (1.05mW), maximum power: 375 uA at 3.6 V(1.35mW), minimum: 200 uA at 2V (0.4mW)

### 2.2.5 Microcontroller A

The SPI serial port on Atmega328 will handle the SPI communication with the transmitter device. The 6-channel 10-bit A/D converter will be able to convert the analog inputs from the sensor modules to digital signals. After bootloading the Atmega328 chip, we will build an Arduino circuit which enable us to program the microcontroller through Arduino Software (IDE) and send the program to the memory on the Atmega328 via USB cable.

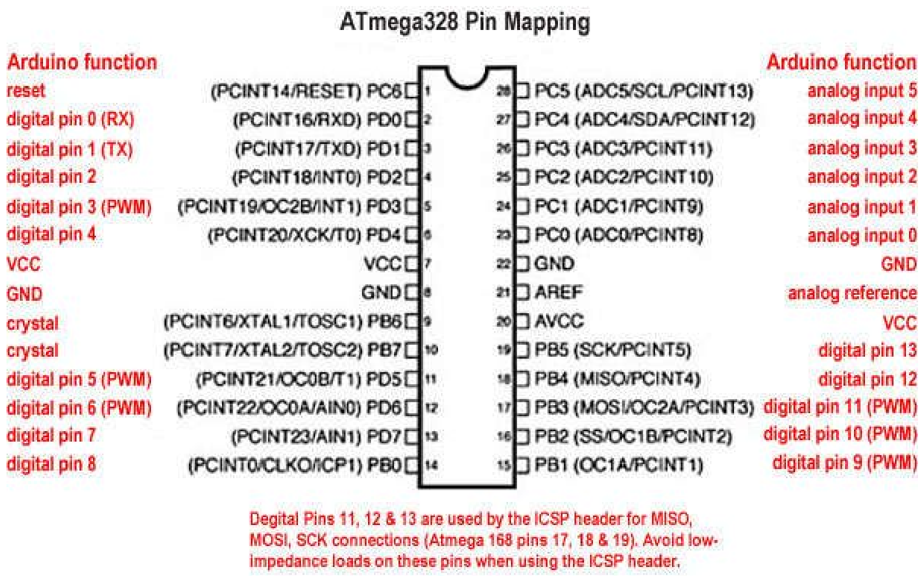


Figure 7. Pin map between Atmega328P and Arduino

Power Consumption:
If an 8MHz (internal) oscillator is used, the operating voltage is 3.43V and the current is 3.6mA. Therefore, the power will be 12.3mV.

If an 8MHz (internal) oscillator is used, the operating voltage is 3.3V and the current is 6.6mA. Therefore, the power will be 21.8mV.

### 2.2.6 Transceiver Module

The transceiver interfaces with the microcontroller ATmega328 through the digital SPI communication. The maximum SPI speed is 8 Mbps and the in-the-air data rate is 1 or 2 Mbps, both of which are fast enough. The range of operation is up to 100 m, which is enough for our project. During normal operation, the microcontroller will continuously send real-time data to the transceiver for it to transmit to the receiver on the car. There is an RF24 library associated with the transceiver available online, which will make the programming of the wireless data transfer less difficult. The operating voltage of NRF24L01 is 1.9-3.6 V. The current consumption is very low, only 9.0mA at an output power of -6dBm and 11.3mA at an output power of 0 dBm. Another advantage of this transceiver is its small physical size.

Power consumption:
The operating voltage is 1.9-3.6 V. The supply current for one channel at 1000 kps is 11.8mA, and at 2000 kps is 12.3 mA. Thus the power consumption range is 22.42-44.28 mW.

### 2.2.7 Ultrasonic Sensor

HC-SR04 Ultrasonic Sensor can provide stable and accurate distance measurements from 2cm to 400cm. The ranging accuracy is up to 3mm. The working frequency is 40Hz. The measuring angle is 15 degrees. All these specs suit our collision detection task. The sensor module will be connected with the pins of the microcontroller B and send distance measurements to the microcontroller for further usage.

Power consumption:
The operating voltage is 5V and the rated current is 15mA. Therefore, the power is 75 mW.

**2.2.8 Power MOSFET (RFP12N10L)**

RFP12N10L n-MOSFET is used for the PWM control circuit. At the Gate, we connect the PWM from microcontroller in series with a 10 kΩ resistor to the ground. The motor is connected between the 9V battery and the Drain. The Source is connected to the ground. Below is the circuit schematic of the power MOSFET circuit.

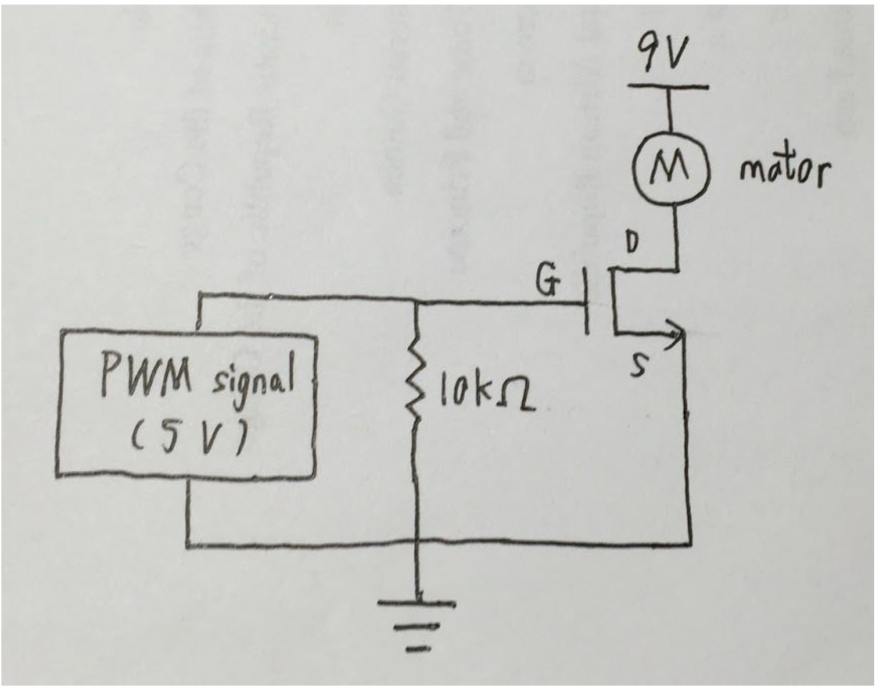


Figure 8. Power MOSFET Circuit

**2.2.9 Eagle Schematics for the PCBs**

We designed two PCBs, one for the circuit on the glove side and the other for the car side. Below are the Eagle circuit schematics for the two PCBs.

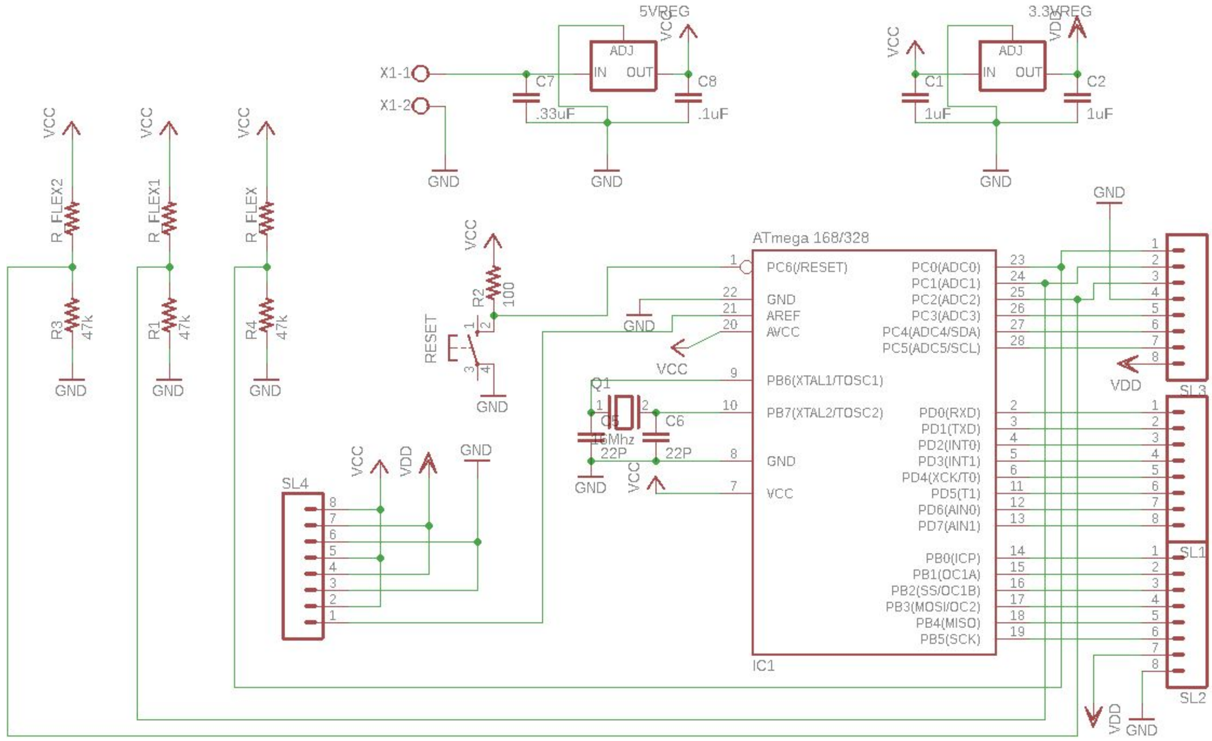


Figure 9. Eagle Schematic of the PCB on Glove Side

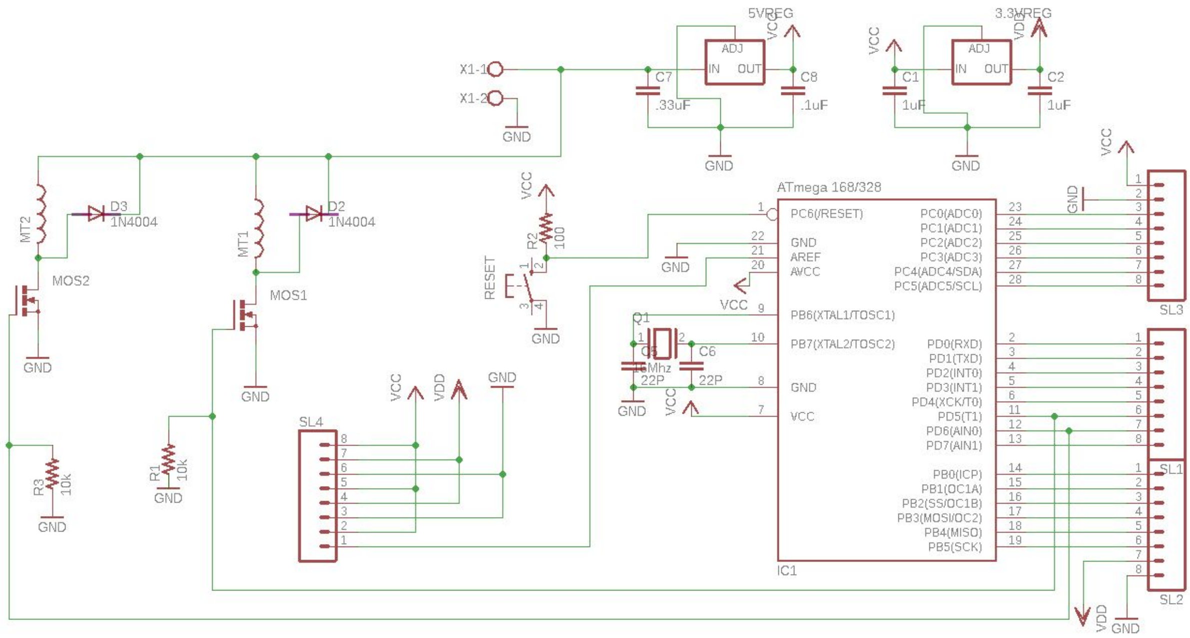


Figure 10. Eagle Schematic of the PCB on Car Side

### 2.2.10 Car's Control Unit Algorithm Flowchart

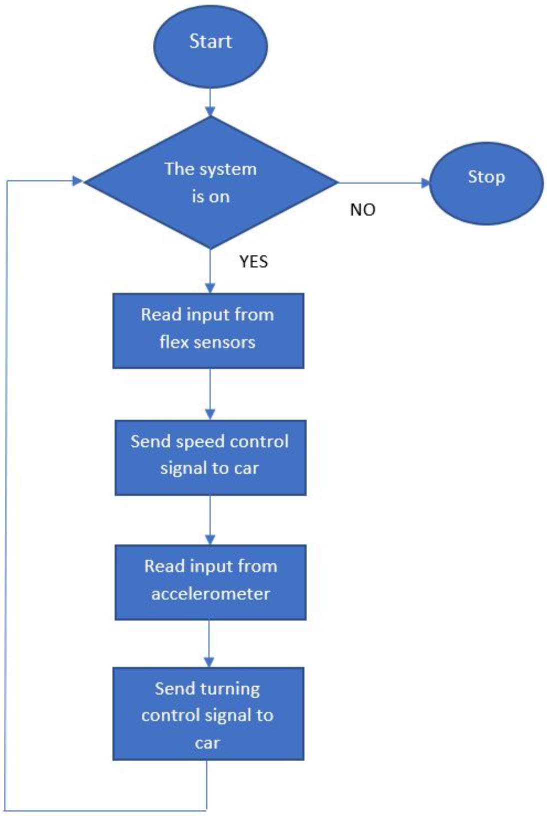


Figure 11. Algorithm of the control unit on the car

# 3 Verification

## 3.1 Flex Sensor

We built the flex sensor circuit and used Arduino board to read in the analog data for several bending degrees for each of the three flex sensors we have. Below is a table of the bending degree-vs-ADC readings.

| | ADC Readings | | |
|---|---|---|---|
| Bending Degree | Flex Sensor A | Flex Sensor B | Flex Sensor C |
| 0 degrees | 600 | 630 | 618 |
| 30 degrees | 470 | 470 | 466 |
| 60 degrees | 400 | 408 | 398 |
| 90 degrees | 350 | 356 | 354 |

Table 1. Bending Degree vs ADC Readings

### 3.2 Accelerometer

We connected the accelerometer module to an Arduino board, displayed the ADC readings of the three axes to the Serial Monitor, and recorded the ADC readings for numerous tilting degrees of each axis. Below is a table of tilt angle-vs-digital reading data for all three axes (0° is when the axis is perpendicular to the earth gravity, +90° is when the positive axis aligns with earth gravity, -90° is when the negative axis aligns with earth gravity):

| | X axis | Y axis | Z axis |
|---|---|---|---|
| 90° | 264 | 263 | 340 |
| 60° | 271 | 277 | 352 |
| 30° | 292 | 292 | 373 |
| 0° | 331 | 326 | 388 |
| -30° | 368 | 364 | 392 |
| -60° | 387 | 379 | 400 |
| -90° | 409 | 393 | 412 |

Table 2. Tilt Angles vs Digital Reading Data

### 3.3 Ultrasonic Sensor

We connected the ultrasonic sensor to an Arduino Uno board, continuously read in the sensing data and displayed it on the Serial Monitor. I used my hand as the object in front of the sensor and tested the sensing accuracy for various distances. Below is a table of the actual distance-vs-sensor output.

| Actual(cm) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sensed(cm) | 3 | 2 | 2 | 3 | 4 | 6 | 7 | 7 | 8 | 10 | 11 | 12 | 13 | 13 | 15 | 16 | 17 | 17 | 18 | 20 |

Table 3. Actual Distance vs Sensor Output

## 4 Costs

In this section, we will split our costs into two parts, component cost and labor cost. Labor cost will be based on our estimation of how long it will take for one technician to assemble one product.

### 4.1 Component Cost

| **Part name** | **Manufacturer** | **Quatity** | **Part Cost($)** |
|---|---|---|---|
| 5V Linear Regulator | STMicroelectronics | 2 | $0.90 |

| | | | |
|---|---|---|---|
| | (L7805CV) | | |
| 3.3V Linear Regulator | Microchip (TC1262-3.3VAB) | 2 | $1.98 |
| Flex Sensor | Spectrasymbol | 3 | $23.85 |
| Power MOSFET | ON Semiconductor (RFP12N10L) | 2 | $1.84 |
| Accelerometer | Analog Devices (ADXL335) | 1 | $11.66 |
| Transceiver Module | Nordic Semiconductor (NRF24L01) | 2 | $2.40 |
| Ultrasonic Sensor | Banana Robotics (HC-SR04) | 1 | $3.95 |
| Microcontroller | Microchip Technology (ATMega 328p) | 2 | $5.00 |
| DC Motor | | 2 | |
| Car frame & Wheel | | 1 | |
| 9V Alkaline battery | AmazonBasics | 3 | $3.57 |
| Flyback diode | ON Semiconductor (1N4148) | 2 | $0.20 |
| Crystal Oscillator | Fox Electronics (FC4STCBMF16.0 ) | 2 | $0.70 |
| PCB manufacturing | | 2 | |
| | | TOTAL | |

## 4.2 Labor Cost

To get an acceptable value of labor cost, we decide to set hour hourly wage to $25, which is an average hourly wage for engineering interns. By estimation, we have spent about 50 hours per person on this project. Using the formula give, our total labor cost will be :

$25/hr x 50hrs x 3ppl x 2.5 = $9375

# 5. Conclusion

We have successfully designed, built, and test our project. The functionalities are within our expectations. When the user wears the glove, he/she is able to complete the designed gestures without any difficulties, and the car is able to respond to the user's gestures correctly. However, our user experience is not perfect. There are delays when the car is responding to the user's commands and sometimes, the car is too sensitive to the user's commands. To address these issues in the future, we are going to do more calibrations to our control signals in order to maximize the user experience.

There are several safety issues arising from our project, both during the design process and in the applications. During the design period, the first big safety aspect is the usage of battery. We rely on batteries to power all other electrical components. Well designed circuitries to connect batteries with, the careful storage and handling of batteries, and the prevention of overheating batteries with extended usage time are all extremely important to the safety of dealing with batteries. Second, there are lots of safety concerns related to building electrical components. These concerns include: soldering safety, circuit testing with power on, and safety of usage of external devices. To minimize the risks, we must comply with the standard safety rules. Third, there's some mechanical work involved in this project. There are safety issues of conducting mechanical work as well, and we must again comply with the standard safety rules. Finally, a remote controlled mobile car also poses safety issues. We can't guarantee the car will be controlled as intended all the time. As a result, we need to make sure the car will not crash with anything or anybody to cause any hazard when testing. This can be done by choosing an empty field, or putting protective barriers on the field. All these safety analysis and precaution/prevention efforts align with the IEEE Code of Ethics, #1: "to hold paramount the safety..." [6].

On the application side of our project, there is also a noticeable potential safety concern. Users sometimes might give bad control commands due to inappropriate usage or even malicious intention. In addition, a hacker might be able to hack into a user's control system and make the mobile device do ill-intended things. In such cases, the mobile device would cause hazard, such as hitting people or destructing infrastructures. This is against the IEEE Code of Ethics, #9: "to avoid injuring others, their property, reputation, or employment by false or malicious action" [6]. One precaution is to build an automatic stop function so that the device will halt when it's too close to certain types of objects.

# 6. Reference

[1] A. Bratukhin, ‘Connect ATmega328 with HC-05 (ZS-040) Bluetooth module’, 2015. [Online]. Avaliable:
http://abratukhin.blogspot.com/2015/04/connect-atmega328-with-hc-05-zs-040.html.
[Accessed: 19-Feb-2018].

[2] STMicroelectronics, ‘LIS331DLH: MEMS digital output motion sensor ultra low-power high performance 3-axes “nano” accelerometer ’, 2009. [Online]. Avaliable:
http://www.st.com/content/ccc/resource/technical/document/datasheet/9c/d6/7d/9b/06/6b/40/9d/CD00213470.pdf/files/CD00213470.pdf/jcr:content/translations/en.CD00213470.pdf.
[Accessed: 19-Feb-2018].

[3] R. Nuggent, ‘Building an Arduino on breadboard’ 2008. [Online]. Avaliable:
https://www.arduino.cc/en/Main/Standalone.
[Accessed: 21-Feb-2018].

[4] Atmel, ‘Atmega328/P: Datasheet Complete’, 2016. [Online]. Avaliable:
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf.
[Accessed: 21-Feb-2018].

[5] PIC Tutorials, ‘nRF24L01 Test board PCB’, 2014. [Online]. Avaliable:
http://wordpress.codewrite.co.uk/pic/2014/01/13/nrf24l01-test-board-pcb/.
[Accessed: 21-Feb-2018].

[6] Ieee.org, "IEEE IEEE Code of Ethics", 2018. [Online]. Available:
https://www.ieee.org/about/corporate/governance/p7-8.html.
[Accessed: 21-Feb-2018].

[7] Mouser, “STMicroelectronics L7805CV”, 2014.[Online]. Available:
https://www.mouser.com/productdetail/?qs=9NrABl3fj%2FqplZAHiYUxWg%3D%3D.
[Accessed: 22-Feb-2018]

[8] Mouser, “ON Semiconductor / Fairchild  RFP12N10L”, 2010.[Online]. Available:
https://www.mouser.com/ProductDetail/?qs=FOlmdCx%252bAA3tZr96h3LX1g%3D%3D.
[Accessed: 22-Feb-2018]

[9] Digi-key, “NMB Technologies Corporation PAN14EE12AA1”, 2014.[Online]. Available:
https://www.digikey.com/product-detail/en/nmb-technologies-corporation/PAN14EE12AA1/P14346-ND/2417070.
[Accessed: 22-Feb-2018]

[10] Analog Device, “AN-1057 Application Note”, 2010. [Online]. Available: http://www.analog.com/media/en/technical-documentation/application-notes/AN-1057.pdf. [Accessed: 21-Feb-2018]

[11] Analog Device, “Small, Low Power, 3-Axis Accelerometer ADXL335”, 2010. [Online]. Available:
http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf
[Accessed: 20-Feb-2018]

[12] Nordic Semiconductor, “Single Chip 2.4 GHz Transceiver nRF24L01”, 2006. [Online]. Available:
https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf
[Accessed: 20-Feb-2018]

[13] Banana Robotics, “HC SR04 Ultrasonic Distance Sensor Module”, 2013. [Online]. Available:
https://www.bananarobotics.com/shop/HC-SR04-Ultrasonic-Distance-Sensor?gclid=Cj0KCQiAzrTUBRCnARIsAL0mqczFmBjQ5RfO9Qw0Q6Z6e5C75gUGBQpfnWbW3_os2lA4AmDzKw_zfjcaAuotEALw_wcB
[Accessed: 21-Feb-2018]

[14] D. Nedelkovski, “Arduino Wireless Communication - NRF24L01 Tutorial”, 2017. [Online]. Available:
http://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/
[Accessed: 21-Feb-2018]