Hand-Controlled Car

ECE 445 Design Document - Spring 2018

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

1. Introduction

1.1 Objective

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. 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. Currently we plan to 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.

1.2 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 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. There are 2 separate switches on the glove and the car. When they are both turned on, the communication starts.

2. Design

2.1 Block Diagram

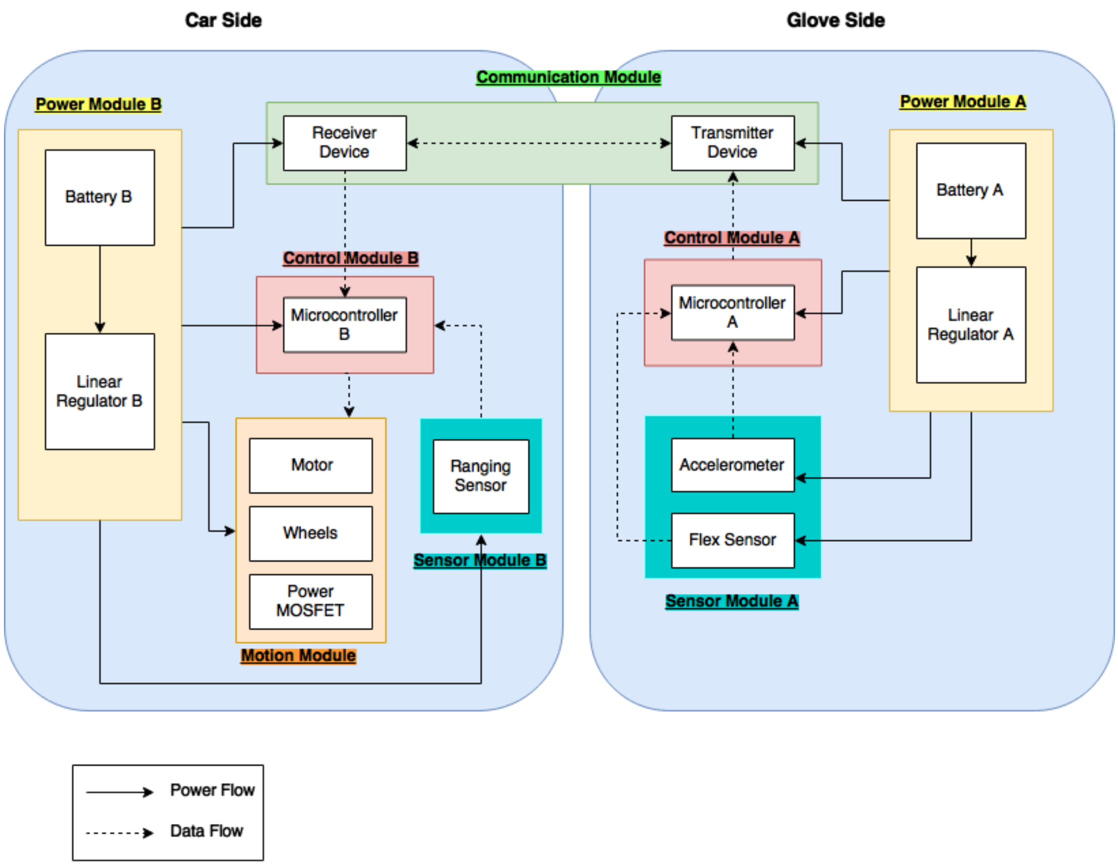

Figure 1. Block Diagram

2.2 Physical Design Diagram

2.2.1 Physical Design Diagram of Glove

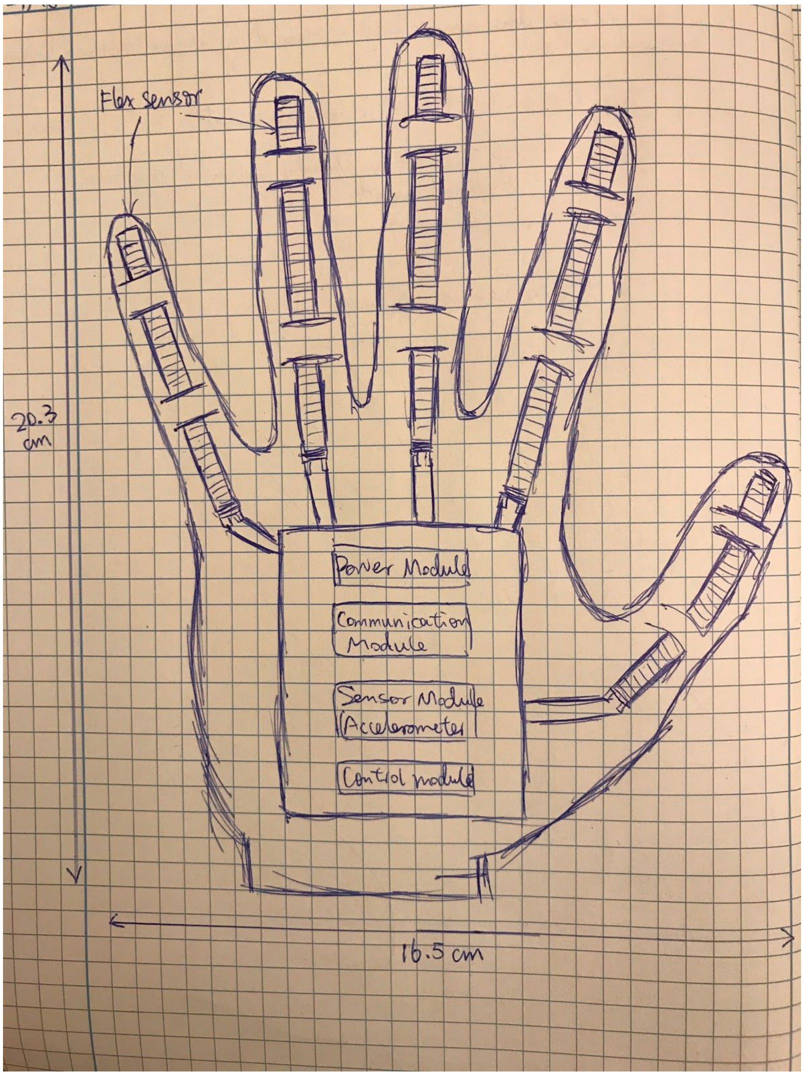

Figure 2. Physical Design Diagram of the Glove

2.2.2 Physical Design Diagram of Car

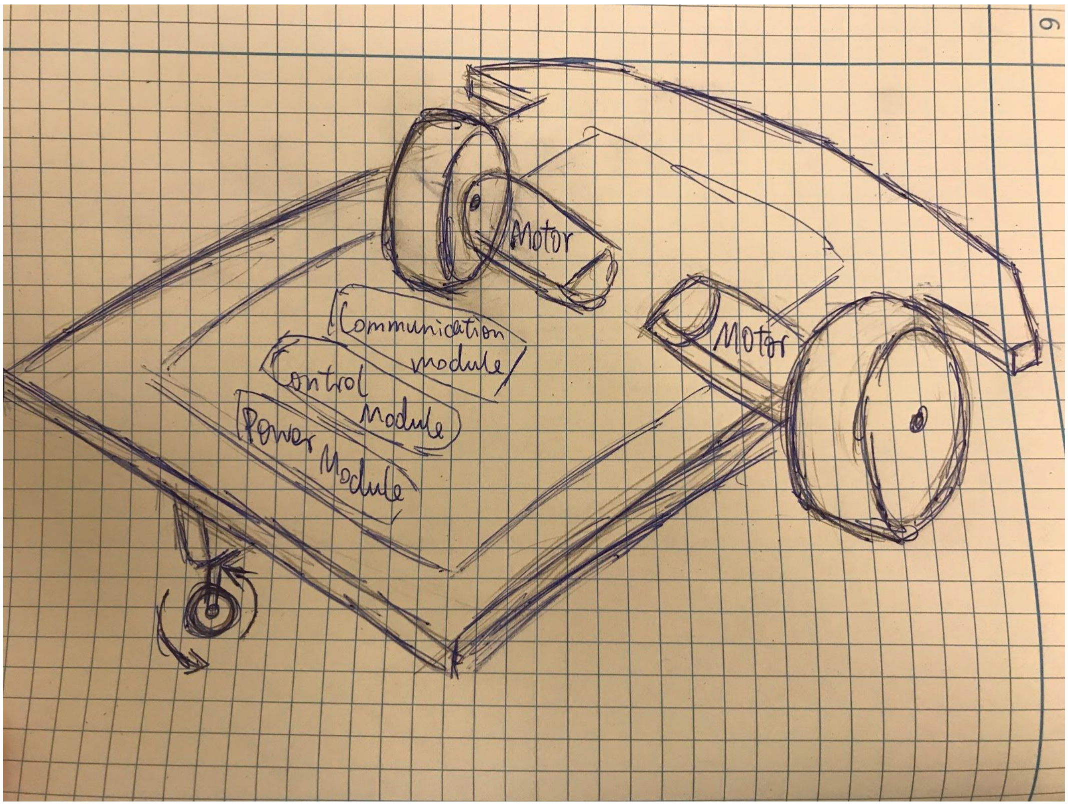

Figure 3. Phyical Design Diagram of the Car

2.3 Block Design

2.3.1 Power Module A

This module serves to provide power to all the electrical components on the glove, including three flex sensors, one accelerometer, a microcontroller, and a wireless communication transmitter. The module will provide stable 5V and 3.3V voltages at all times. It consists of a battery and a linear voltage regulator.

2.3.1.1 Battery A

A 12V Energizer A23 battery will be used here to provide power. The battery itself 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.

Requirements	Verification

1. Must be able to provide a voltage of 7-12V under continuous operation(Minimum input voltage for regulator is 7V) 2. Must stay under 60C under all conditions	1. a. Before using the glove everytime, use a voltmeter to measure the voltage level of battateries. b. If voltage level is dropping close to 7V, meaning the battery is about to be dead, change to a new battery 2. a. Use hand to feel the temperature, if fingers cannot stay on battery over 1sec, take the battery out and let it cool down

2.3.1.2 Linear Voltage Regulator A

A linear voltage regulator is used here to regulate the voltage level to 5V. After doing some comparison, the STMicroelectronics L7805CV meets all of our requirement. It takes wide range of voltage input from 7V to 35V. Its output current is 1A. It can operate normally under up to 125C environment, which makes it a very sufficient product for our project. On the glove, when some power in the battery is consumed, voltage will drop and won't be at the 12V level. With the help of this linear voltage regulator, microcontroller and sensors will always be able to have a stable 5V input voltage.

Power Consumption:

The operating voltage is 4.8-5.2V, and the current is 5mA-1A. Therefore, the power will be 0.024-5.2W.

Requirements	Verification
1. Provides 5V+/- 5% from a 12V source at 1A output current 2. Maintain temperature under 125C	1. a. Use a constant-current circuit, draw 1A current from this linear voltage regulator b. Use oscilloscope to monitor output voltage, ensuring it's within the 5V+/- 5% range. 2. a. When doing the step above, use an IR thermometer to ensure the IC temperature stays under 125C.

2.3.2 Sensor Module A

This module contains the two types of sensors on the glove that are needed for hand gesture detection. The first type of sensor is flex sensor, and it will sense the curl of the fingers. The second type of sensor is accelerometer, which can sense the orientation of the hand. The sensor output data from this module is transferred to the Control Module A for further usage.

2.3.2.1 Flex sensor

Three flex sensors will be used to sense the curl of the fingers (This gesture controls the drive/stop as well as the speed of the car). They will be attached to the back of the three longest fingers. A 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).

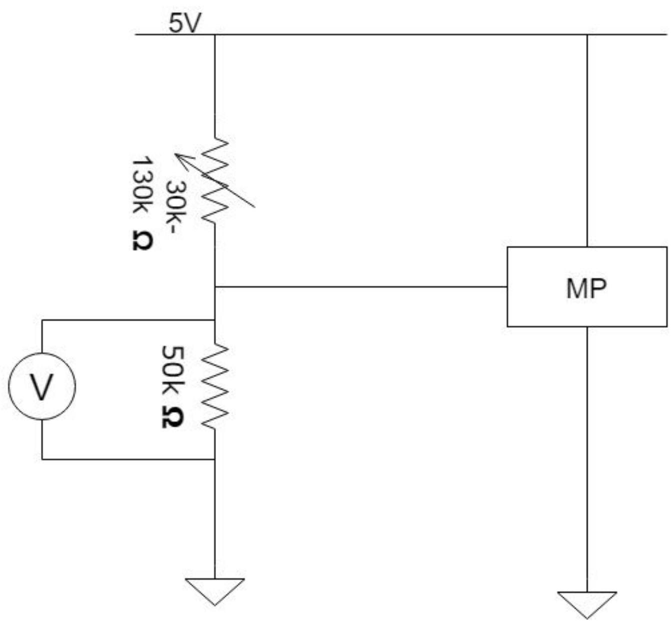

Figure 4. Circuit for Feeding Flex Sensor Measurement into a Microcontroller

For component testing, we built the above 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

Requirements	Verification
1. The input voltage range to the microcontroller pins (shown in the circuit diagram above) must vary from 1.389V to 3.125V	a. Like above, connect a voltmeter across the 50kOhm resistor. (Flex sensor is stretched now) Ensure voltage is around 3.125V b. Gradually bend the flex sensor, ensure the voltage is gradually decreasing c. When the flex sensor is bent to its limit, ensure voltage is around 1.389V

2.3.2.2 Accelerometer

An accelerometer will be used to sense the orientation/tilt angle of the glove which controls the turning of the car. The model we will use is ADXL 335 3-Axis Accelerometer. It 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)

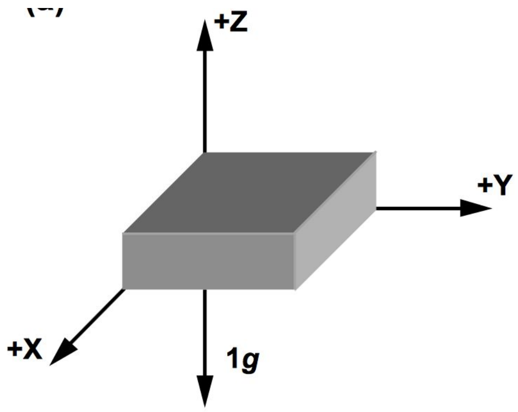

Figure 5. Accelerometer on a flat surface

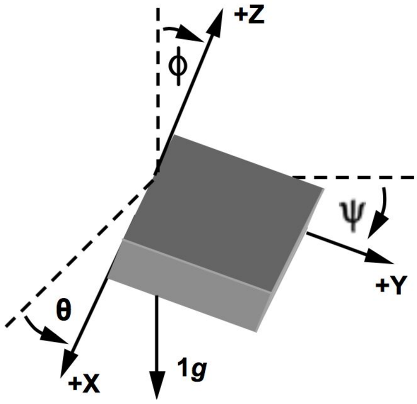

Figure 6. Accelerometer at anTilted Orientation

For component testing, 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

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)

Requirements	Verification
1. ADXL335 must be able to sense and transfer its output at a minimum frequency	1. a. connect the ADXL335 with an

of 100 Hz 2. Its analog voltage output must change at least 100mV per 15 degrees of rotation around any axis	Arduino Uno b. setup properly and write codes to receive real-time output data from the accelerometer c. use the timing functionalities in Arduino to calculate the frequency of the ADXL335 output data, and make sure it's above 100 Hz 2. a. connect the ADXL335 to an oscillascope b. manually rotate the chip around any axis at steps of 15 degrees c. record the corresponding axis's output voltage value, make sure it changes at least 100 mV d. repeat for a total rotation of ∓ 90 degrees for all three sensing axes

2.3.3 Control Module A

This control module on the glove will continuously take in analog sensing data from the Sensor Module A, convert it to digital signals, and send properly assembled data packages to the transmitter device in the communication module. This module consists of a microcontroller.

2.3.3.1 Microcontroller A

We will use Atmega328 microcontroller for the control module on the glove. 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.

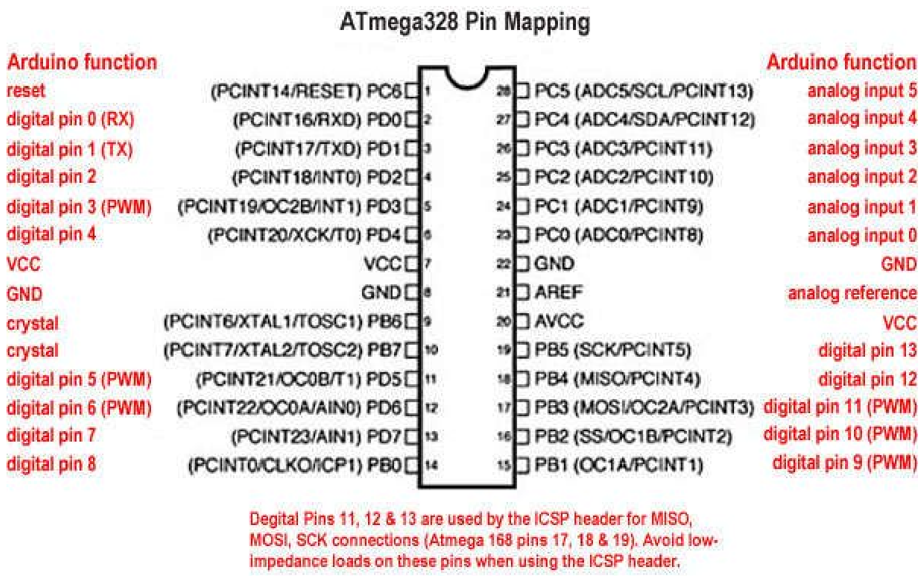

Figure 7. Pin map of Atmega328

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.

Requirements	Verification
1. The Atmega328 chip must be bootloaded before using. 2. The Vcc has the ratings of 1.8V to 5 V, maximum 6V. The voltage on the A/D pin with respect to ground can take the range from -0.5V to Vcc+0.5V.	a. Before pulling the Atmega chip out of the Arduino board, make sure it has been programmed several times. b. Before connecting any component to the microcontroller, make sure to measure its voltage using the multimeter to check if it's in the acceptable range.

2.3.4 Communication Module

This module serves to implement the wireless communication between the glove side and the car side. It consists of a transmitter device on the glove and a receiver device on the car. The transmitter device continuously receives the real-time data sent from the control module A on the glove, and transmit it wirelessly to the receiver device on the car. The control module B continuously reads the data received by the receiver device and performs further processing.

2.3.4.1 Transmitter Device

We will use an NRF24L01 transceiver module to wirelessly transmit real-time data from the glove side to the car side. It will pair with another identical NRF24L01 transceiver module on the car (set as receiver) to form the communication. 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.

<table>
<tr><th>Requirements</th><th>Verification</th></tr>
<tr><td>1. It must be able to receive the custom data packages from the microcontroller, and transmit them to the receiver NRF24L01 15m away, at a minimum rate of 100 packages per second.
2. The VCC pin on the transceiver module should connect to 3.3 V supply.</td><td>a. connect one NRF24L01 transceiver to an ATmega328 microcontroller
b. set the transceiver as transmitter
c. connect another NRF24L01 to another ATmega328, placed at 15m away from the transmitter
d. set it as receiver
e. Through ATmega328 programming interface, send custom data packages to the transmitter at a rate of 120 packages/second
f. Through ATmega328 programming interface at the receiving end, record the number of packages received in one second. Make sure at least 100 packages are received by the receiver and sent to the microcontroller
2.
a. Before connecting the VCC, use multimeter to measure the voltage to make sure it is 3.3 V.</td></tr>
</table>

2.3.4.2 Receiver Device

For the receiver device, we will use another NRF24L01 transceiver module by setting it as receiver. It will continuously receive the real-time data sent by the transmitter NRF24L01, and then send it to the microcontroller B on the car. The communication between the receiver and microcontroller B is the same as that between the transmitter and microcontroller A, except that the direction of the data transfer is reversed. The specs of the receiver is also the same as that of the transmitter, since they are two identical NRF24L01 modules.

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.

Requirements	Verification
1. The receiver must receive data packages and send them to the microcontroller with minimum 95% accuracy (compared with the original data packages at the transmitting end) 2. The VCC pin on the transceiver module should connect to 3.3 V supply.	a. do the same steps as in the verification for the transmitter, except this time, send 100 packages only once from the transmitter end b. Through the ATmega328 programming interface at the receiving end, record the received packages and make sure at least 95 of them match the original packages 2. a. Before connecting the VCC, use multimeter to measure the voltage to make sure it is 3.3 V.

2.3.5 Power Module B

This module serves to provide power to all the electrical components installed on the car. Components include, two 12V motors, a microcontroller, a wireless communication receiver device. This power module provides continuous and stable 12V to two motors and 5V to microcontroller and the receiver.

2.3.5.1 Battery B

Two 12V Energizer A23 battery, directly meeting voltage requirement of two 12V motors. The battery itself only weighs 0.8 ounces, with a dimension of 2.9x1.8x0.5 inches. It has enough power to continuously support the components on the car. Even when batteries are losing power therefore dropping voltage level, it will still be able to correctly operate two motors on the car.

Requirements	Verification

1. Must be able to provide a voltage of 7-12V under continuous operation(Minimum input voltage for regulator is 7V) 2. Must stay under 60C under all conditions	1. a. Before using the glove everytime, use a voltmeter to measure the voltage level of batteries. b. If voltage level is dropping close to 7V, meaning the battery is about to be dead, change to a new battery 2. a. Use hand to feel the temperature, if fingers cannot stay on battery over 1sec, take the battery out and let it cool down

2.3.5.2 Linear Voltage Regulator B

Same issue as above, a linear voltage regulator is used here to regulate the voltage level to 5V. After doing some comparison, the STMicroelectronics L7805CV meets all of our requirement. It takes wide range of voltage input from 7V to 35V. Its output current is 1A. It can operate normally under up to 125C environment, which makes it a very sufficient product for our project. On the glove, when some power in the battery is consumed, voltage will drop and won't be at the 12V level. With the help of this linear voltage regulator, microcontroller and sensors will always be able to have a stable 5V input voltage.

Power Consumption:

The operating voltage is 4.8-5.2V, and the current is 5mA-1A. Therefore, the power will be 0.024-5.2W.

Requirements	Verification
1. Provides 5V+/- 5% from a 12V source at 1A output current 2. Maintain temperature under 125C	1. a. Use a constant-current circuit, draw 1A current from this linear voltage regulator b. Use oscilloscope to monitor output voltage, ensuring it's within the 5V+/- 5% range. 2. a. When doing the step above, use an IR thermometer to ensure the IC temperature stays under 125C.

2.3.6 Sensor Module B

This module generates the distance sensing data needed for the car's anti-collision feature. It consists of a ranging sensor that will continuously output the distance sensing data. The data is transferred to the Control Module B for further processing.

2.3.6.1 Ranging Sensor

We will use HC-SR04 Ultrasonic Ranging Sensor Module for distance measurement. It will be placed at the front of the car and will face forward. It 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.

For component testing, 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

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

Requirements	Verification
1. The sensor must provide the distance measurements of an object placed in a range of 30cm-2m away in front of the car with at least 90% accuracy.	a. Connect the sensor with a ATmega328 board and setup properly b. place an object in front of the sensor starting from 30cm away and ending at 2m with steps of 10cm. c. at each step, record the distance measurement by the sensor d. calculate the accuracy and make sure it's above 90% for all the steps

2.3.7 Control Module B

This control module on the car will be able to use SPI communication to control the communication module on the car to receive information from the glove, process the information and give control signals to the motion module which is responsible for the movement of the car.

2.3.7.1 Microcontroller B

We will use Atmega328 microcontroller for the control module on the car. The SPI serial port on Atmega328 will handle the SPI communication with the receiver device. The Port D (PD[7:0]) , which is is an 8-bit bi-directional I/O port with internal pull-up resistors, will be responsible for the analog PWM output for the motors on the car. 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.

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.

Requirements	Verification
1. The Atmega328 chip must be bootloaded before using. 2. The Vcc has the ratings of 1.8V to 5 V, maximum 6V.	a. Before pulling the Atmega chip out of the Arduino board, make sure it has been programmed several times. b. Before connecting any component to the microcontroller, make sure to measure its voltage using the multimeter to check if it's in the acceptable range.

2.3.8 Motion Module

This module takes in the control signals from the control unit (microcontroller B) and makes the car move in the desired way. The module contains two motors, power MOSFET, and wheels. The motions of the car include turning left/right and increase/decrease speed. Especially, the turning of the car will be achieved by different spinning speeds of the two frontal wheels.

2.3.8.1 Motors

Two PAN14 motors will be used to drive our car. PAN14 motor is one of the cheapest but high-performance motors. One PAN14 motor weighs only 39g. It has a rated voltage of 12V. Under its rated load(4.9mN*m), it has rated load current of 0.563A. The motors will be controlling the speed of the wheels. A PWM signal generated by microcontroller is sent to the RFP12N10L Power MOSFET, which is connected with a motor in series. The speed change of motors can be adjusted by changing the duty cycle of PWM signal. The car will turn left and right with two motors spinning in different speed. They are fed the control signals from microcontroller B and make the individual wheels rotate accordingly.

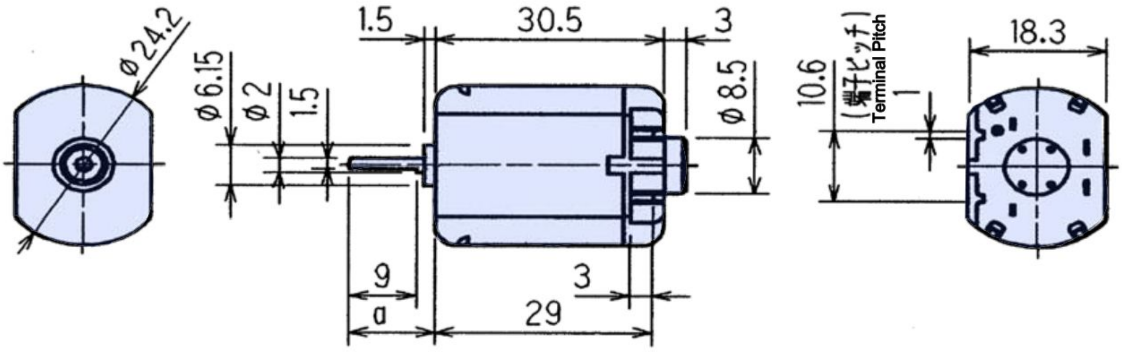

Figure 8. Physical Diagram of Motor

Requirements	Verification
1. Can operate with VDC 12V+/-1V 2. Motor current doesn't go over 1.02A under any operating condition	1, 2. a. Connect motor with an Energizer A23 battery(12V). Battery doesn't have to be full b. Install wheel on the motor c. Use a voltmeter to monitor the battery voltage; use a current probe to monitor the motor current d. Ensure both of the values are within the range

2.3.8.2 Power MOSFET (RFP12N10L)

In our design, the speed of the motor is controlled by the PWM power supply. The higher the duty cycle of PWM, the faster the motor speed. Our ATmega328 microcontroller is able to output an PWM with high of 5V. However, our motor has an operating voltage of 12V. Thus, me and my teammates designed an MOSFET circuit to use the microcontroller PWM signal to control the on and off of the motor.

We use RFP12N10L n-MOSFET to achieve this switch. At the Gate, we connect the PWM from microcontroller in series with a 10 kΩ resistor to the ground. We put a 220 Ω resistor to represent the motor between the 12V supply and the Drain. Then, we connect the source to the ground. Below is the circuit schematic of the power MOSFET circuit.

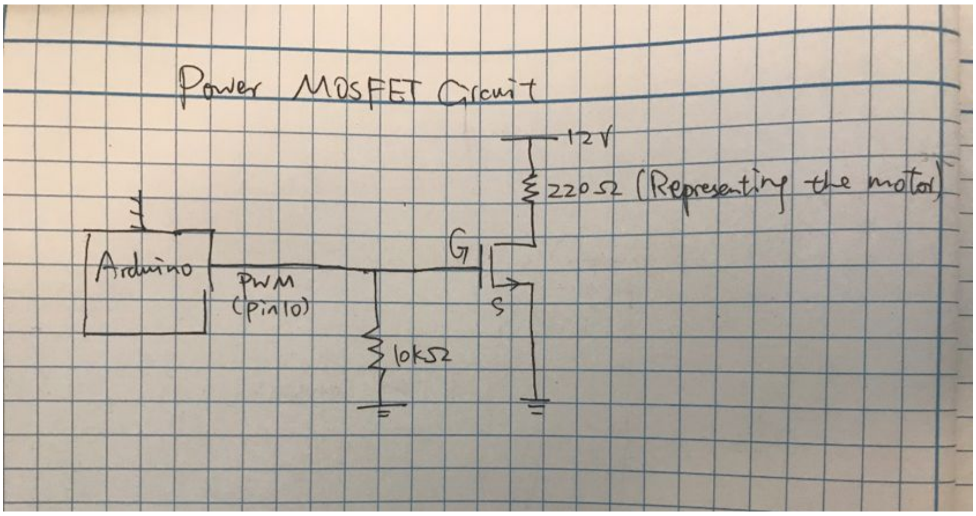

Figure 9. Power MOSFET Circuit Schematic

Requirements	Verification
1. RFP12N10L must have a voltage drop less than 0.5V 2. It must reacts effectively when the PWM duty cycle changes	1. a. Connect RFP12N10L Power MOSFET with motor and an Arduino Uno b. Connect a volt meter across the MOSFET c. Run code changing duty cycle for 20% every 2 seconds d. Monitor voltage across the MOSFET, ensuring it's within the range 2. a. Keep the connection from Step 1, keep the code running b. Look at the motor, see if it is able to react to every PWM duty cycle change fastly.

2.3.8.3 Wheels

There will be three wheels for the car. Two front wheels will be individually powered by two motors. The third wheel is a dummy wheel without any motor connection, placed at the center on the rear. It can rotate all degrees freely and its main purpose is to make the turning of the car smoother.

Requirements	Verification
1. Wheels can stably stay on the motor under motor full speed	a. Install wheels on both motors, put car upside down b. Connect motors to 12V battery, ensure it operates in full speed. Let it run for 60sec. c. Check to see if wheels are still installed tightly on the motor

2.4 Circuit Schematics

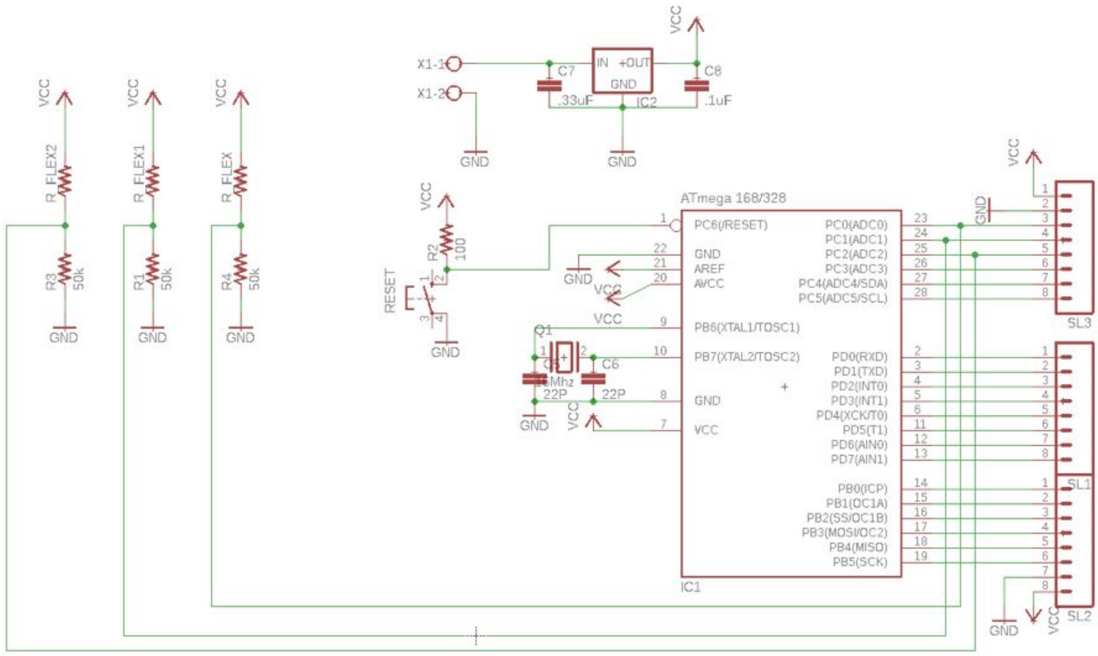

Figure 9. Accelerometer Circuit on the Glove Schematic

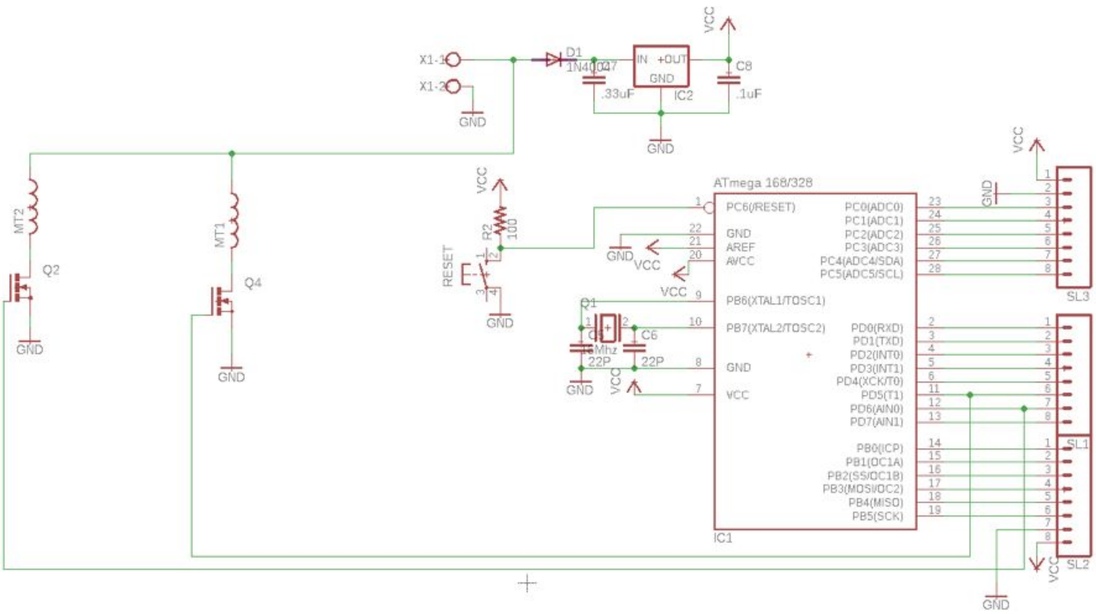

Figure 10. Control Circuit on the Glove Schematic

2.5 Control Unit Flowchart

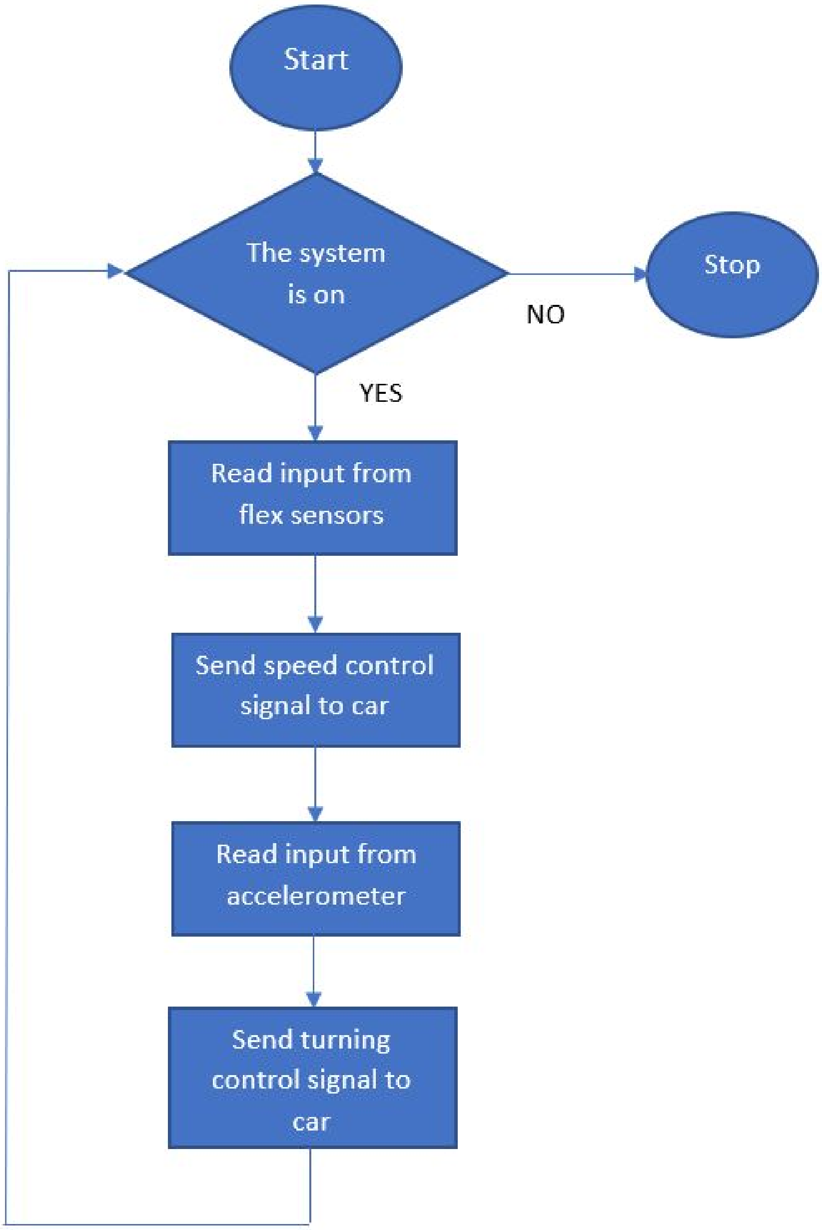

Figure 11. Algorithm of the microcontroller on the glove

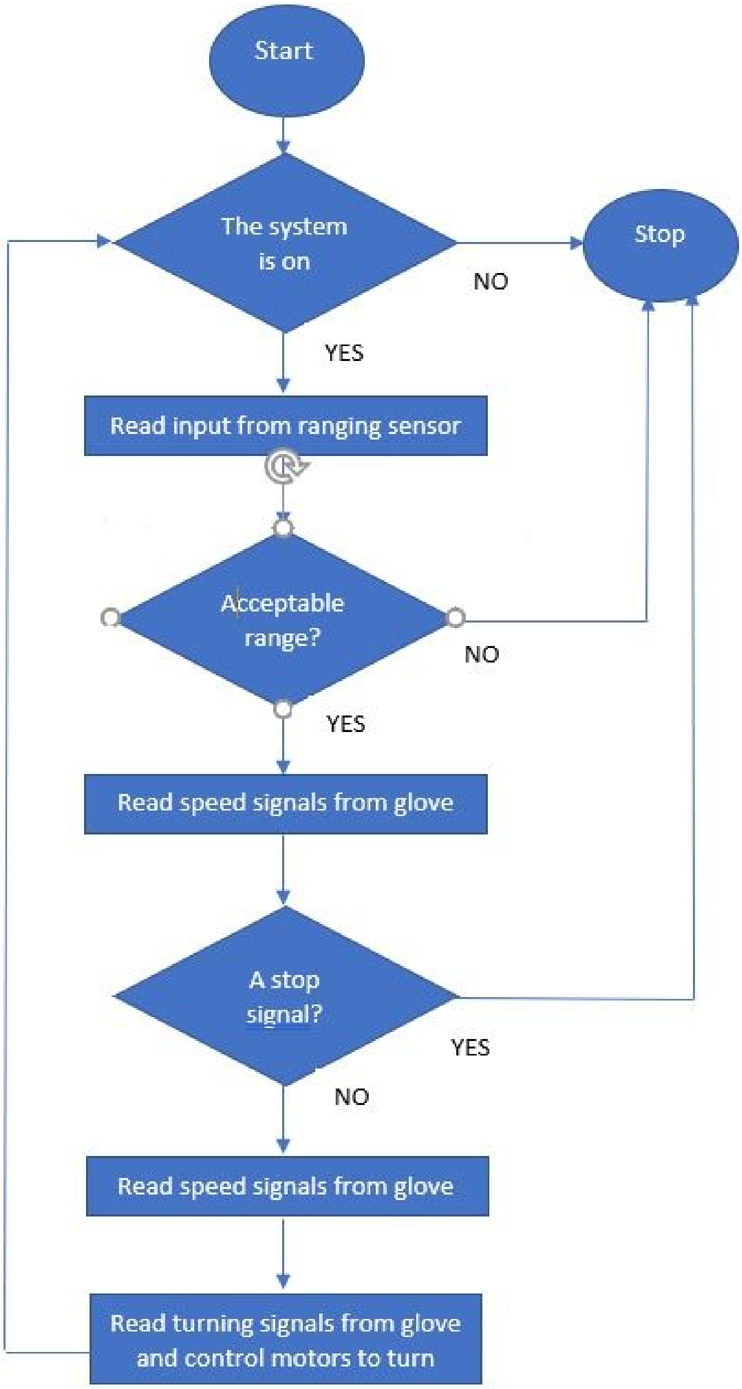

Figure 12. Algorithm of the microcontroller on the car

2.6 Tolerance Analysis

For the tolerance analysis, we will focus on the ADXL335 accelerometer in the sensor module A. We chose this component for two main reasons. First, its proper measurements of the hand orientation are essential for the correct turning of the car. Second, compared with other sensors in the project, the accelerometer has higher precision requirements and thus has more stringent tolerance.

The important feature of the accelerometer is its measurement sensitivity, namely the voltage change per given amount of acceleration change. As can be seen in the plots below, the typical sensitivity is 0.303 V/g for x-axis, 0.306 V/g for y-axis, and 0.300 V/g for z-axis.

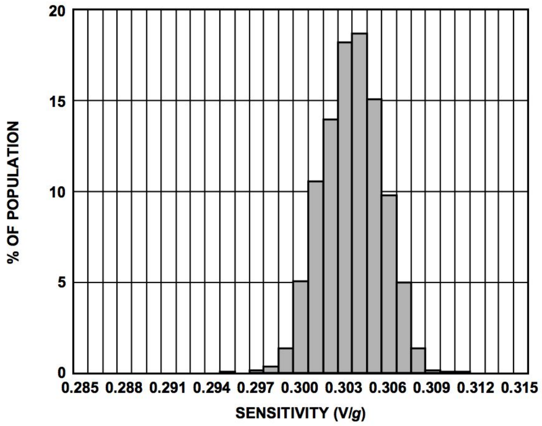

Figure 13. X-Axis Sensitivity at 25℃, Vs = 3V

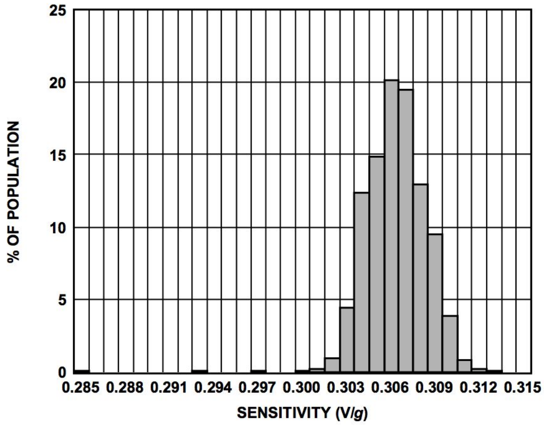

Figure 14. Y-Axis Sensitivity at 25℃, Vs = 3V

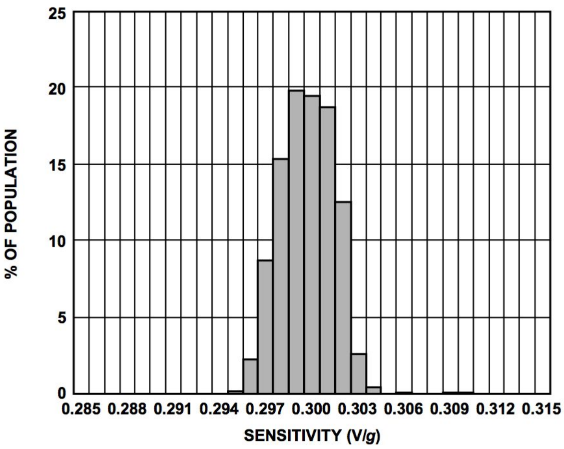

Figure 15. Z-Axis Sensitivity at 25℃, Vs = 3V

For our task of hand inclination measurement, the static acceleration due to gravity will change between +g to -g, corresponding to the hand rotation between +90 degrees to -90 degrees. We need to consider how much the sensitivities can deviate from the expected values so that the microcontroller will still be able to obtain the accurate hand orientation measurement. The ATmega328p microcontroller has 10-bit ADC, which means it can do 1024 bits of resolution. As a result, it can produce a resolution of 5V/1024 = 0.005V. If we want to be able to distinguish a change of 15 degrees tilt angle (2g/12 = g/6 change in acceleration), the minimum sensitivity of any axis is 0.005V per g/6, which is 0.03V/g. From the large of number of samples in the plots above, even with potential deviations, all three axes' sensitivities are still much larger than the minimum requirement. Another good thing here to note is that the large difference also allows other external electrical noises and voltage level fluctuation to exist without affecting the measuring accuracy.

3. Cost

Part	Cost
12V to 5V STMicroelectronics Voltage Regulator *2 (Mouser; L7805CV)	$0.45 * 2 = $0.9
Fairchild Semi Logic level Power MOSFET *2 (Mouser; RFP12N10L)	$0.92 * 2 = $1.84
PAN14 DC motor *2 (Digikey; PAN14EE12AA1)	$4.66 * 2 = $9.32
Energizer 12V A23 battery * 12(pack)(Amazon)	$9.49
ADXL335 Accelerometer (Sparkfun)	$15.95
Flex Sensor *3(Sparkfun)	$7.95 * 3 = $23.85
NRF24L01 Transceiver *2 (Amazon)	$5.99
HC SR04 Ultrasonic Sensor (Banana Robotics)	$2.99
ATmega328p *2 (Adafruit)	$5.95 *2 = $11.9

Total Cost of Parts: $82.23
Our development human labor cost is calculated as below:
$40/hr * 10hrs/week * 10 weeks * 3 people = $12,000
As a result, the approximated total cost of this project is $82.23 + $12,000 = $12,082.23.

4. Schedule

Week	Tianyang Zheng	Jingyu Li	Bohan Hu
2/26	Prepare for design review; Place order for all the components	Prepare for Design Review; Place order for all the components	Prepare for Design Review; Place order for all the components
3/5	Finish all the PCB designs related with accelerometer, flex	Begin working on the PCB design of microcontroller	Start working on the design of glove and Car PCB; Finish the

	sensor, transmitter and receiver, ranging sensor	circuits on the glove and car.	mechanical design of the car
3/12	Get correct output data from the accelerometer through different tilting testings (on oscilloscope).	Order all the PCBs we need. Wire and test the flex senor circuit using breadboard. Begin using Arduino to program the motors.	Figure out the installation of battery connections and power supply circuitry
3/19	build and test the data transfer between accelerometer and microcontroller using Arduino.	Begin using Arduino to program the communication modules.	Assist other two members with communication setup
3/26	Complete the reliable data transfer from sensors to the microcontroller. Make sure the data is reasonable.	Make communication modules work on Arduino	Continue assist other two members with communication and other setup; Place order of modified PCBs(if needed)
4/2	Assemble all the components on the glove.	Assemble all the modules on the glove. Test all the modules on the car	Assemble all the modules on glove side; Test functionality of components on the car
4/9	Assemble the car with all the components on it. Test functionalities of the car.	Assemble all the modules on the car and test the functionalities of the car.	Assemble all the modules on the car; Test the functionality of communication
4/16	Final testing on the entire system, and fix all the bugs	Final testing on all the designs and troubleshooting	Final testing on all the functionalities, fix all the bugs.
4/23	Start writing final report	Start working on final report	Wrap up; Start writing final report
4/30	Finish the report. Prepare final presentation.	Finish report and prepare final presentation	Finish report; Prepare final presentation

5. Safety and Ethics

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]