FINAL PROJECT REPORT **Remote Control Car**

Xin Jin (xinj3)
Haoyu Wang (haoyuw7)
Hongshuo Zhang (hz13)

Introduction

Problem description

Most accidents happen due to people's carelessness. If all cars have a system that warns drivers and automatically stops the car behind close obstacles or pedestrians, many accidents can be avoided. Our project is to design a user-controllable car that warns the user if there is an obstacle near the car. The car will eventually stop if the user is still trying to make the car moving forward when there is an obstacle. Our project has a real-world application if equipped with advanced and more precise sensors. If all cars are equipped with our system, the rate of severe and fatal traffic accidents will decrease. An automatic following system enables people to follow the former car if they are heading to the same destination. Thus the driver in the back can take a rest. Due to the lack of sensors and technical problems, we can only design a simplified mode for the car.

Design concept

This project consists of a car that can be controlled by an app on the phone using an arduino UNO and an arduino Bluetooth module together with accessories. The app has the basic function of making the car move forward, backward, left and right. An H-bridge will satisfy the need. When an object lying in the way of the car, the car will stop automatically and a buffer will make noise as long as ultrasonic sensors detect that the object is too close to the car. For instance, the car is moving forward and a block is put in front of the car with a distance of less than d (the “dangerous” distance). The buffer will alarm people of the block and will not move

forward however the driver touches the “Front” button. There is also an one-dimensional Pet Mode. When the Pet Mode is on, the car will chase the object in front of it and keep a certain distance. When the object is moving away, it will go forward automatically to seek for it and go backward while the object is moving too closely. In addition, an LED system indicates which mode the car is in.

Analysis of components

Characterisation of each sensor

The **ultrasonic sensor** works by taking in an oscillating signal into the TRIG pin and outputting a signal from the ECHO pin. The amount of time that the signal stays active is greater If the distance between the sensor and the obstacle becomes greater and vice versa.

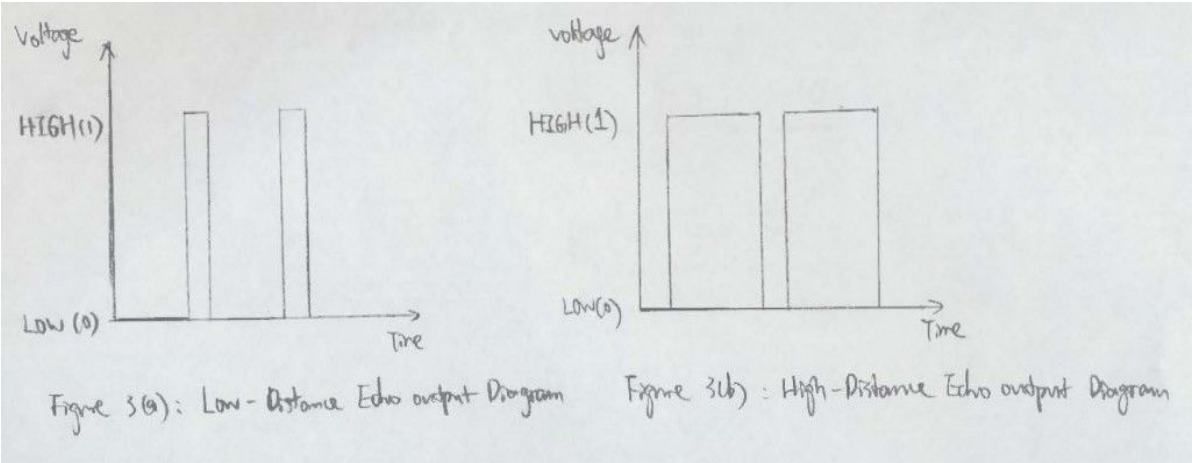


Figure 3(a): Low-Distance Echo output Diagram

Figure 3(b): High-Distance Echo output Diagram

In practice, the duty cycle of the ECHO signal does not exactly conform linearly with the distance from the ultrasonic sensor because, as I discussed previously, obstacles come in all shapes and forms. Some obstacles may have tricky curvatures while other obstacles may have

different sound insulating performances, which might delay or absorb the sound waves sent form the TRIG pin. Luckily, the dependence between duty cycle and distance is not too far off from a linear one. Therefore, we are certain that the ultrasonic sensor is a relatively reliable tool for us to use in our project.

Distance cm	Duty Cycle
2.89 cm	3.89%
10 cm	12.56%
24 cm	21.77%
58 cm	43.65%
102 cm	71.16%
123 cm	71.24%

Table 2: Duty cycle w.r.t to distance

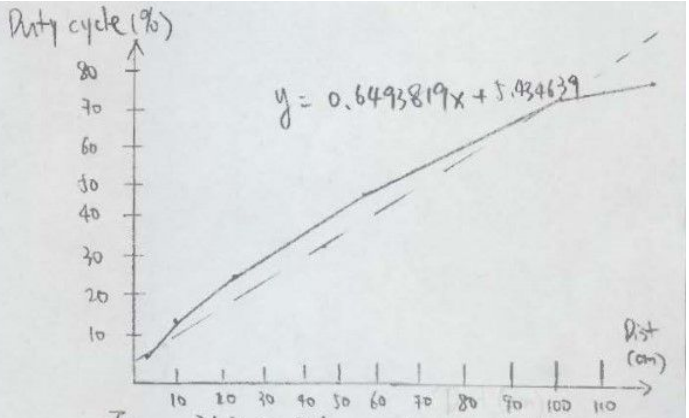


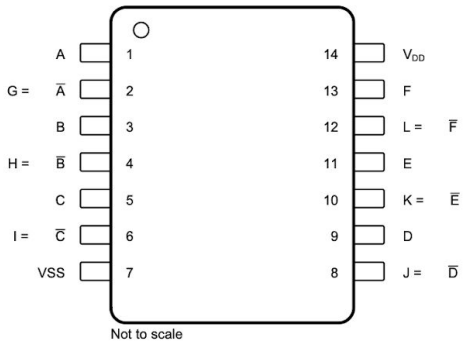
Figure 3(c): Distance - Duty Cycle dependence

Besides that, since we are not able to get a NOT gate, we are using a Schmitt trigger to control the behaviour of our status LEDs. Here is a list of our three status LEDs.

- Red LED indicates that there is an obstacle in front of the car. If there is an obstacle or barrier in front of it, only the Red LED turns on.
- Green LED indicates that there is no *detectable* obstacle in front of the car. If there are no *detectable* obstacles, only the Green LED turns on.
- Blue LED indicates that the **Pet Mode** is turned on.

5 Pin Configuration and Functions

D, J, N, NS, PW Packages
14-Pin SOIC, CDIP, PDIP, SO, TSSOP
Top View



Not to scale

Pin Functions

PIN		I/O	DESCRIPTION
NO.	NAME		
1	A	I	Channel A input
2	G = $\overline{A}$	O	Channel A inverted output
3	B	I	Channel B input
4	H = $\overline{B}$	O	Channel B inverted output
5	C	I	Channel C input
6	I = $\overline{C}$	O	Channel C inverted output
7	V_{SS}	—	Ground
8	J = $\overline{D}$	O	Channel D inverted output
9	D	I	Channel D input
10	K = $\overline{E}$	O	Channel E inverted output
11	E	I	Channel E input
12	L = $\overline{F}$	O	Channel F inverted output
13	F	I	Channel F input
14	V_{DD}	—	Power supply

Among these status LEDs, only the Red LED and the Green LED are connected to the inverter. The Red LED is connected in series with the Buzzer. If the Buzzers turns on, the Red LED also turn on, indicating that there is a detectable obstacle in front and vice versa.

The Red LED is then connected to the input of the inverter (NOT gate). If the Red LED turns on, there will be no signal in the output. So, it makes perfect sense to connect the Green LED to the output of the inverter.

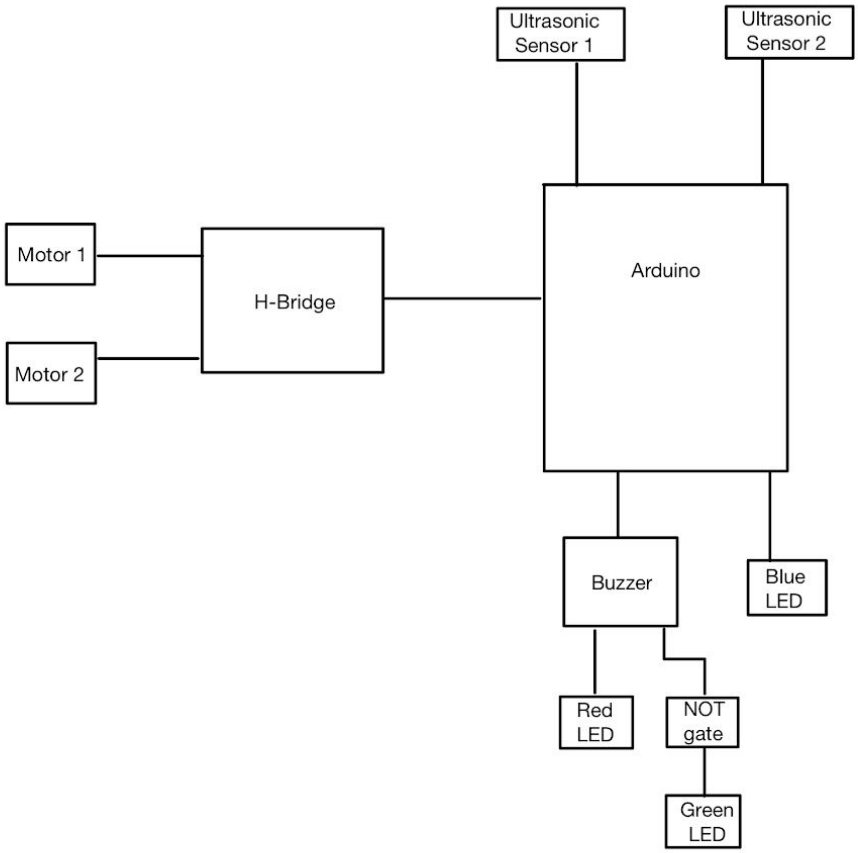
Design Considerations

Since this car can be controlled manually with our phones, we, the controllers, can stop the car from colliding with an obstacle. However, humans do make mistakes. Therefore, we are using the signals from both ultrasonic sensors to prevent us to collide the car with an obstacle. If the distance to the front sensor is below a certain threshold, we are no longer able to control the car to go forward. On the other hand, if the distance to the rear sensor is below a certain threshold, we are no longer able to control the car to go backward.

Later, we also added a new mode called **Pet Mode.** If this mode is activated, there is no need for us control our car manually any more. It automatically keeps a safe distance from the object in front of it. If the object in front of it stops, it also stops; if the object in front of it starts to move, it starts to move as well.

Design Description

Block Diagram



The diagram above shows the main components of this project.

- Two motors are connected to the H-Bridge.
- H-Bridge is connected to the Arduino.
- Two ultrasonic sensors are connected to the Arduino.
- A buzzer is connected to the Arduino.
- Blue LED is connected to the Arduino.
- Red LED is connected in series with the buzzer.
- Green LEd is connected in parallel with the buzzer through a NOT gate.

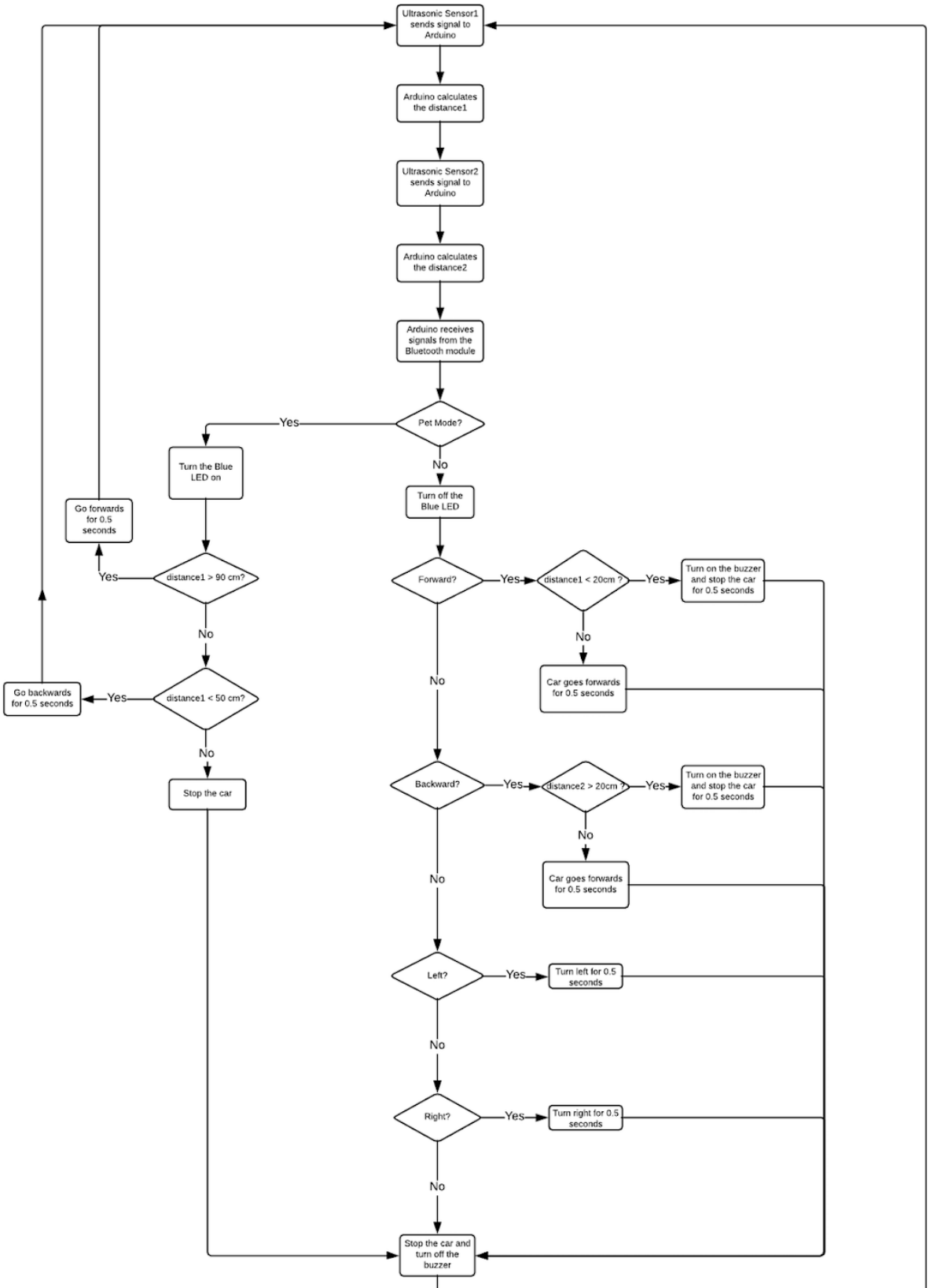
The flowchart below explains the functionality of the codes in the Arduino.

First, the Arduino will receive signals from two ultrasonic sensors and calculate the distance 1 and distance 2 correspondingly. Distance 1 stands for the distance of any obstacle in front of the car, while distance 2 stands for the distance of any obstacle behind the car.

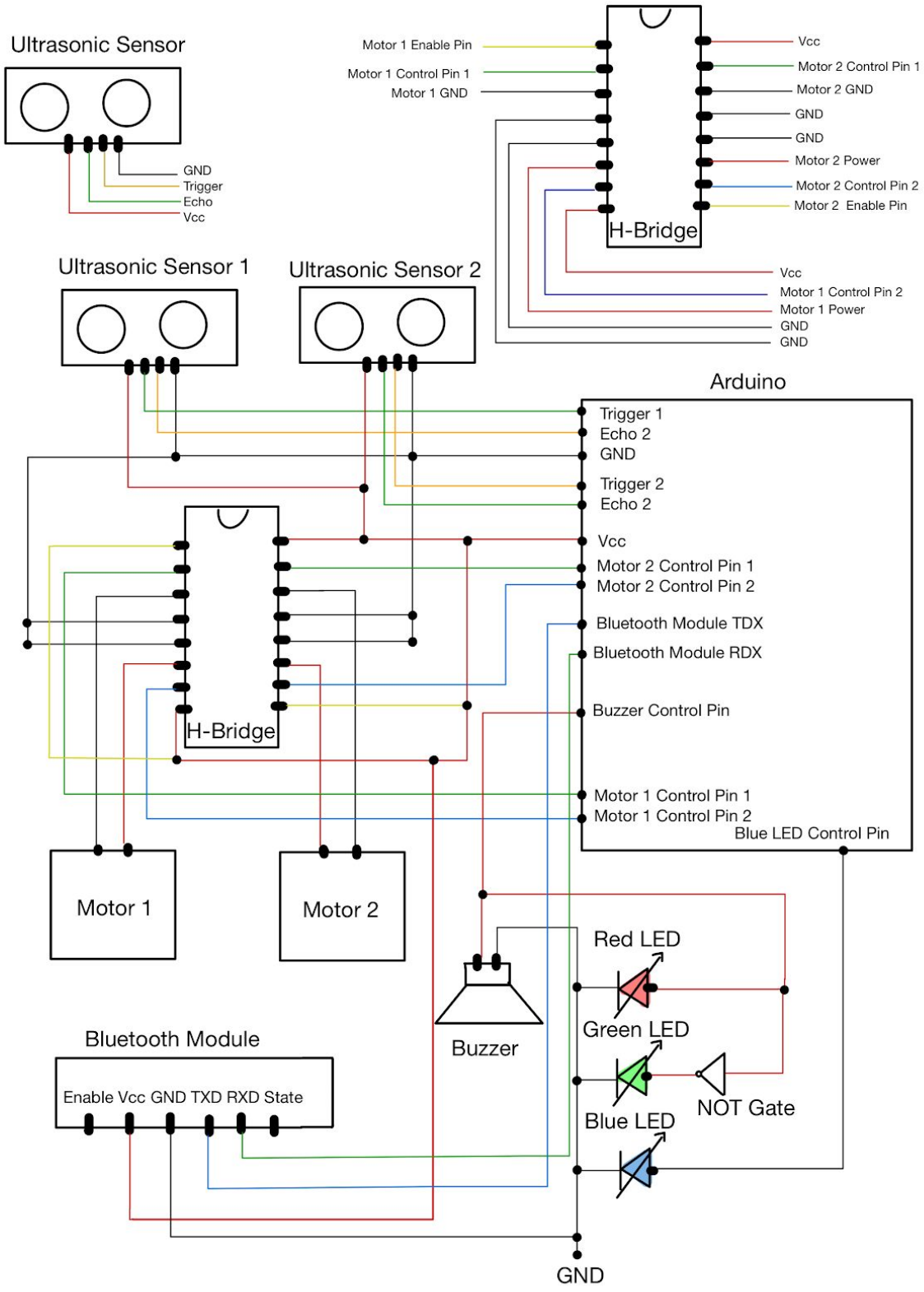
Then, Arduino will receive signals from the bluetooth module. The bluetooth module will send text 1 to 6 to Arduino. 1, 2, 3 and 4 represent forward, backward, left and right respectively. 5 and 6 represent the ON and OFF of the Pet Mode. If it is 5, the Arduino will check if the distance 1 is in the range of 50 cm to 90 cm. If it is in this range, the car will remain stop. If distance 1 is greater than 90 cm, the car will go forwards, or if the distance 1 is less than 50 cm, the car will go backwards until the distance 1 falls back into the range.

If the signal sends from the bluetooth module is 1 or 2, Arduino will check if there is any obstacles in front of or behind the car (distance 1 or distance 2 less than 20 cm). If there is an obstacle, Arduino will prevent the car from any forward or backward movement depending on where the obstacle is and turn on the buzzer to alert the driver. Arduino will only allow the opposite movement to where is no obstacle. There is no restriction on any left turn or right turn movement.

For LEDs, blue LED will be on when the car is in Pet Mode, green LED will be on when there is no obstacle, and red will be on when is there is obstacles. The blue LED is controlled directly by Arduino. Red LED is connected in series with the buzzer, which means the red LED will be on whenever the buzzer is on. The green LED is connected as the output of a NOT gate, while the NOT gate shares the input with the buzzer, which means whenever the buzzer or the red LED is not on, the green LED will be on.

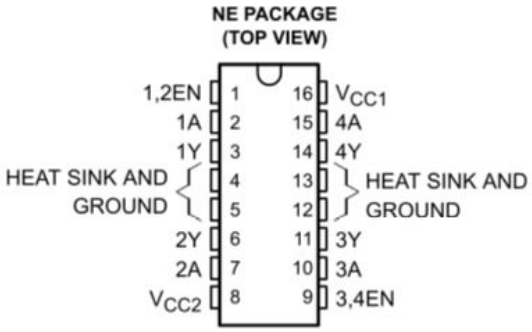

Ultrasonic Sensor1 sends signal to Arduino
Arduino calculates the distance1
Ultrasonic Sensor2 sends signal to Arduino
Arduino calculates the distance2
Arduino receives signals from the Bluetooth module
Pet Mode?
Yes
No
Turn the Blue LED on
Go forwards for 0.5 seconds
distance1 > 90 cm?
Yes
No
Go backwards for 0.5 seconds
distance1 < 50 cm?
Yes
No
Stop the car
Turn off the Blue LED
Forward?
Yes
distance1 < 20cm ?
Yes
Turn on the buzzer and stop the car for 0.5 seconds
No
Car goes forwards for 0.5 seconds
No
Backward?
Yes
distance2 > 20cm ?
Yes
Turn on the buzzer and stop the car for 0.5 seconds
No
Car goes forwards for 0.5 seconds
No
Left?
Yes
Turn left for 0.5 seconds
No
Right?
Yes
Turn right for 0.5 seconds
No
Stop the car and turn off the buzzer

Circuit Schematics



The ultrasonic sensor has 4 pins connected to the Arduino: Vcc, GND, trigger and echo. H-Bridge has 16 pins connected to the Arduino, which altogether control two motors. Each 8 pins control one motor: 2 GND, 1 Vcc, 2 control pins and 2 motor pins. Two control pins will be connected with the Arduino. All the devices share the same Vcc and GND with the Arduino. Ultrasonics sensor send signals to Arduino. The Arduino then sends the signals to H-Bridge and the buzzer. The H-Bridge controls the behavior the motors.

Below is the logic of the H-Bridge



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, non-inverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to circuit board ground plane with multiple solid vias
V_{CC2}	8	—	Power VCC for drivers 4.5V to 36V
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
V_{CC1}	16	—	5V supply for internal logic translation

Enable pins are connected to ground to enable the H-Bridge. 1Y, 2Y and 3Y, 4Y, are connected to two motors. 1A, 2A, 3A and 4A are the control pins. When 1A and 2A (3A and 4A) are both HIGH or LOW, the motor will be off. When 1A and 2A (3A and 4A) are different, the motor will

go in one direction when 1A (4A) is HIGH and 2A (3A) is LOW. The motor will go in another direction when 1A (4A) is LOW and 2A (3A) is HIGH.

Conclusion

Lessons learned

We get to understand the principles of H-bridges, ultrasonic sensors and Bluetooth modules. At the same time, we successfully connect them to our circuit to actually drive our car. We trained our skills of programming using the arduino language and MIT online app making program. In addition, we develop ideas in the designing of the project and improve practical ability via assembling the car and its components.

Self-assessment

We accomplished most goals made in the beginning of the project. However, the automatic parking system is not attached as there is not enough time. There is still improvement for it. The major challenge is the code. The car could not stop when we put the obstacle in front of it when we finished our first version. However, we successfully tackled the problem of the app and codes and make it work eventually. Above all, this is a successful project that still has space for improvement.