Tsang Chun Leung 54802143
Yiu Pui Kwan 54811987

SM2716
Project Documentation –
Two Modes Car

Video Demo

https://youtu.be/TeFk_-BY0SI

Technical Report

1. Description of the system

a. What has been done

We have created a car with two operating modes including automatic detection and remote control via Bluetooth connection. We have written an app for the Bluetooth connection to the car. For the automatic detection mode, the car detects its distance from the surrounding objects. If the distance of them is shorter than 50cm, the car will turn to other direction to avoid it. For the remote control mode, players can operate the car by controlling its moving direction with the app after connecting the smartphone to the car by Bluetooth. The two modes can be selected and changed within the app.

b. How it was built

Tools:

L298n motor controller
HC-05 Bluetooth Serial Module
Arduino UNO
Motor x 2
Ultrasonic Sensor
3.7V battery

Software:

Arduino
Appinventor
Android system

Other:

Wheel x 2
Wooden plate x 2

- Main components -

Arduino, 2 DC Motors, Ping, HC-05 Bluetooth Serial Module, L298N Motor Controller, Jumper Wires, Breadboard, 2 3.7V Batteries.

- Building Method –

Battery:

Providing voltage (12V) to the L298N motor controller for motors power supply by connecting the positive pole of the batteries to the 12V pin on it and the negative pole to the GND pin.

L298N Motor Controller

Controlling motors' speed and direction by connecting the motors to Arduino. Supplying 5V output for powering Arduino with connection of the 5V pin to VIN pin on Arduino and its GND pin to Arduino GND pin.

DC Motor:

2 DC motors for the wheels on 2 sides of the car are connected to the L298N module. The polarity of the motors is the same on both inputs so that the 2 DC motors turns in the same direction. The input pins IN1, IN2, IN3 and IN4 on the L298N driver board are connected to the digital (PWM) pins 3, 5, 6, 9 on Arduino respectively for controlling the turning direction of motors. The motor turning direction is controlled by sending a HIGH or LOW signal to the drive board. To move forward, IN1 and IN4 are HIGH, IN2 and IN3 are LOW so that two motors will turn in one direction. The opposite signals for moving forward is the way the car moves backward. To turn left, only the motor on the right will turn on in one direction, and hence IN4 is HIGH and others are LOW. And to turn right, only motor on the left will turn on in one direction so IN1 is HIGH and others are LOW.

Arduino:

Giving information and conveying commands to the pin for motors, Bluetooth serial module and Ping with their connections. It also controls the 5V voltage supply to them by connecting their GND pins, and its 5V pin to their VCC pins.

Ping:

Detecting the distance of the closest object by sending and receiving the echo wave of ultrasound from the object. Its Trig pin is connected to pin 13 and Echo pin is connected to pin 12 on Arduino. The duration of receiving echo is converted to distance in cm unit. If the object is close enough (<50cm) to the Ping, the motors will be triggered to on or off respectively to adjust the direction and avoid hitting it.

HC-05 Bluetooth serial module:

Inserting the Bluetooth function into the car so that it can access to the phone's Bluetooth connection. With the use of the Bluetooth connection and an app, remote control function can be activated. The TXD on the module is connected to pin 10 (RXD) on the Arduino and the RXD on the module is connected to pin 11 (TXD) on the Arduino.

- Difficulties and resolution -

i.) Finding a suitable app to connect the Bluetooth module.

We wrote an app ourselves.

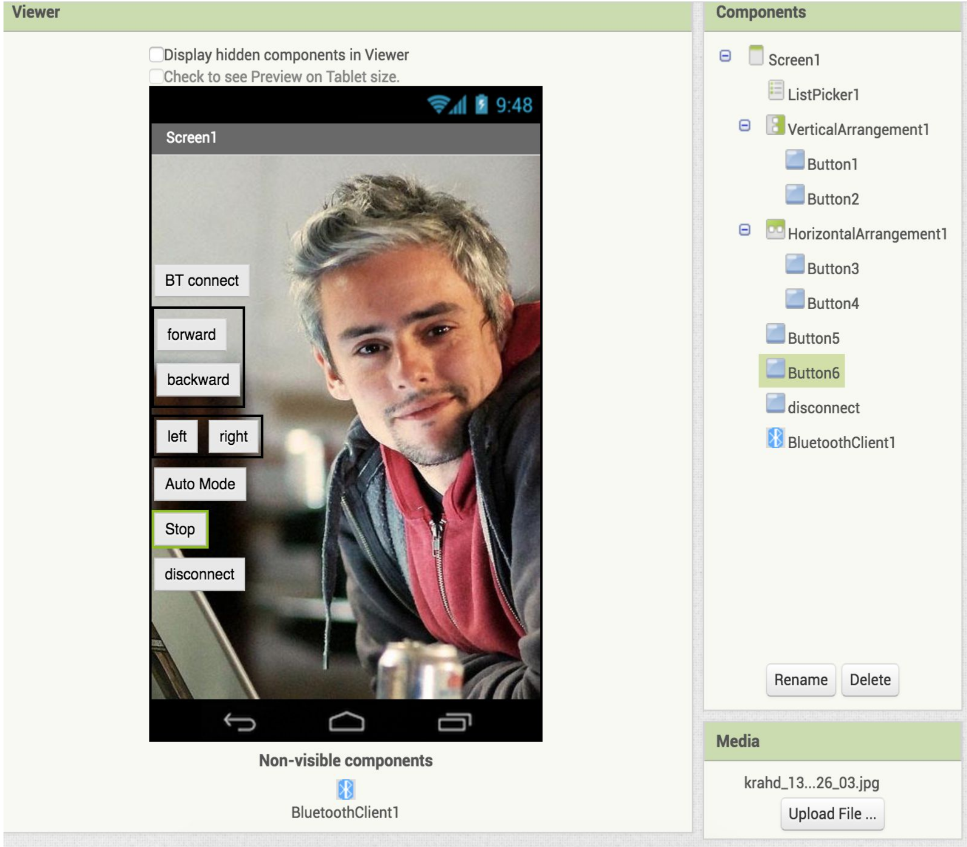
ii.) iPhone cannot connect to the HC-05 Bluetooth serial module.

We borrowed an Android smartphone.

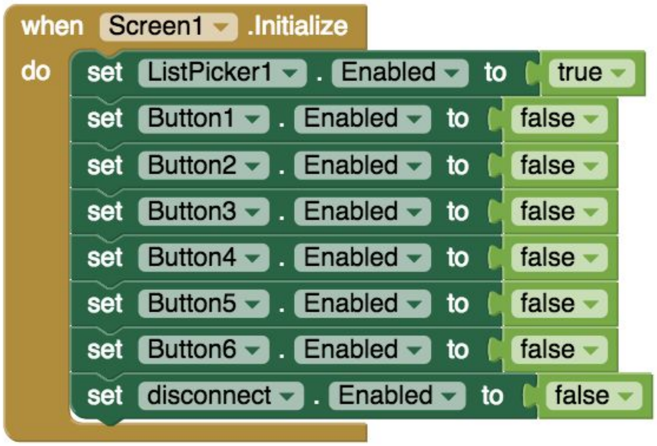
iii.) 9V battery cannot drive the car

We use two 3.7V batteries instead.

- App -

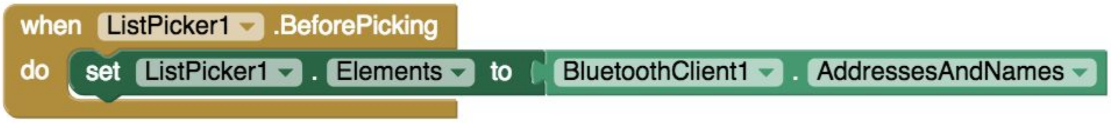


We created an App for controlling the car on http://appinventor.mit.edu/explore



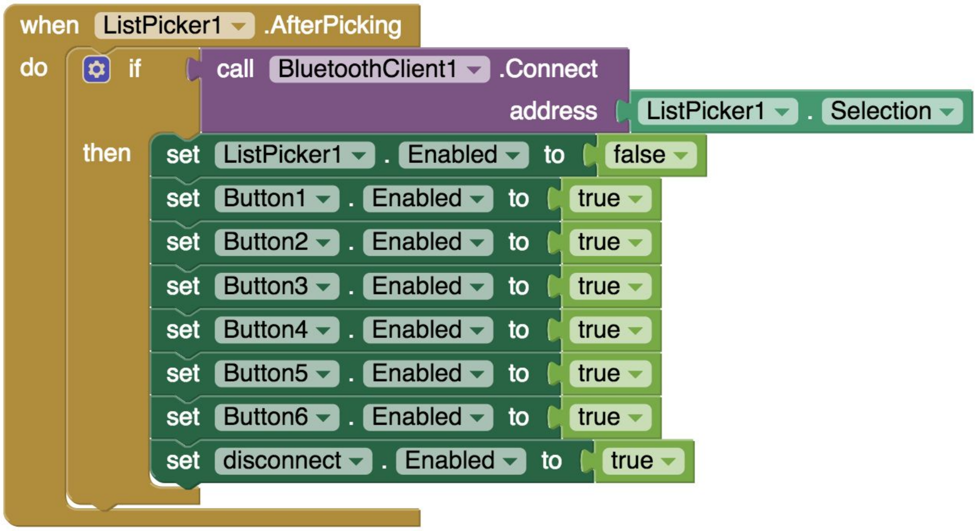
```
when Screen1 .Initialize
do  set ListPicker1 . Enabled to true
    set Button1 . Enabled to false
    set Button2 . Enabled to false
    set Button3 . Enabled to false
    set Button4 . Enabled to false
    set Button5 . Enabled to false
    set Button6 . Enabled to false
    set disconnect . Enabled to false
```

First, Screen1 is initialized. Before connecting to the Bluetooth module of the car, only the “BT connect” button is working.



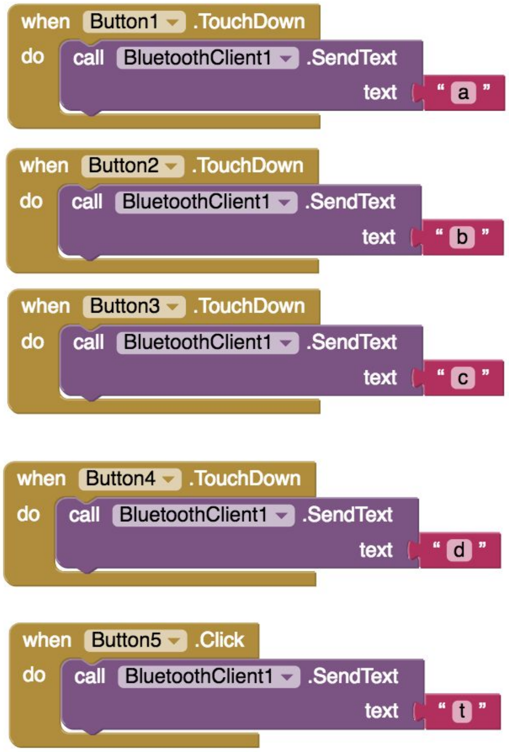
```
when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```

After pressing the “BT connect” button, it searches the Bluetooth devices and user can choose the one for the car.



```
when ListPicker1 .AfterPicking
do  if    call BluetoothClient1 .Connect
                              address ListPicker1 . Selection
    then  set ListPicker1 . Enabled to false
          set Button1 . Enabled to true
          set Button2 . Enabled to true
          set Button3 . Enabled to true
          set Button4 . Enabled to true
          set Button5 . Enabled to true
          set Button6 . Enabled to true
          set disconnect . Enabled to true
```

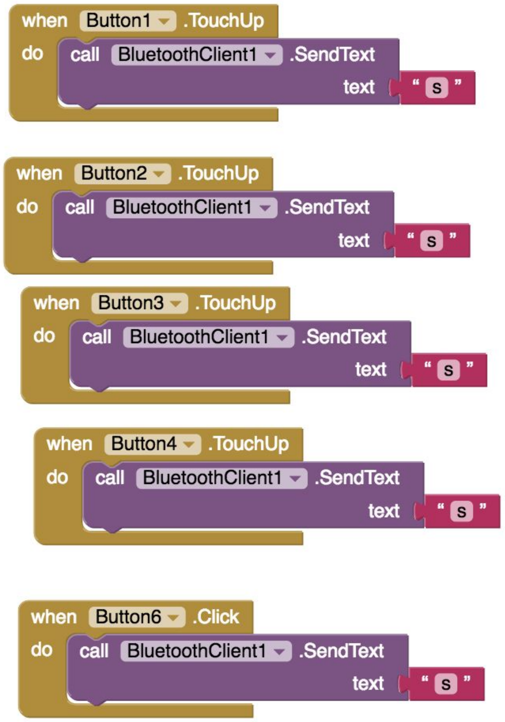
After connecting to the Bluetooth device of the car, “BT connect” button is disabled and other buttons are accessible.



```
when Button1 .TouchDown
do  call BluetoothClient1 .SendText
                           text " a "

when Button2 .TouchDown
do  call BluetoothClient1 .SendText
                           text " b "

when Button3 .TouchDown
do  call BluetoothClient1 .SendText
                           text " c "

when Button4 .TouchDown
do  call BluetoothClient1 .SendText
                           text " d "

when Button5 .Click
do  call BluetoothClient1 .SendText
                           text " t "
```

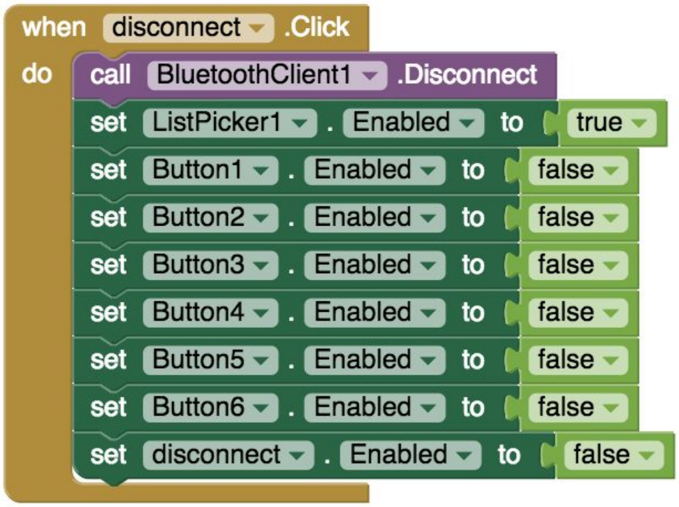
When the user presses on the button for direction, the App will send a character to the Arduino to trigger motors to turn on so that the car moves in that direction.
Button1 “Forward” sends the text “a” which is received as “97” in ASCII format.
Button2 “Backward” sends the text “b” which is received as “98” in ASCII format.
Button3 “Left” sends the text “c” which is received as “99” in ASCII format.
Button4 “Right” sends the text “d” which is received as “100” in ASCII format.

If the user clicks on Button5, text “t” is received as “116” in Arduino and it activates the automatic detection mode so the car moves itself.



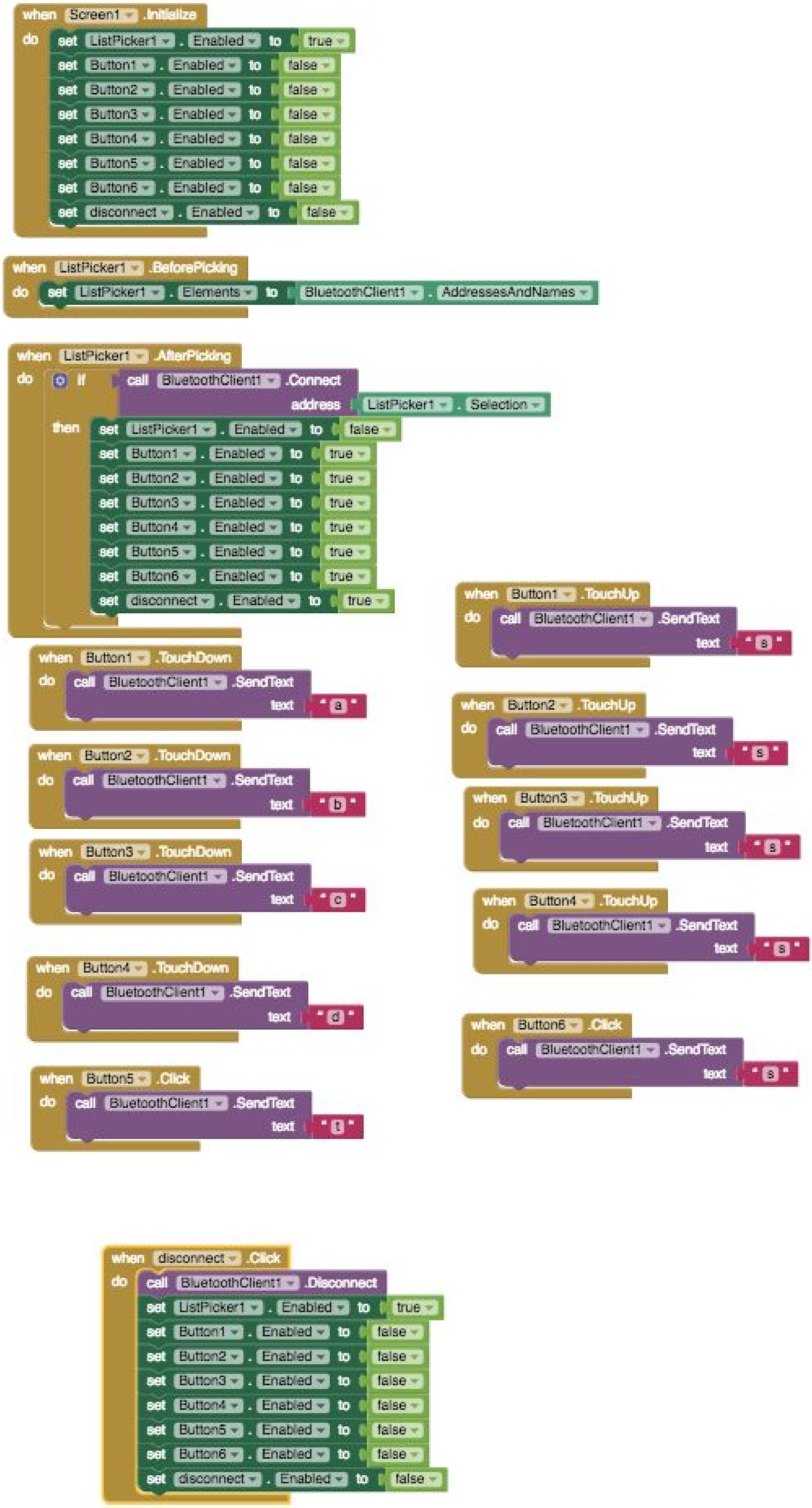
```
when Button1 .TouchUp
do  call BluetoothClient1 .SendText
                           text " s "

when Button2 .TouchUp
do  call BluetoothClient1 .SendText
                           text " s "

when Button3 .TouchUp
do  call BluetoothClient1 .SendText
                           text " s "

when Button4 .TouchUp
do  call BluetoothClient1 .SendText
                           text " s "

when Button6 .Click
do  call BluetoothClient1 .SendText
                           text " s "
```

When the user releases the button for direction in controlling mode or click on “Stop” in automatic detection mode, it sends text “s” (115 in ASCII format) to the Arduino. The motors stop turning so the car stops and quits the automatic detection mode.



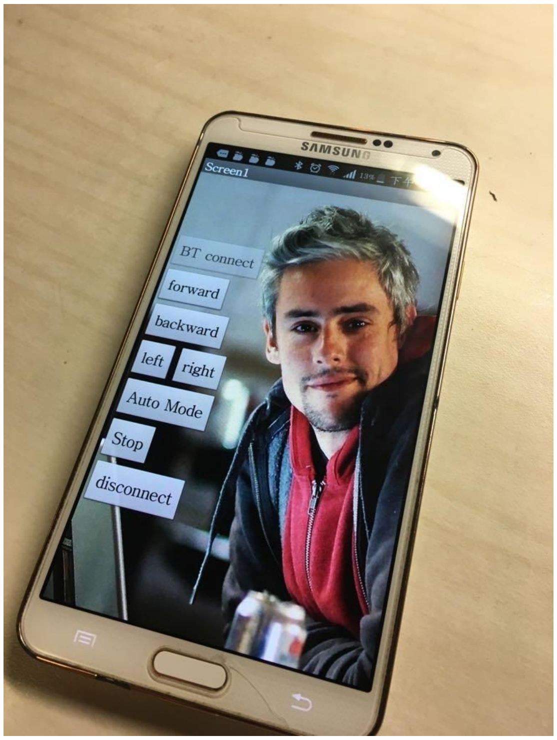
```
when disconnect .Click
do  call BluetoothClient1 .Disconnect
    set ListPicker1 . Enabled to true
    set Button1 . Enabled to false
    set Button2 . Enabled to false
    set Button3 . Enabled to false
    set Button4 . Enabled to false
    set Button5 . Enabled to false
    set Button6 . Enabled to false
    set disconnect . Enabled to false
```

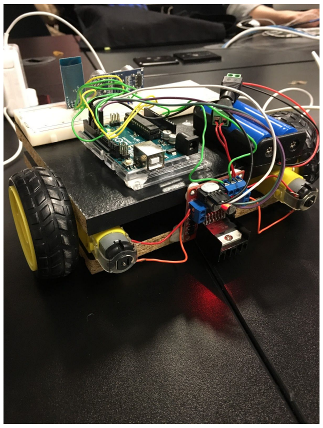
If the button “disconnect” is clicked, Bluetooth connection will be broken off. The operation buttons controlling the car are disabled except “BT connect”. The buttons will be working again when it is connected to the Bluetooth device again.

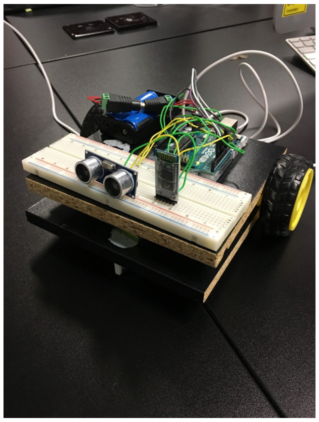


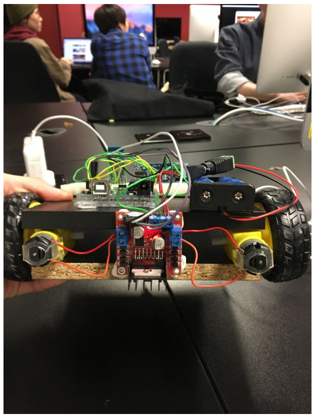
```
when Screen1 .Initialize
do  set ListPicker1 . Enabled to true
    set Button1 . Enabled to false
    set Button2 . Enabled to false
    set Button3 . Enabled to false
    set Button4 . Enabled to false
    set Button5 . Enabled to false
    set Button6 . Enabled to false
    set disconnect . Enabled to false

when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if   call BluetoothClient1 .Connect
                              address ListPicker1 . Selection
    then set ListPicker1 . Enabled to false
         set Button1 . Enabled to true
         set Button2 . Enabled to true
         set Button3 . Enabled to true
         set Button4 . Enabled to true
         set Button5 . Enabled to true
         set Button6 . Enabled to true
         set disconnect . Enabled to true

when Button1 .TouchDown
do  call BluetoothClient1 .SendText
                          text " a "

when Button2 .TouchDown
do  call BluetoothClient1 .SendText
                          text " b "

when Button3 .TouchDown
do  call BluetoothClient1 .SendText
                          text " c "

when Button4 .TouchDown
do  call BluetoothClient1 .SendText
                          text " d "

when Button5 .Click
do  call BluetoothClient1 .SendText
                          text " ! "

when Button1 .TouchUp
do  call BluetoothClient1 .SendText
                          text " s "

when Button2 .TouchUp
do  call BluetoothClient1 .SendText
                          text " s "

when Button3 .TouchUp
do  call BluetoothClient1 .SendText
                          text " s "

when Button4 .TouchUp
do  call BluetoothClient1 .SendText
                          text " s "

when Button6 .Click
do  call BluetoothClient1 .SendText
                          text " s "

when disconnect .Click
do  call BluetoothClient1 .Disconnect
    set ListPicker1 . Enabled to true
    set Button1 . Enabled to false
    set Button2 . Enabled to false
    set Button3 . Enabled to false
    set Button4 . Enabled to false
    set Button5 . Enabled to false
    set Button6 . Enabled to false
    set disconnect . Enabled to false
```

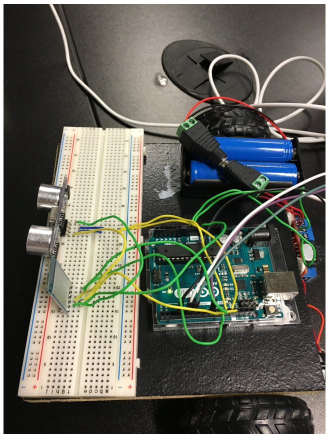
SAMSUNG
下午 11:46
Apps
HK Gov Jobs
政府工通知你
電話
兒童模式
QR碼掃瞄器
WeC
WhatsApp
效能優化精靈
滴滴出行
音樂
蝦米音樂
KMB . LWB
HkdCny
蘋果動新聞
BOCHK
阿斯達克
28Hse
移轉到 iOS
sendawordby bt
blue
Knives Out


SAMSUNG
Screen1
BT connect
forward
backward
left
right
Auto Mode
Stop
disconnect









- Development process -

Week 11:

Researching and collecting tools.

Week 12:

Starting to install the car with motors, motors drive board and ping, and to type in pin control information to Arduino. (Auto detection part)

Week 13:

Installing the car with Bluetooth serial module and type in pin control information to Arduino. (Remote control part) And writing an app for it.

Week 14:

Testing the car and correcting the errors. And filming a video about the product.

c. Why it is interesting

We find the car interesting because of several reasons. Firstly, it is playful and interactive. The car in obstacle avoiding mode interacts with the environment by sensing and avoiding objects around it. The car in Bluetooth controlling mode interacts with humans as they control its direction to move. Players can have fun with it by conducting competition or game. Secondly, there are two modes for selection. Even only one player can compete with other cars with obstacle avoiding mode. Consequently, the players will be less easily bored with it. Also, it can be useful at the same time since it can transport small object from one place to the other place. Moreover, it is a low cost remote control car compared with those sold in the market. It is convenient to play with as almost everyone owns a smartphone.

d. Who has done similar things

In fact, there were many similar Arduino projects done by others before.
They demonstrated the production process which was useful for us and cleared our confusion, helping us a lot for the project.

Examples of multiple modes car:

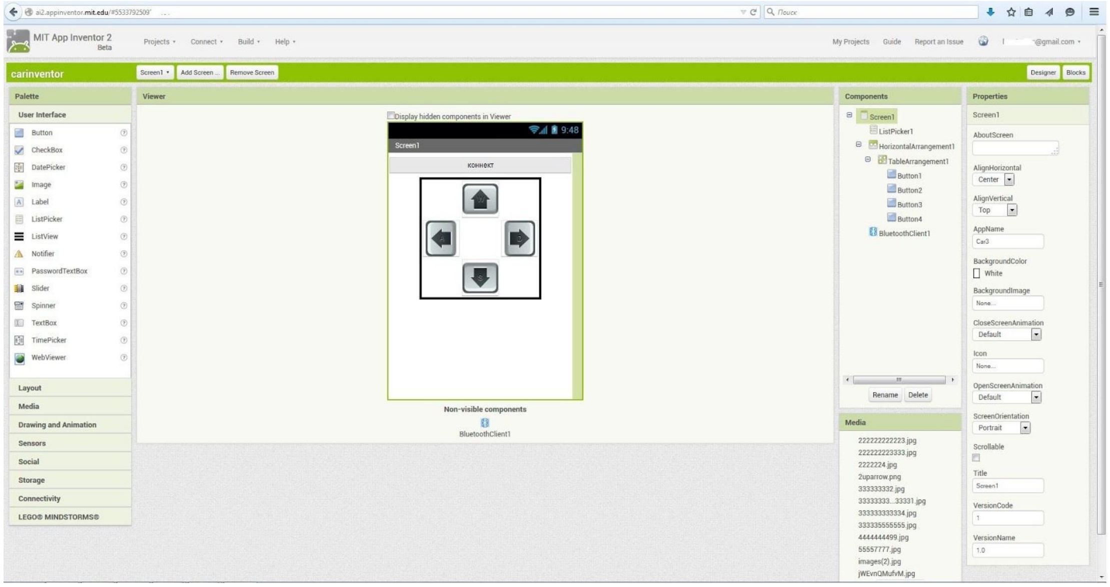
i.) HK Arduino Starter Kit - 4WD Ai Car
http://www.uxbobo.com

ii.) Muhammed Azhar -
Smartphone controlled , obstacle avoiding and wall follower robot.
http://www.robotechmaker.com/2016/07/make-your-first-arduino-robot-best.html

Examples of Bluetooth controlling Arduino with App Inventor:

i.) CAVEDU – App Inventor sending signals to Arduino via Bluetooth http://blog.cavedu.com/programming-language/appinventor/appinventorandarduinowithbluetooth/

ii.) CAVEDU - App Inventor controlling on/off of Arduino LED via Bluetooth http://blog.cavedu.com/programming-language/appinventor/雙a計劃-part1：app-inventor-經由藍牙控制-arduino-led-亮滅/



Screen1

AboutScreen

AlignHorizontal: Center

AlignVertical: Top

AppName: Car3

BackgroundColor: □ White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

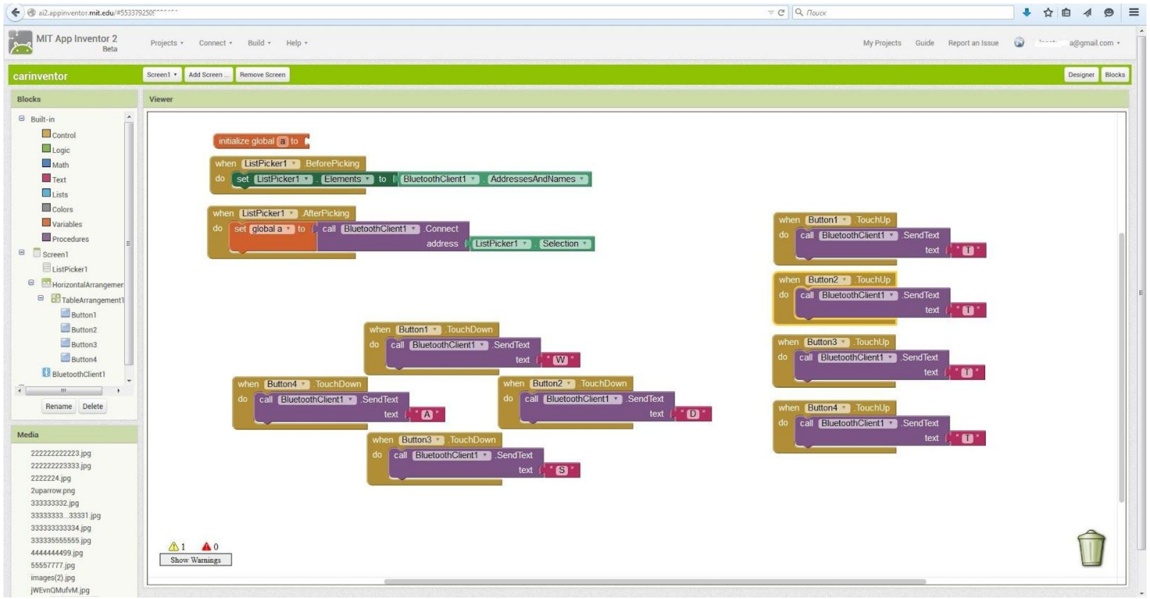
ScreenOrientation: Portrait

Scrollable: □

Title: Screen1

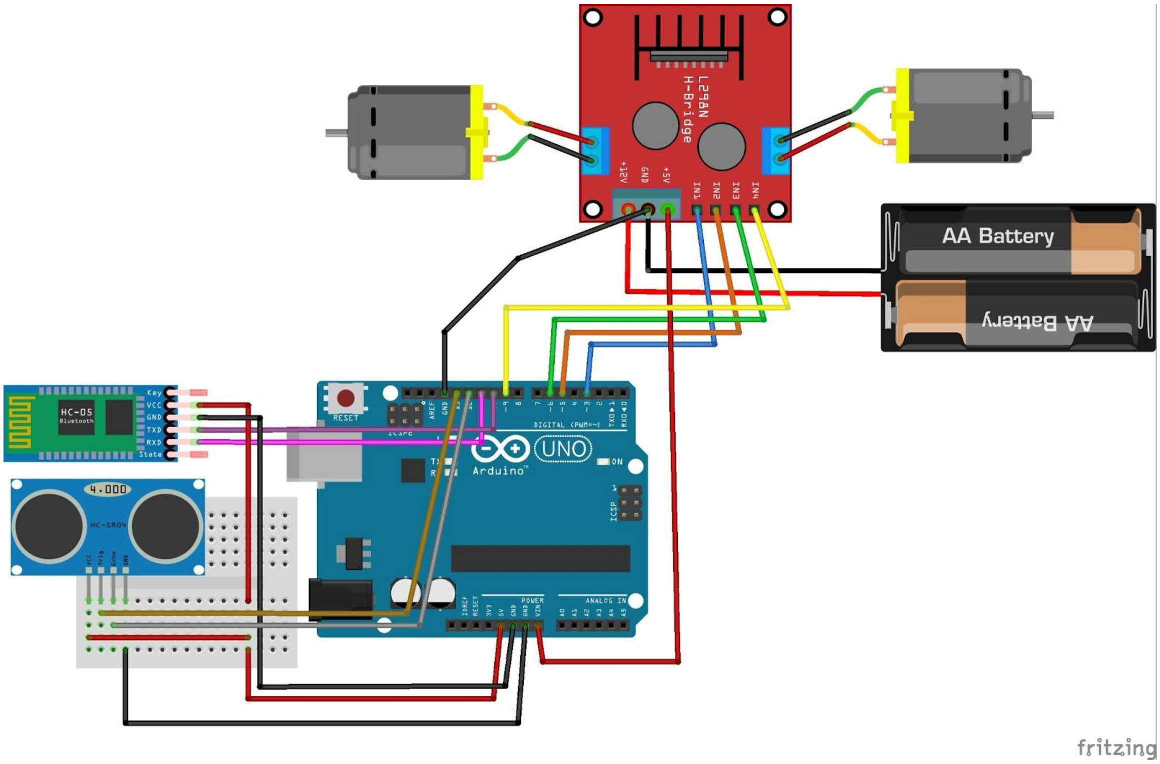
VersionCode: 1

VersionName: 1.0



```
initialize global a to

when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  set global a to call BluetoothClient1 .Connect
                                   address ListPicker1 . Selection

when Button1 .TouchDown
do  call BluetoothClient1 .SendText
                          text " W "

when Button4 .TouchDown
do  call BluetoothClient1 .SendText
                          text " A "

when Button2 .TouchDown
do  call BluetoothClient1 .SendText
                          text " D "

when Button3 .TouchDown
do  call BluetoothClient1 .SendText
                          text " S "

when Button1 .TouchUp
do  call BluetoothClient1 .SendText
                          text " T "

when Button2 .TouchUp
do  call BluetoothClient1 .SendText
                          text " T "

when Button3 .TouchUp
do  call BluetoothClient1 .SendText
                          text " T "

when Button4 .TouchUp
do  call BluetoothClient1 .SendText
                          text " T "
```

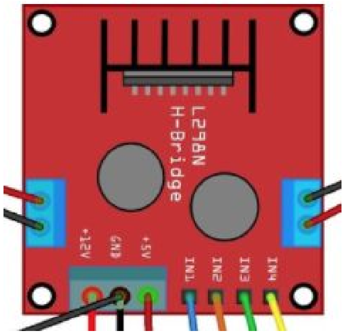
2. Diagram of the electronic circuit



3. Arduino code

```
#include <SoftwareSerial.h>
#include <Wire.h>
int trig = 13;
int echo = 12;
int in1 = 3;
int in2 = 5;
int in3 = 6;
int in4 = 9;
SoftwareSerial I2CBT(10,11); //define Arduino PIN10 and PIN11 as RXD and TXD


void setup() {
  Serial.begin(9600);
  I2CBT.begin(9600);//bluetooth baud rate
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
  byte cmmd[20];
  int insize;
  while(1){
    //read message from bluetooth
    if ((insize=(I2CBT.available()))>0){
      Serial.print("input size = ");
      Serial.println(insize);
      for (int i=0; i<insize; i++){
        Serial.print(cmmd[i]=char(I2CBT.read()));
        Serial.print("\n");
      }
    }

    //actions according to the signals received
    switch (cmmd[0]) {
    case 97:
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,HIGH);
      break;

    case 98:
      digitalWrite(in1,LOW);
      digitalWrite(in2,HIGH);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      break;

      case 99:
      digitalWrite(in1,LOW);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,HIGH);
      break;

      case 100:
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,LOW);
      break;

      case 115:
      digitalWrite(in1,LOW);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,LOW);
      break;

      case 116:
      digitalWrite(trig, LOW);
      delayMicroseconds(2);
      digitalWrite(trig, HIGH);
      delayMicroseconds(5);
      digitalWrite(trig, LOW);
      int duration = pulseIn(echo, HIGH);
      int cm = duration / 29 / 2;
      if(cm>50){
         digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,HIGH);
        } else {
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,LOW);
          }
      break;
```

```
    } //Switch
  } //while
}
```



In the motor controller L298N, 4 pins(int1,int2,int3,int4) responsible for different motor.
Int 1 and 2 are for the left side motor. Int 3 and 4 are for the right side motor.
For instance: If int 1 is High and int 2 is Low. Motor will rotate in one direction.
Oppositely, if int 1 is Low and int 2 is High. Motor will rotate in opposite direction.

Therefore(for example):

```
  case 97:
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,HIGH);
      break;
```

↓

Means moving forward.

```
case 98:
     digitalWrite(in1,LOW);
      digitalWrite(in2,HIGH);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      break;
```

↓

Means moving backward.

4. Future works / things that we would like to improve

We would like to improve the automatic detection part of the car. As our car can only detect the obstacle in front of the car, it would be more awesome if it can detect obstacle from different sides and then adjust its direction. Therefore, we want to set up the Ping which can rotate by 360 degrees with the help of a stepper motor. Since the angle and speed of rotation of stepper motor can be accurately and easily controlled, it is suitable to use it for a Ping sensor to detect surrounding objects around the car in order to avoid hitting them so that the car can adopt a smoother route.

Also, we had planned to make a car with LDR sensor sensing light for its acceleration before. However, we found difficulties to precisely control it and we did not have enough time to complete it. Therefore, for the future works, we would like to finish our original plan as we think that it would be more interesting to play with that version.