Problem Statement:

We chose to build a Bluetooth-controlled Robo-Car model using Arduino that used an android app as its remote controller.
This was a group project of Arkadeep Acharya (2101AI41) and Aritra Bhaduri (2101AI40).

Task Division

As this project has different domains like hardware and software, so we decided to divide our work into two major divisions, and I did the Arduino code and designing the android remote, while the circuit design was done by Aritra. Although we initially started with divided tasks, later, our work overlapped after we had completed the initial prototype designs of circuits and the app.

Approach

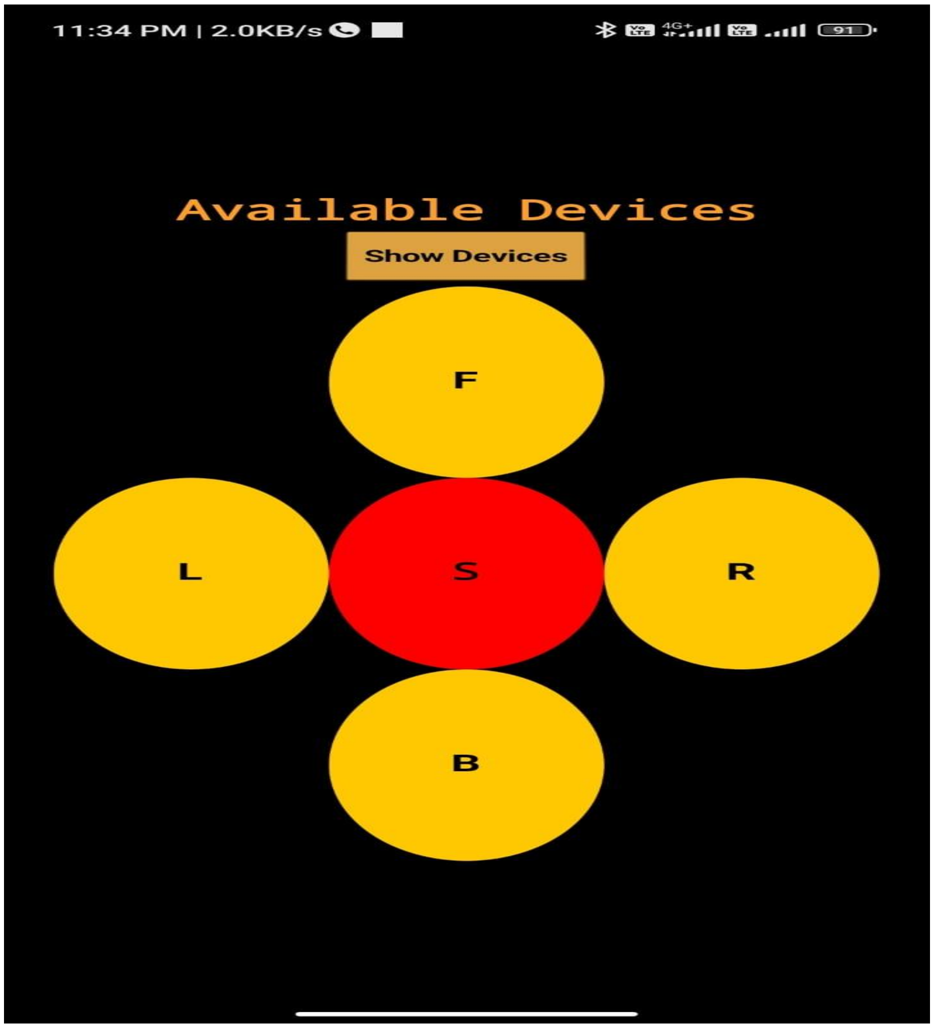
We created the motor driving code for the testing purpose of the circuitry, which we tested by loading into an arduino board and testing it in different ways. After that, I started the code for the HC 05 bluetooth module that we were using in this project. After a few issues, we finally got that bit working. At last I developed the android remote control app using MIT App inventor owing to its simplicity and how easily bluetooth can be integrated in the apps.

Implementation Details

The implementation of a remote controlled car is based on sending different characters to the arduino for different instructions using the bluetooth app, which is then handled by the arduino in a standard switch case format like the example shown below:

```
#include <SoftwareSerial.h>



char t;
```

```
void setup() {
pinMode(5,OUTPUT);   //left motors forward
pinMode(4,OUTPUT);   //left motors reverse
pinMode(3,OUTPUT);   //right motors forward
pinMode(2,OUTPUT);   //right motors reverse
Serial.begin(9600);
digitalWrite(12, HIGH);
SoftwareSerial Serial(10,11);

}

void loop() {

if(t == 'F'){            //move forward(all motors rotate in forward
direction)
  digitalWrite(5,HIGH);
  digitalWrite(3,HIGH);
}

else if(t == 'B'){      //move reverse (all motors rotate in reverse
direction
  digitalWrite(4,HIGH);
  digitalWrite(2,HIGH);
}

else if(t == 'L'){      //turn right (left side motors rotate in
forward direction, right side motors doesn't rotate)
  digitalWrite(3,HIGH);
}

else if(t == 'R'){      //turn left (right side motors rotate in
forward direction, left side motors doesn't rotate)
  digitalWrite(5,HIGH);
}
else if(t == 'S'){      //STOP (all motors stop)
  Serial.println("Stop");
  digitalWrite(5,LOW);
  digitalWrite(4,LOW);
  digitalWrite(3,LOW);
  digitalWrite(2,LOW);
}
delay(100);
}
```

- 

In our case, we have used 5 instructions represented by a set of 5 different characters:

- ‘F’ = Forward: Move the car continuously in the forward direction.
- ‘B’ = Backward: Move the car continuously in the backward direction.
- ‘L’ = Left: Move the car continuously in the left direction.
- ‘R’ = Right: Move the car continuously in the right direction.

- ‘S’ = Stop: Stop the motion of the car

I used the Arduino library softwareserial.h for receiving data via the tx and rx ports of arduino.
I used MIT App Inventor to create a working remote control that sends these signals from the phone to the car via bluetooth. It can connect easily with an already paired HC-05 sensor and can directly send any instruction via the HC-05 module to the arduino.
The .aia file and the .apk and the arduino code has been provided in this drive link.

Problems faced

While coding, we ran into a number of problems which and their solutions are listed:

- Problem uploading code into arduino: Many times we found out that arduino had trouble while uploading code showing error messages like unable to access port, programmer failed to respond. Mostly these were solved either by reconnecting the arduino, or in some unfortunate cases, replacing the entire arduino. In one case we faced this error because we were using the ports 0, 1 while uploading, and when we disconnected those ports it seemed to work again.
- Problem receiving and reading bluetooth signals: A lot of time was spent debugging and trying to fix the process of receiving the signals from bluetooth. Although the issue was mostly due to the circuit, we also had to fix the code as we made some errors in implementing the proper use of softwareserial library.