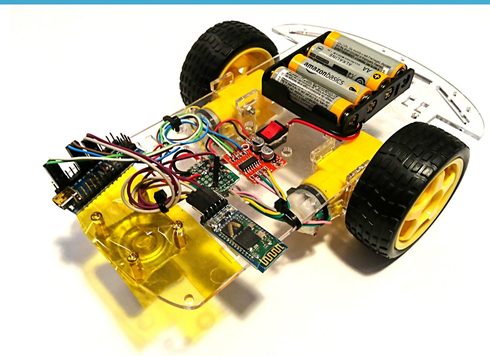
CT3102 - Introduction to Robotics & IoT

Android Controlled Bluetooth Robot Car

GROUP 2: Finn (ID 19010401), Pawel (ID 19024794) and Suzan (ID 19022075)

LONDON METROPOLITAN UNIVERSITY



Introduction

In this Module, CT3102 (Introduction to Robotics and Internet of Things), our group was introduced to basic hardware and software topics with relevance to robotics and the Internet of Things at a Foundational Level. The Case Study for this was to design a Remote Controlled Car of our own design, and produce code to control this car from a mobile device.

The design of the car introduced us to a variety of sensors, such as temperature, anti-collision, line, and infrared sensors electrical components, such as LEDs, diodes, and batteries and motor components such as AC motors, components widely used in modern robotics. It also exposed us to key fundamentals in programming, with particular focus to Arduino and C, the programming languages we used to design the programs needed to remotely control our car.

Aims & Objectives

The primary objective of this case study was twofold; construct a car using various sensory, electrical, and motor components, and using the Arduino programming language to build a program that we could use to remotely control the car from a mobile device.

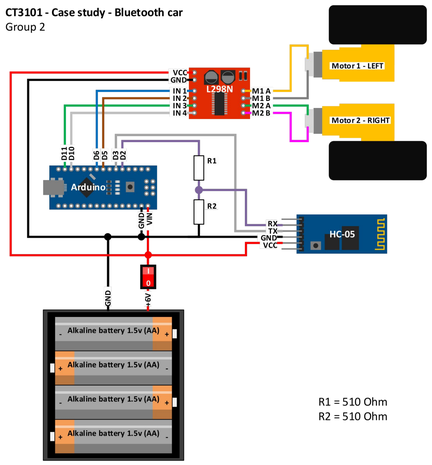
Components Used/Background

In the construction of our car, the following components and materials were used to produce it;

- Arduino Nano Board: used as a brain of the car. It executes program stored on and allows all components to communicate together.
- Motor Driver L298N: used as an interface for the two motors that allow the car to drive.
- Bluetooth module HC-05: used to send and receive Bluetooth communications to and from the car.
- AA Batteries x4: used to power the motors.
- A Switch: used to turn the car on and off.
- Motors w/ wheels x2: used to allow the car to drive, and connected to the Motor Driver.
- Car Base: used as the base for the car, constructed of plastic.
- Rear Wheel: used for additional control for the car's movement.
- Component Board: housed resistor dividers and to split power to all other components of the car.

Hardware Design & Construction

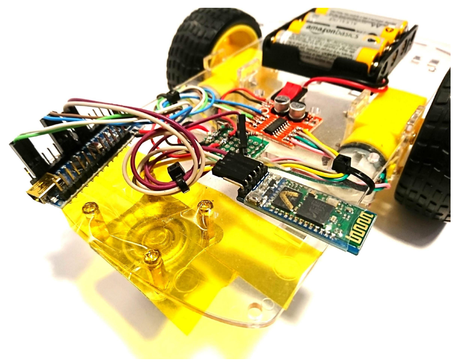
The main element of the car is a Arduino board to which all modules are connected. Whole circuit is powered by 4 batteries each 1.5v connected in series which gives 6v in total. Arduino board is powered through "VIN" pin because it needs 5v to function correctly (VIN input can take up to 15v and on board voltage regulator will output 5v for the microprocessor). Pins D11, D10, D6 and D5 are connected to motor driver. The motor driver board is controlling pair of motors connected to wheels through reduction gears.
Bluetooth module HC-05 has logic at 3.3v so two resistors R1 and R2 for data input were needed to reduce voltage, the HC-05 board can be supplied by 6v so running it directly from batteries works fine. Pins D2 and D3 on Arduino board are connected to HC-05.
Switch is used to turn On or Off whole circuit.



R1 = 510 Ohm
R2 = 510 Ohm

Diagram of a completed circuit

Two resistors (R1 & R2) are soldered on universal PCB located in the middle of the car, the PCB is also used to split power to Bluetooth module, motor driver and Arduino board.
A tape was used to prevent components touching metal screws holding rear wheel and making electric short in the circuit.



Finished build of a car

Software Design & Development

The code we developed to control our car is relatively simple in broadstroke, but it works. The key sections of the program are in the Motor Values, and the transmission of data from the Arduino board to a Bluetooth Receiver.

```
// Defining pins for motors controler (this are not a variables - #define is only used for compiler)
#define MotorA_L 5
#define MotorA_R 6
#define MotorB_L 10
#define MotorB_R 11

// Adding library that will menage serial connection between arduino and HC-05 Bloototh board
#include <SoftwareSerial.h>

// Creating an object 'BTSerial' bor serial connection that can be referenced later
SoftwareSerial BTSerial(3, 2);   // RX | TX


void setup() {
  // Setting pin mode for motors as output
  pinMode(MotorA_L, OUTPUT);
  pinMode(MotorA_R, OUTPUT);
  pinMode(MotorB_L, OUTPUT);
  pinMode(MotorB_R, OUTPUT);

  // Setting board rate for serial object taht will allow to communicate with HC-05 - Bloototh module
  BTSerial.begin(38400); //38400 is default speed for HC-05
}
```

In this section of code, each both Motors in both types of movement (front and back) are being assigned to Pins on the board attached to the car. This board will respond to messages sent to it by the Bluetooth Receiver, telling it which Motors should start and in what direction.

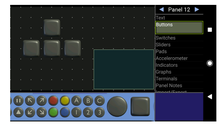
```
void loop() {

  // Checking if there is data stored in buffer...
  if (BTSerial.available() == true){

    // Read character from buffer and save it to variable 'data'
    // this also eraces byte from buffer
    char data = BTSerial.read();

    if(data == 'W'){                  // Forward button pressed on android app
      LeftMotorSpeed(255);
      RightMotorSpeed(255);
    }else if(data == 'w'){    // Forward button released on android app
      LeftMotorSpeed(0);
      RightMotorSpeed(0);

    }else if(data == 'S'){    // Backwards button pressed on android app
      LeftMotorSpeed(-255);
      RightMotorSpeed(-255);
    }else if(data == 's'){    // Backwards button released on android app
      LeftMotorSpeed(0);
      RightMotorSpeed(0);

    }else if(data == 'A'){    // Left button pressed on android app
      //LeftMotorSpeed(-255);  // Line diasbled because car was rotating too fast
      RightMotorSpeed(255);
    }else if(data == 'a'){    // Left button released on android app
      RightMotorSpeed(0);

    }else if(data == 'D'){    // Right button pressed on android app
      LeftMotorSpeed(255);
      //RightMotorSpeed(-255); // Line diasbled because car was rotating too fast
    }else if(data == 'd'){    // Right button released on android app
      LeftMotorSpeed(0);
    }
  }
}
```

In this section of code, the program is checking what data is being sent to the Arduino board. With respect to the prior definitions, the board checks what buttons out of W, A, S, and D are being pressed and makes the motors move either forward or backwards depending on what data is received. Inputting "W" will make both motors move forward, "S" will make both motors move back, "A" will make the left motor move back and the right forward, and "D" will make the left motor move forward and the right backwards.

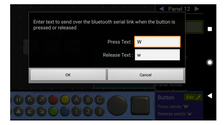
```
// Function for setting speed to left motor
// range [-255, 255], 0 = Stop
void LeftMotorSpeed(int speed) {
  if (speed > 0) {                  // if speed is greater than 0
    analogWrite(MotorA_L, speed);
    digitalWrite(MotorA_R, LOW);
  } else if (speed < 0) {           // if speed is less than 0
    digitalWrite(MotorA_L, LOW);
    analogWrite(MotorA_R, -speed); //the value is negative so we have to invert it (- and - equals +)
  } else {                          //if speed is equal to 0
    digitalWrite(MotorA_L, LOW);
    digitalWrite(MotorA_R, LOW);
  }
}

// Function for setting speed to right motor
// range [-255, 255], 0 = Stop
void RightMotorSpeed(int speed) {
  if (speed > 0) {                  // if speed is greater than 0
    analogWrite(MotorB_L, speed);
    digitalWrite(MotorB_R, LOW);
  } else if (speed < 0) {           // if speed is less than 0
    digitalWrite(MotorB_L, LOW);
    analogWrite(MotorB_R, -speed); //the value is negative so we have to invert it (- and - equals +)
  } else {                          //if speed is equal to 0
    digitalWrite(MotorB_L, LOW);
    digitalWrite(MotorB_R, LOW);
  }
}
```

Android HMI

Screenshots of the function of each element on the Android screen.

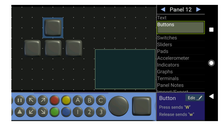


We chose four square buttons

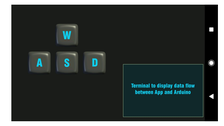


The Bluetooth serial link was program to respond and move forward when 'W' was press.

The data that was transmitted between the App and the Arduino program displayed on the panel.



When the button is press its send capital letters and when the button
is released it sends lower case letters.



W: When 'W' is press the car will move forward at a speed of 255.
S: When 'S' is press the car will move backward.
A: When 'A' is press the car will move to the left.
D: When 'D' is press the car will move to the right.

Conclusion

This case study have allowed us to showcase our talents and the ability to work as a team. The evidence is in the finished product, our robotic car.

We can confidently say that our Robotic Car was able to be controlled remotely. Our car moved forwards, backwards, to the left and to the right by the touch of a button on the Android phone. This was made possible via Bluetooth, Arduino program and other components.

During the assembling and construction of our robotic car, we were exposed to various equipment such as DC motors and how they were connected to enable the movement of our car.

Introduction to Robotics and IOT was interesting, informative and enlightening. Some of the skills that we have learnt in this module we know are transferable, for example, programming.

Acknowledgements

Firstly, we would like to thank our tutor Mr. Onadim for making this module extremely interesting.
We would also like to thank one of our group member Pawel, who made it possible for us to collaborate on our poster remotely. He provided a folder through google drive that gave each member access to contribute to the poster.
Well done to our group for working effectively and remotely as a team.