

Figure 1.1

Figure 1.2

Bluetooth Remote Control Car

CT3102 – Introduction to Robotics & IOT (Submitted 13 MAY 2020)

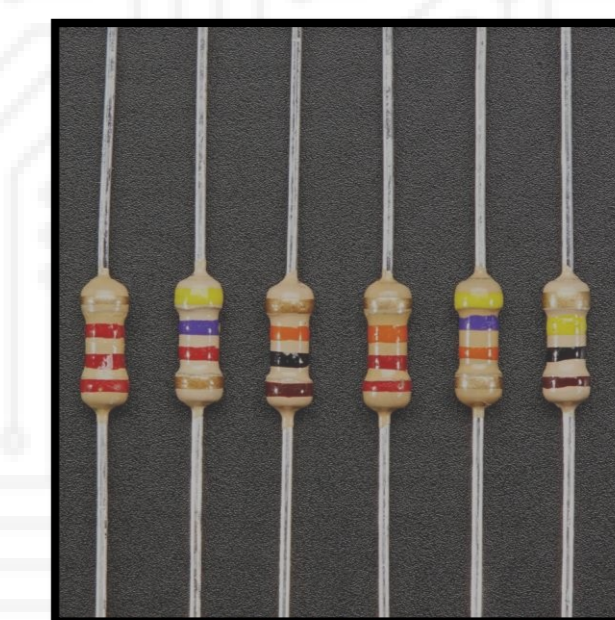


Figure 1.3

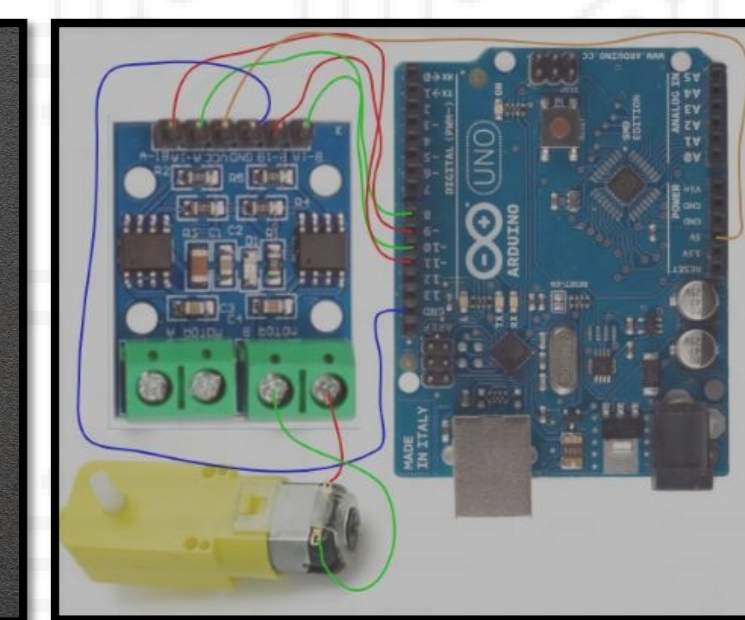


Figure 1.4

Introduction

This project is about building our own remote controlled car. We have been given this case study in order to gain an understanding of the fundamentals of how electronics are made, their functions and how communication occurs.

Starting with simple LEDs and (resistors to stop them burning out); progressing through the various components led to understanding of more complex ones.

We learned that motors must be driven by DC driver modules, that Bluetooth communication can be achieved by use of the HC-05 module & that the Arduino Uno “computer” can be used to tie everything together into what resulted into the fantastic robot car we have built, and of course the C programming language used to give the Uno our precise instructions.

Aims & Objectives

By the end of this projects we aim to have gained understanding of the various components used & code written.

We expect to have our robot cars fully operational & remotely controlled using our smartphone application.

Finally we expect to have a bit of fun too; this way learning is greatly enhanced.

Components Used & Background

Below we’ve made a “component catalogue” of what’s used to achieve our bluetooth robot car. **Note:** Relevant partial descriptions are given.

Hardware Design & Construction

To assemble the car follow these steps.

1. Firstly screw the Arduino Uno onto the chassis along with the L9110s behind it, ensuring to leave space for the 6V PSU. All from the back to front of the car in a linear fashion to ensure good weight distribution.
2. Next screw the motors onto the sides into the matching 3D cut holes. Attach wheels the wire motors to L9110s as shown in figure 1.4 (Top Right ↗)
3. Then we are going to substitute the 4xAA battery pack for a 5V USB power bank as it’s easily recharged & a USB A to B cable can power the Arduino Uno that using the onboard USB Jack.
4. Finally add the wheels and tidy up the cables so they don’t obstruct the wheels at any point whilst moving.
5. Done. That’s it for the hardware design!

Note: Due to temporary university closure & changing study environments we were unable to provide as many diagrams & photos as originally intended. Nevertheless, the hardware build of this robot car was highly engaging and we would recommend that anyone interested in robotics & iot constructs one for themselves.

Acknowledgements

We highly appreciate Mr. Onadim for his expertise in teaching us this module. Curiosity & interest have never come so easy. Thank You.

Software Design & Development

Without code, the robot car would not meet the aim’s & objectives as it would not be able to handle bluetooth data, be controlled by a smartphone or simple move at all.

We used the C/Arduino programming language to achieve desired outcomes. Below is the simplified version of the code.

```
void setup() { //Initial Setup, This Code Will Run Once at Start.
char BLData; // Set Bluetooth data char.
//Below - Set Pins for Motor Output, DV-Driving View
pinMode(9,OUTPUT); //FV, Left Wheel, Green Wire
pinMode(10,OUTPUT); //DV, Left Wheel, Yellow Wire
pinMode(11,OUTPUT); //FV, Right Wheel, Pink Wire
pinMode(12,OUTPUT); //DV, Right Wheel, Grey Wire
Serial.begin(9600); //Start serial communication
}

// We've written 5 working movement functions. Stopcar, Forwardscar,
Backwardscar, ClockwiseRotate, AntiClockwiseRotate. We show 2 of
them below to save space & avoid repetition. Full code in separate file.

void ClockwiseRotate() {
digitalWrite(9,LOW);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(12,HIGH);
}

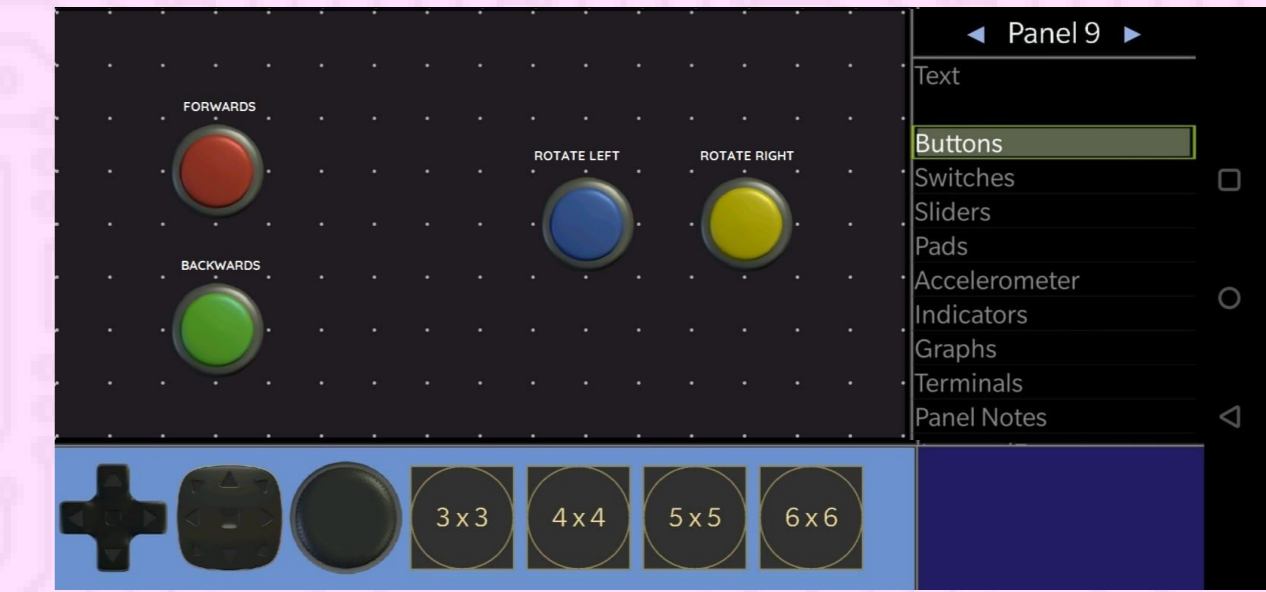
void ForwardsCar() {
digitalWrite(9,HIGH);
digitalWrite(10,HIGH);
digitalWrite(11,HIGH);
digitalWrite(12,HIGH);
}

void loop() { //Program Starts & Loops this section.
if (Serial.available()) //Checks for Bluetooth Serial
{
BLData=Serial.read(); //Reads for bluetooth information.
if (BLData=='A')
forwardsCar();
else if (BLData=='B')
backwardsCar();
else if (BLData=='C')
ClockwiseRotate();
else if (BLData=='D')
AntiClockwiseRotate();
else
stopCar();
}
}
// END OF CODE
```

We programmed our app to send letters A,B,C & D from our smartphones to the HC-05. This information is then read by the serial monitor and as per our code, the various movement functions are executed.

Android Human-Machine Interface

In order to send bluetooth data to our car (specifically HC-05 module) we used a free pre-made application called “Bluetooth Car” By keuwsoft. Shown Below.



This was the most simple and effective design we came up with. To support intended two hand use we placed the forwards and backwards buttons on the far left side and the rotating buttons on the right to correspond with hand placement. This makes it easy for the user to control the car without having to watch the screen carefully. We used colorful buttons to further enhance this experience.

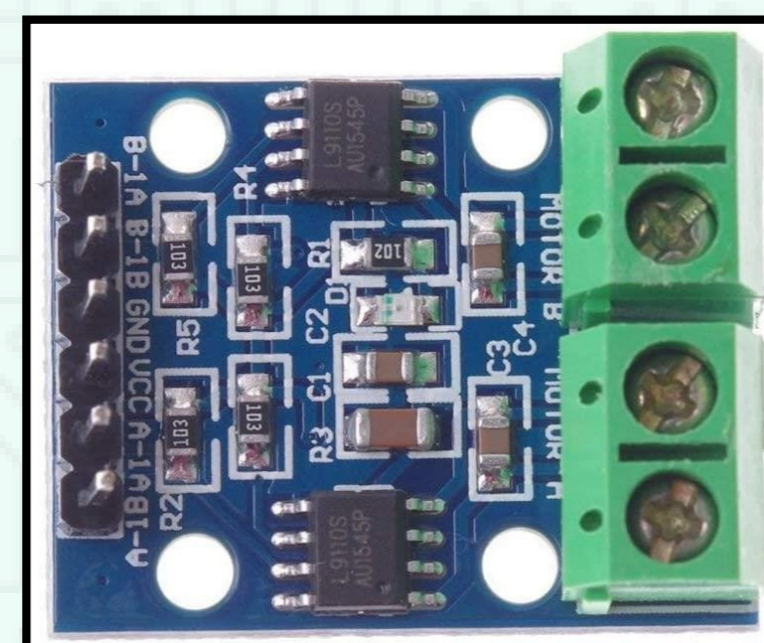
On a technical level, we instructed the apps to send the char “A” whilst the red button is held. In this case, the car moves forwards until the user lets go. This then makes the car stop (else function). The other buttons send B,C & D accordingly to the code on the ← left.

Conclusion

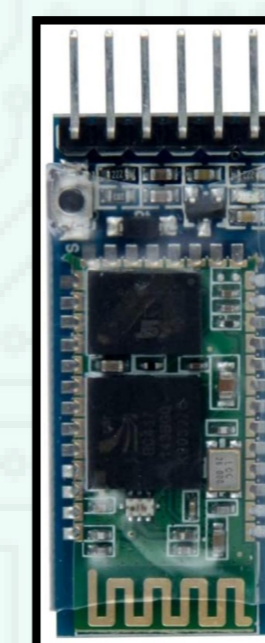
All aims & objectives have been met, even the have fun one. This project was successful and seeing the car move for the first time was highly motivating. We have learned very much about all that was taught and even ventured into side projects like renaming the HC-05 module to “Supercar”. Arduino code looks far more friendly now and we have new practical skills learned whilst putting the hardware together. The two most important things we have learned is that of teamwork & troubleshooting. The social talking loop is more important that the void loop.



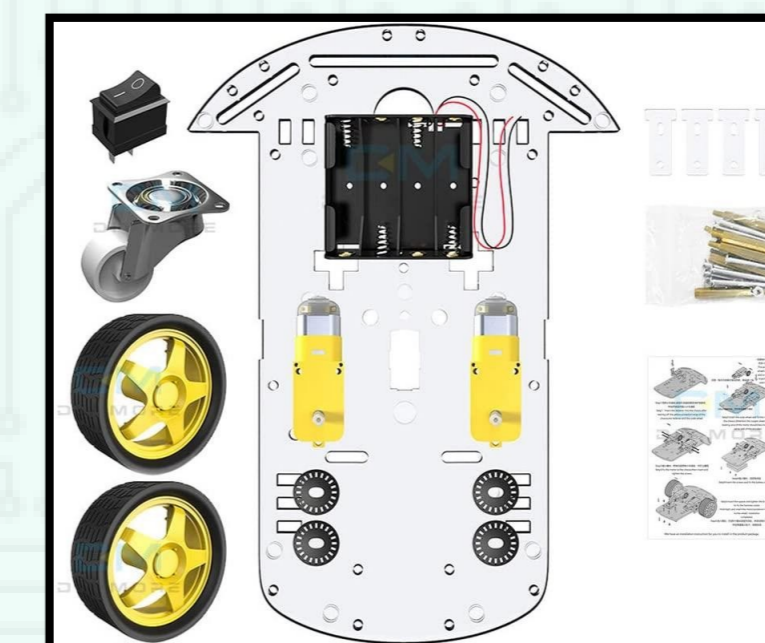
← Arduino Uno is a microcontroller with digital i/o pins that allow data to be processed or passed on to other components.



← L9110s DUAL DC Motor driver. This literally “drives” the motor, or in this case two motors. Apply power and the motor spins.



← HC-05 Bluetooth module. This chip can send and receive wireless data on the bluetooth frequencies. (2.402 to 2.480GHz)



← 2WD Robot Car Chassis. Contains 3D printed plastic parts, rubber wheels, a switch, 4x1.5v AA Batteries (6V) PSU & 2 DC motors. All other components have been screwed into this chassis in appropriate places for good weight distribution.