Shri Vile Parle Kelavani Mandal's

Narsee Monjee College of Commerce & Economics

(Autonomous)

Bachelor of Science (IT) Programme 2022

TYBSC(IT) Semester V Division A

Course Name: Internet of Things

Topic : Arduino Smart Phone Controlled Wi-Fi Car

Submitted by:

Student's Name	SAP ID	Roll No
Sharvari Birajdar	45207200049	A005
Arya Chavan	45207200034	A006
Anjana Kizhiyapat	45207200022	A020

Date of Submission: 16/09/2022

INDEX

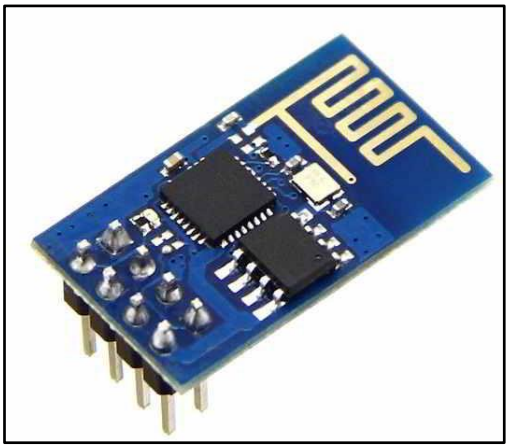
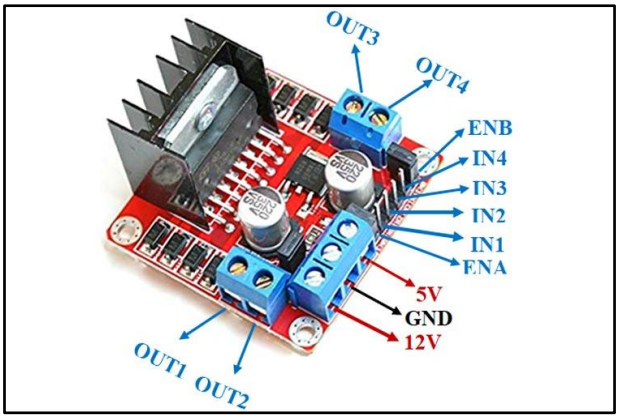
Sr No	Content	Pg no
1.	Introduction	2
2.	Working of the Project • Hardware Components • Circuit diagram • Explanation • Source code • Output	 3-5 5 5-6 6-9 9-10
3.	References	10

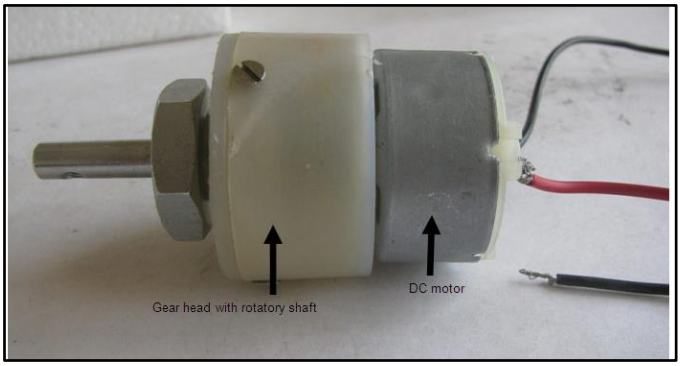
Introduction

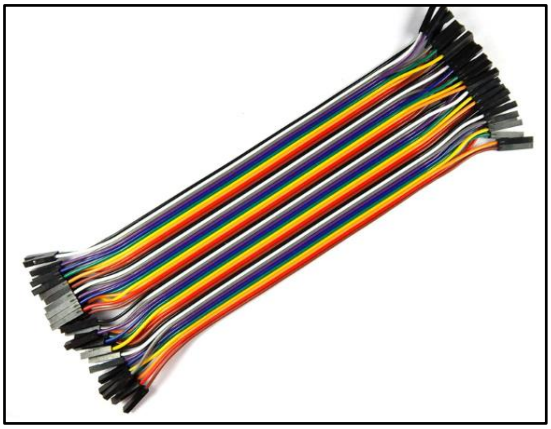
- This documentation gives a detailed explanation of the making and working of a car which supports Wi-Fi module and is operated by using a smartphone. The application used for running the Wi-Fi controlled car is called Remote XY.
- RemoteXY is a mobile application used to control the Arduino and other microcontroller devices from a smartphone or tablet. In other words, it is an online designer of graphical interface to control Arduino via smartphone or tablet.

Working of the Project

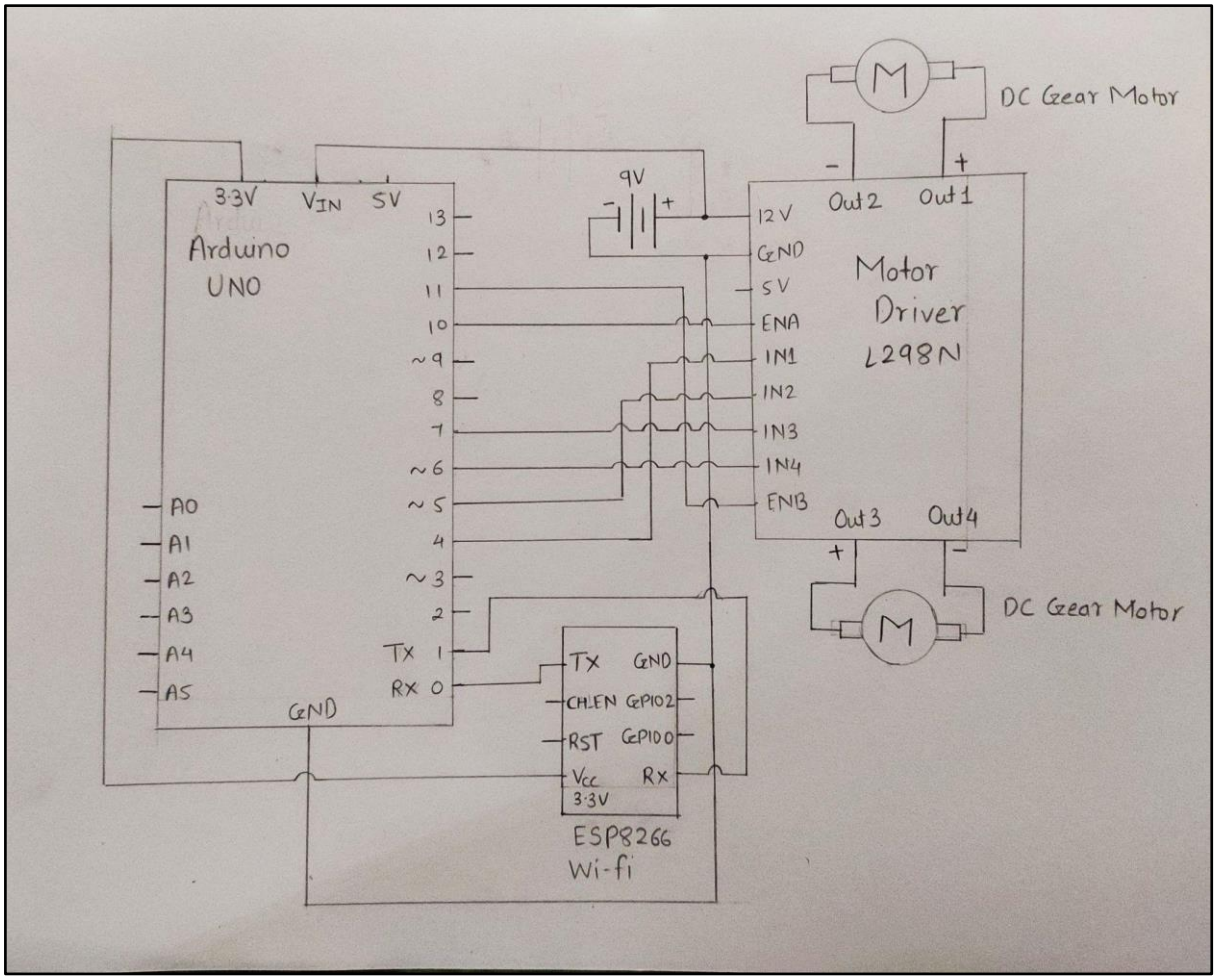
- Hardware Components:

Arduino UNO	
ESP8266 Wifi module	
L298N Motor Driver Module	

DC gear motor	
Robot Wheels	
Caster Wheel	
9V Battery with Battery Clip Connector	

Jumper Wires	

- Circuit diagram:



- Explanation:
  - The model is implemented using an Arduino UNO board along with a Wi-Fi module and a motor driver.

- The 9V battery acts as an external power source (connected to Vin) to the DC gear motor. There are two DC gear motors, each of these are connected to OUT1, OUT2 and OUT3, OUT4 respectively. These motors are attached to a pair of wheels which would be run with the help of the 9V battery.
- When the code is uploaded to the Arduino board, the Wi-Fi module gets connected to the smartphone through the RemoteXY application. One can control the vehicle using the joystick configured by us in the application.

- Source code:

```
1  //////////////////////////////////////////
2  //        RemoteXY include library       //
3  //////////////////////////////////////////
4
5  // RemoteXY select connection mode and include library
6  #define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT
7
8  #include <RemoteXY.h>
9  #include <RemoteXYApi.h>
10 #include <RemoteXYApiData.h>
11 #include <RemoteXYCloudServer.h>
12 #include <RemoteXYComm.h>
13 #include <RemoteXYComm_AT.h>
14 #include <RemoteXYComm_ESP8266.h>
15 #include <RemoteXYComm_Ethernet.h>
16 #include <RemoteXYComm_WiFi.h>
17 #include <RemoteXYConnection.h>
18 #include <RemoteXYConnectionCloud.h>
19 #include <RemoteXYConnectionServer.h>
20 #include <RemoteXYConnectionStream.h>
21 #include <RemoteXYDebugLog.h>
22 #include <RemoteXYFunc.h>
23 #include <RemoteXYStream.h>
24 #include <RemoteXYStream_BLEDevice.h>
25 #include <RemoteXYStream_BLEPeripheral.h>
26 #include <RemoteXYStream_BluetoothSerial.h>
27 #include <RemoteXYStream_CDCSerial.h>
28 #include <RemoteXYStream_HardSerial.h>
29 #include <RemoteXYStream_SoftSerial.h>
30 #include <RemoteXYThread.h>
31 #include <RemoteXYWire.h>
```

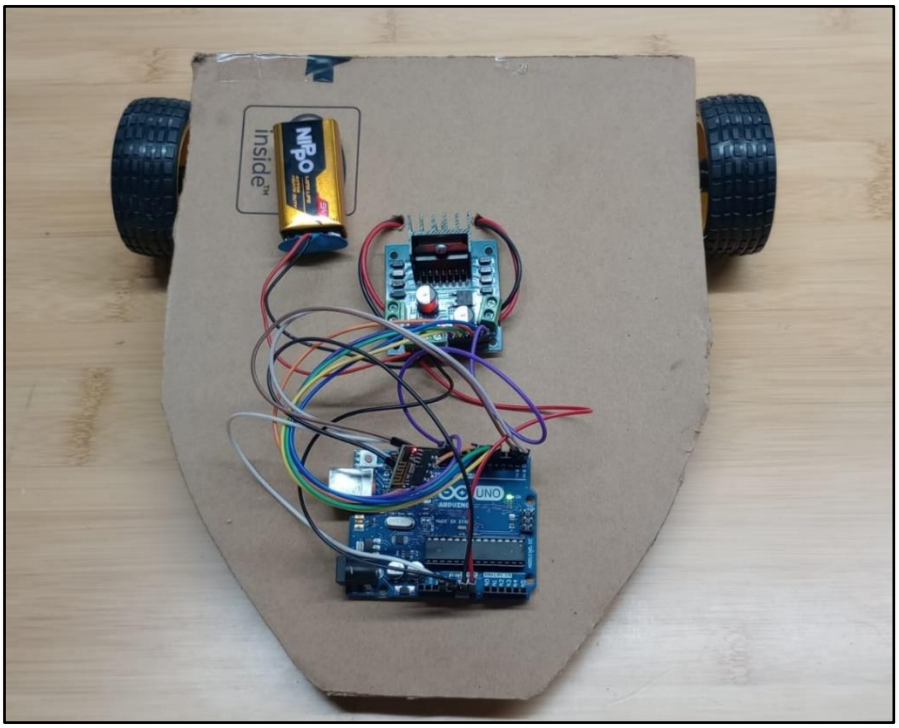
```
32  #include <RemoteXYWireCloud.h>
33  #include <RemoteXYWireStream.h>
34  // RemoteXY connection settings
35  #define REMOTEXY_SERIAL Serial
36  #define REMOTEXY_SERIAL_SPEED 115200
37  #define REMOTEXY_WIFI_SSID "RemoteXY"
38  #define REMOTEXY_WIFI_PASSWORD "12345678"
39  #define REMOTEXY_SERVER_PORT 6377
40
41
42  // RemoteXY configurate
43  #pragma pack(push, 1)
44  uint8_t RemoteXY_CONF[] =   // 19 bytes
45    { 255,2,0,0,0,12,0,16,31,0,5,54,32,17,30,30,2,26,31 };
46
47  // this structure defines all the variables and events of your control
48  interface
49  struct {
50
51      // input variables
52    int8_t joystick_1_x; // from -100 to 100
53    int8_t joystick_1_y; // from -100 to 100
54
55      // other variable
56    uint8_t connect_flag;  // =1 if wire connected, else =0
57
58  } RemoteXY;
59  #pragma pack(pop)
60
61  /////////////////////////////////////////
62  //          END RemoteXY include          //
63  /////////////////////////////////////////
64
65  //define MOTOR Controlling pins
66  #define robot_LEFT_motor1 4 // in1
67  #define robot_LEFT_motor2 5 // in2
68  #define LEFT_motor_speed 10
69
70  //define RIGHT Motor pins
71  #define robot_RIGHT_motor1 6 // in4
72  #define robot_RIGHT_motor2 7 // in3
73  #define RIGHT_motor_speed 11 // enable controls the dc motor
```

```
74
75  //define two arrays with list of pins each other
76
77  uint8_t Right_Motor[3] =
78  {robot_RIGHT_motor1,robot_RIGHT_motor2,RIGHT_motor_speed};
79  uint8_t LEFT_Motor [3]=
80  {robot_LEFT_motor1,robot_LEFT_motor2,LEFT_motor_speed };
81
82  //Speed control of motors
83  void Wheel (uint8_t * motor,int v)// v=motor speed control=pointer
84  to an array of pins
85  {
86    if (v > 100) v=100;
87    if (v <-100) v=100;
88    if (v > 0) // move forward
89    {
90      digitalWrite (motor [0],HIGH);
91      digitalWrite (motor [1],LOW);
92      analogWrite (motor [2],v * 2.55);
93
94    }
95    else if(v<0) // reverse
96    {
97      digitalWrite (motor [0],LOW);
98      digitalWrite (motor [1],HIGH);
99      analogWrite (motor [2], (-v) * 2.55); // -v cuz it is going in reverse
100
101
102   }
103 else // halt or stop
104 {
105    digitalWrite (motor [0],LOW);
106     digitalWrite (motor [1],LOW);
107     analogWrite (motor [2],0);
108
109 }
110 }
111
112 void setup()
113 {
114   RemoteXY_Init ();
115
```

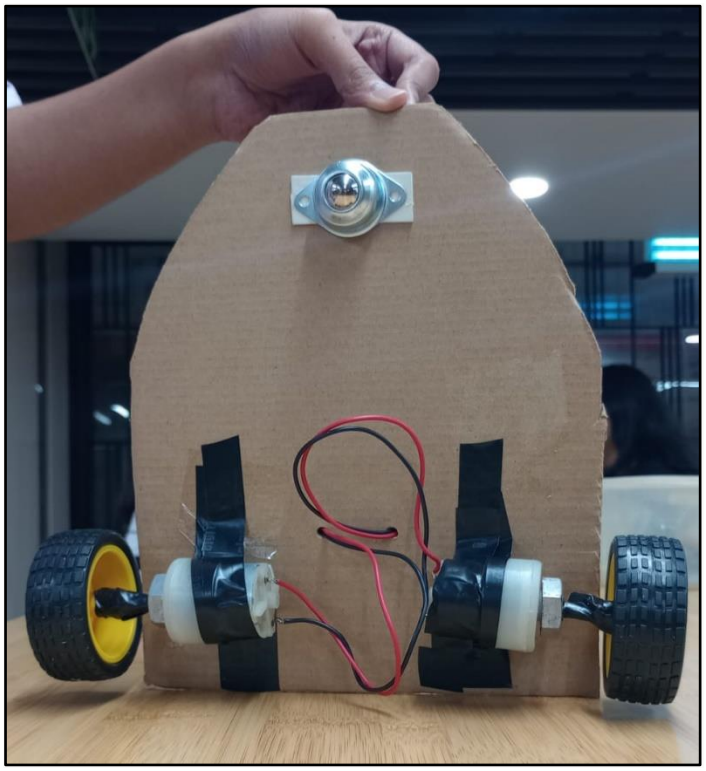
```
116
117   //initialization pins
118 pinMode (robot_LEFT_motor1,OUTPUT);
119 pinMode (robot_LEFT_motor2,OUTPUT);
120 pinMode (robot_RIGHT_motor1,OUTPUT);
121 pinMode (robot_RIGHT_motor2,OUTPUT);
122
123 }
124
125 void loop()
126 {
127   RemoteXY_Handler ();
128
129
130   //manage the motor
131  Wheel (Right_Motor, RemoteXY.joystick_1_y -
132 RemoteXY.joystick_1_x);
133  Wheel (LEFT_Motor, RemoteXY.joystick_1_y +
134 RemoteXY.joystick_1_x);
135 }
```

- Output:

[Front-Side]



[Back-Side]



[Joystick designed using RemoteXY]

References

- Final Year Project Journey
- How to Make DIY Arduino Obstacles Avoiding Robot with IR Sensor without ULTRASONIC Sensor.
- How To Make WiFi Controlled Robot Car Using esp8266_01 module || RemoteXY Controlled WIFI Robot
- How to make Mobile Bluetooth controlled robot car. Electronic Arduino project.