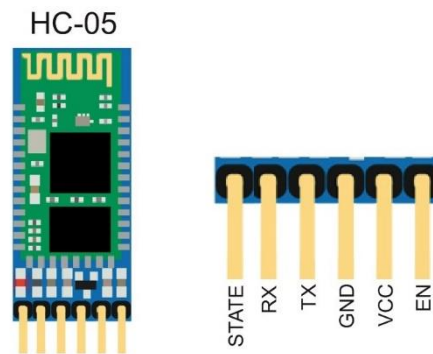


BLUETOOTH CAR CONTROLLED WITH ANDROID APP

Bluetooth Shield for Arduino UNO

The main reason for popularity of Arduino UNO platform is that it's easy, cheap, open source and has a lot of resources on internet. So to start with we will use the most commonly used Bluetooth shield for Arduino HC-05. See the image below.



Connect PINs as:

Arduino Pins	HC 05 Pins
VCC	3.3V/5V (As on Module)
GND	GND
RX	TX
TX	RX

DESCRIPTION

We just need to wire module with Arduino as above. Now this module uses a protocol known as Serial Communication. It is same mode as used in our

USB's. In this mode we send and receive data over two wires. You can see RX and TX pin. In simple words:

RX means **Receiver**

TX means **Transfer**

So we connect these opposite. As RX of Module is connected to TX of Arduino and TX of Module is connected to RX of Arduino. So Arduino Send using TX module receive using RX and module send using TX and Arduino receive using RX. So in this way whole communication task place.

So as a whole an Android phone connects to Bluetooth module and send data to it. Where Arduino is working in serial and reads that data in serial. Where we parse the data and use it for specific purpose

Default Settings for HC - 05

The default settings for new modules are







- Name = HC-05
- Password = 1234
- Baud rate in communication mode = 9600

(Android meets Arduino workshop)

Circuit:

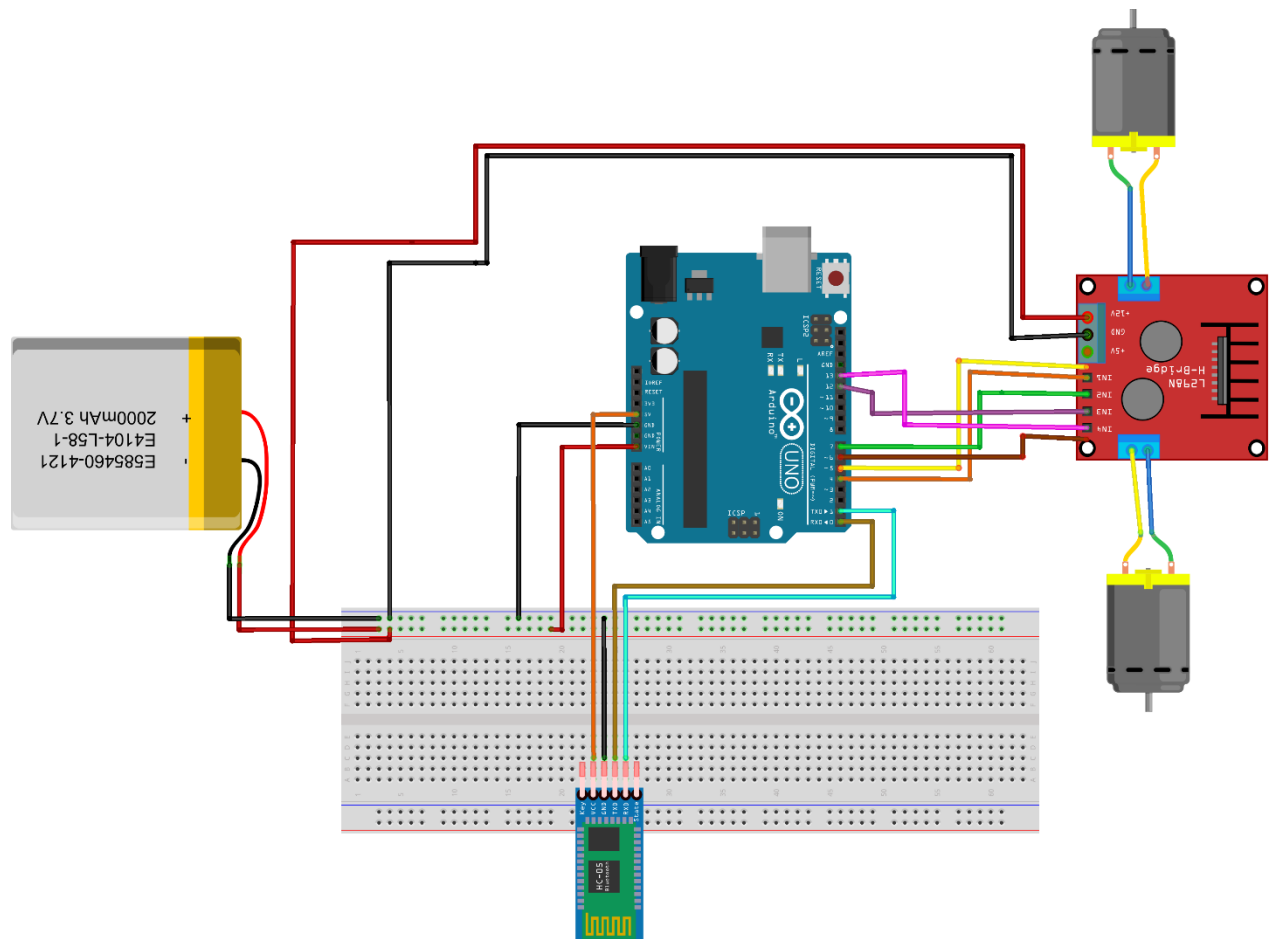
In this implementation, we will connect the motor driver with the wheels and then connect the wires with Arduino. An external battery is connected in order to turn on the motor driver. HC-05 Bluetooth module has four pins. It's RX and TX pins will be connected with TX and RX pins of Arduino respectively. HC-05 needs 5V to turn on which will be supplied from the Arduino.

COMPONENTS:

-  H-Bridge Motor Driver
-  Arduino UNO
-  HC-05
-  Car set with wheels
-  Breadboard
-  Wires

Android Mobile with App

Here is the schematic diagram of the project that you need to recreate.



fritzing

Here is the code for this project:

```
//Initialize Motor 1  
int motor1Pin1 = 4;  
int motor1Pin2 = 7;  
int enable1Pin = 5;
```

```
//Initialize Motor 2  
int motor2Pin1 = 12;  
int motor2Pin2 = 13;  
int enable2Pin = 6;
```

```

int state;
void setup() {
  // sets the pins as outputs:

  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(enable2Pin, OUTPUT);
  pinMode(led,OUTPUT);
  digitalWrite(led,HIGH);
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  //if some data is sent, reads it and saves in state
  if(Serial.available())
  {
    state = Serial.read();
  }
  delay(100);
  // if the state is '1' the DC motor will go forward

  if (state == 1) // forward
  {
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    analogWrite(enable1Pin, 180);
    analogWrite(enable2Pin, 180);
    Serial.println("Forward");
  }

  else if (state == 2) // left
  {
    //turn left
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, HIGH);
    analogWrite(enable1Pin, 250);
    analogWrite(enable2Pin, 250);

    delay(150);

    state = 5;
    Serial.println("left");
  }
}

```

```

else if (state == 5) // Stop
{
    digitalWrite(enable1Pin, 0);
    digitalWrite(enable2Pin, 0);
    delay(300);
    Serial.println("stop");
}

else if (state == 3) // right
{
    //TURN RIGHT
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    analogWrite(enable1Pin, 250);
    analogWrite(enable2Pin, 250);

    delay(150);

    state = 5;

    Serial.println("right");
}

else if (state == 4) //reverse
{

    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, HIGH);
    analogWrite(enable1Pin, 180);
    analogWrite(enable2Pin, 180);
    delay(300);
    Serial.println("reverse");
}

}

```

Future Additions:

 Lights will be controlled with Android App.