

VIT -AP university

WIRELESS GESTURE CONTROLLED CAR

by:

* K.Sai Pranay[17BCD7021]
* P.Manoj Kumar[17BEV7012]
* N.Rushikesh[17BCD7063]
* K.Prathyusha[17MIS7035]
* G.Vamsidhar Reddy[17MIS7021]

GUIDE: PROF.JOHN PRADEEP

INDEX

1.     Introduction

2.    Background

3.     Problem Definition

4.     Objective

5.    Methodology/Procedure

6.    Results and Discussion

7.    Conclusion and Future Scope

8.    References

9.    Codes in Appendix

INTRODUCTION:

\*\*\*\*How can we control the car?\*\*\*\*

|  |  |
| --- | --- |
| Previous methodology | Our methodology |
| 1.joystick controlled car  2.keyboard controlled car  3.phone controlled car  4.voice controlled car | Wireless gesture  controlled car |

\*\*\*\*What we are going to do with this idea?\*\*\*\*

Now a day’s wireless communication is a hot topic in industry. so, we are also moving to wireless.we can control car using wireless communication with the help of sensor called accelerometer.

\*\*\*\*what is the mode of communication?\*\*\*\*

In this project we are using serial communication using Bluetooth.

\*\*\*\*General example in which this project was helpful?\*\*\*\*

Suppose,if there is any paralysis patient who needs some support for everything can use this system. Accelerometer which we are using is very sensitive and can measure the motion very precisely. The person can use this model, so that he can do some basic works like moving from one place to another place without any support.

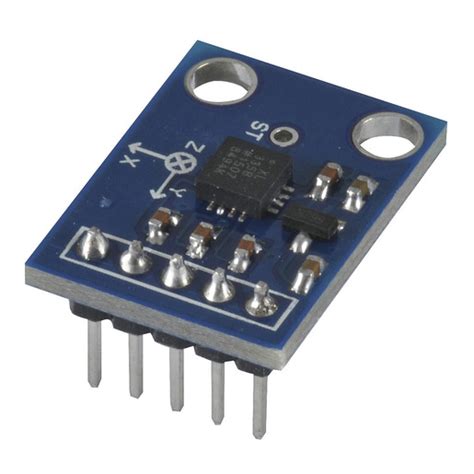
Because of using wireless transmission he can get rid of so many wires, so that it is very easy to use. We can observe the motion through video which we have implemented already in this project.

BACKGROUND

Components:

|  |  |
| --- | --- |
| component | Quantity |
| 3 axis accelerometer | 1 |
| Arduino(UNO) | 2 |
| L298D Motor drivers | 2 |
| Camera module(Logitech) | 1 |
| Raspberry PI(model-3B) | 1 |
| HC 05 bluetooth module | 2 |

3 axis accelerometer:

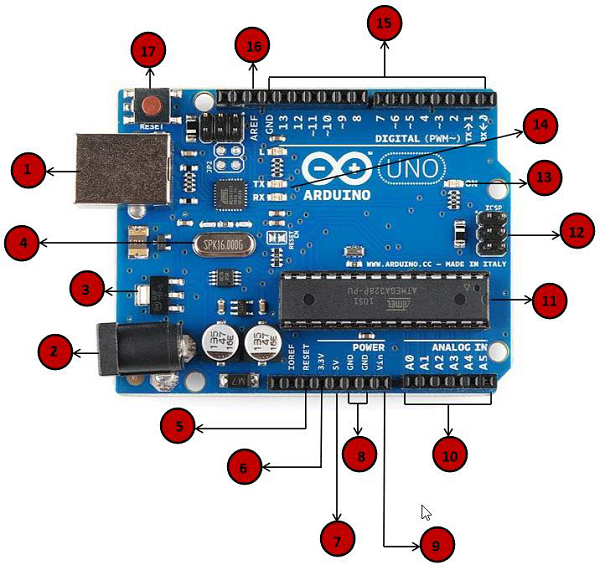


An accelerometer is a device that measures [properacceleration](https://en.wikipedia.org/wiki/Proper_acceleration).[Proper acceleration, being the acceleration (or rate of change of [velocity](https://en.wikipedia.org/wiki/Velocity)) of a body in its own instantaneous [rest frame](https://en.wikipedia.org/wiki/Rest_frame),is not the same as coordinate acceleration, being the acceleration in a fixed coordinate system. For example, an accelerometer at rest on the surface of the Earth will measure an [acceleration due to Earth's gravity](https://en.wikipedia.org/wiki/Gravitational_acceleration), straight upwards (by definition) of g ≈ 9.81 m/s2. By contrast, accelerometers in [free fall](https://en.wikipedia.org/wiki/Free_fall) (falling toward the center of the Earth at a rate of about 9.81 m/s2) will measure zero.

The 3-Axis Accelerometer consists of three –5 to +5 g accelerometers mounted in one small block. Using the appropriate data collection hardware and software, you can graph any of these components, or calculate the magnitude of the net acceleration. The 3-Axis Accelerometer can be used for a wide variety of experiments and demonstrations, both inside the lab and outside.

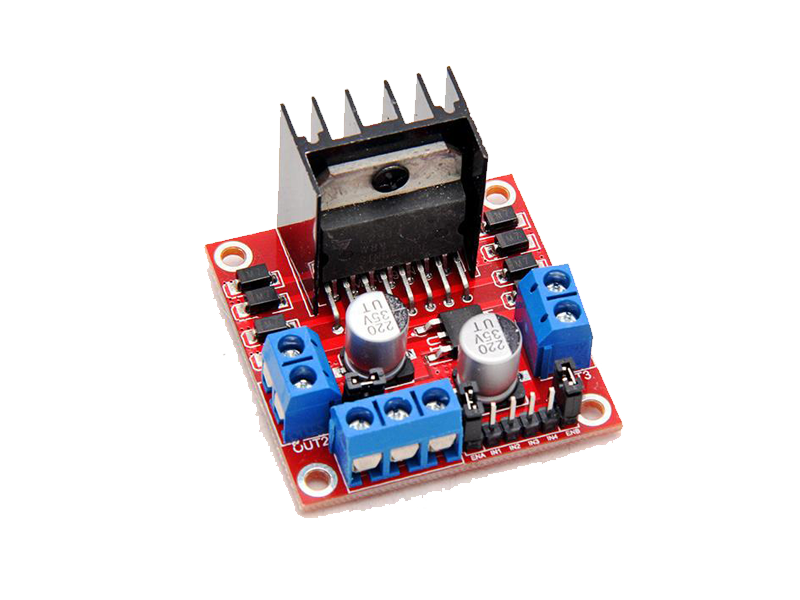
Arduino (UNO):

Arduino UNO board is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



|  |  |
| --- | --- |
| Power USB | **Power USB**  Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1). |
| Barrel Jack | **Power (Barrel Jack)**  Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2). |
| Voltage Regulator | **Voltage Regulator**  The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements. |
| Crystal Oscillator | **Crystal Oscillator**  The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz. |
| Arduino Reset | **Arduino Reset**  You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5). |
| Pins | **Pins (3.3, 5, GND, Vin)**   * 3.3V (6) − Supply 3.3 output volt * 5V (7) − Supply 5 output volt * Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. * GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. * Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply. |
| Analog pins | **Analog pins**  The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor. |
| Main microcontroller | **Main microcontroller**  Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet. |
| ICSP pin | **ICSP pin**  Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus. |
| Power LED indicator | **Power LED indicator**  This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection. |
| TX and RX LEDs | **TX and RX LEDs**  On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process. |
| Digital I/O | **Digital I/O**  The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM. |
| AREF | **AREF**  AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. |

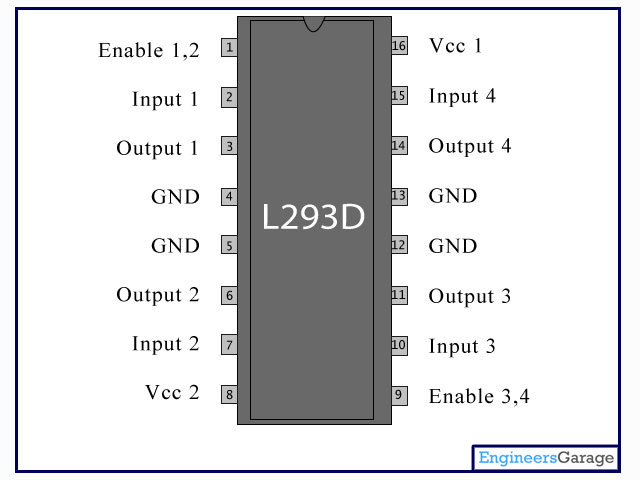
L298D Motor drivers:



L293D is a dual [H-bridge](http://www.engineersgarage.com/electronic-circuits/h-bridge-motor-control) motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.



**Pin Description:**

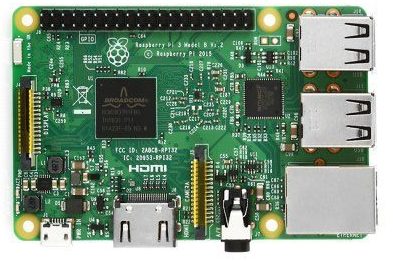
|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Enable pin for Motor 1; active high | Enable 1,2 |
| 2 | Input 1 for Motor 1 | Input 1 |
| 3 | Output 1 for Motor 1 | Output 1 |
| 4 | Ground (0V) | Ground |
| 5 | Ground (0V) | Ground |
| 6 | Output 2 for Motor 1 | Output 2 |
| 7 | Input 2 for Motor 1 | Input 2 |
| 8 | Supply voltage for Motors; 9-12V (up to 36V) | Vcc 2 |
| 9 | Enable pin for Motor 2; active high | Enable 3,4 |
| 10 | Input 1 for Motor 1 | Input 3 |
| 11 | Output 1 for Motor 1 | Output 3 |
| 12 | Ground (0V) | Ground |
| 13 | Ground (0V) | Ground |
| 14 | Output 2 for Motor 1 | Output 4 |
| 15 | Input2 for Motor 1 | Input 4 |
| 16 | Supply voltage; 5V (up to 36V) | Vcc 1 |

Camera module(Logitech):



* Web Camera specifically designed and optimized for Professional quality video streaming on social gaming and entertainment sites like Twitch and YouTube
* Stream and record vibrant, true-to-life HD 1080P video at 30 frames per second or 720P at 60Fps
* Background replacement technology (powered by personify) allows you to integrate your live image and any background Scene, right onto your stream (only works with Windows 7 and above)
* Full HD glass lens and premium autofocus deliver razor-sharp, clear video in consistent high Definition while two Built-in mics capture your voice in rich Stereo Audio
* Record clear videos even in dim or poorly backlit settings with automatic light Correction

Raspberry PI(model-3B):



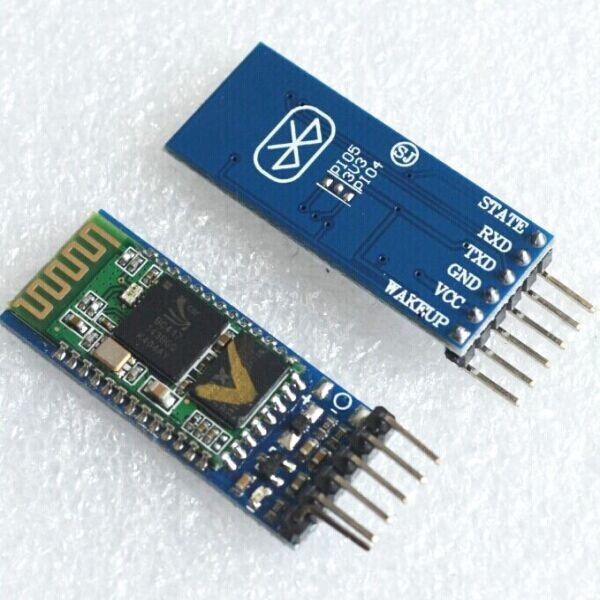
The **Raspberry Pi** is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation) to promote teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries).[[5]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-6)[[7]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-7) The original model became far more popular than anticipated,[[8]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-1000x-8) selling outside its [target market](https://en.wikipedia.org/wiki/Target_market) for uses such as [robotics](https://en.wikipedia.org/wiki/Robotics). It does not include peripherals (such as [keyboards](https://en.wikipedia.org/wiki/Keyboard_(computing)) and [mice](https://en.wikipedia.org/wiki/Mouse_(computing))) and [cases](https://en.wikipedia.org/wiki/Computer_case). However, some accessories have been included in several official and unofficial bundles.[[8]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-1000x-8)

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation). After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with [Eben Upton](https://en.wikipedia.org/wiki/Eben_Upton) as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

## Features:

* Broadcom BCM2837 chipset running at 1.2 GHz
* 64-bit quad-core ARM Cortex-A53
* 802.11 b/g/n Wireless LAN
* Bluetooth 4.1 (Classic & Low Energy)
* Dual core Videocore IV® Multimedia co-processor
* 1 GB LPDDR2 memory
* Supports all the latest ARM GNU/Linux distributions and Windows 10 IoT
* microUSB connector for 2.5 A power supply
* 1 x 10/100 Ethernet port
* 1 x HDMI video/audio connector
* 1 x RCA video/audio connector
* 4 x USB 2.0 ports
* 40 GPIO pins
* Chip antenna
* DSI display connector
* microSD card slot
* Dimensions: 85 x 56 x 17 mm

HC 05 bluetooth module



**Description:**

The HC-05 Bluetooth Module has 6 pins- Vcc, GND, TX, RX, Key, and LED. It comes pre-programmed as a slave, so there is no need to connect the Key pin, unless you need it change it to Master Mode.

The major difference between Master and Slave modes is that, in Slave mode the Bluetooth module cannot initiate a connection, it can however accept incoming connections. After the connection is established the Bluetooth module can transmit and receive data regardless of the mode it is running in. If you are using a phone to connect to the Bluetooth module, you can simply use it in the Slave mode. The default data transmission rate is 9600kbps.

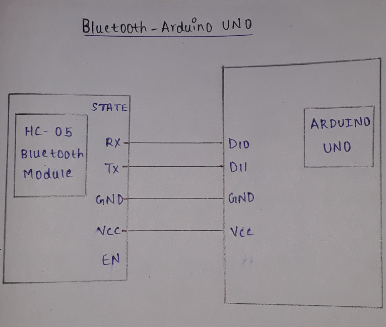
The range for Bluetooth communication is usually 30m or less. The module has a factory set pin of “1234” which is used while pairing the module to a phone.

**HC-05 Specification:**

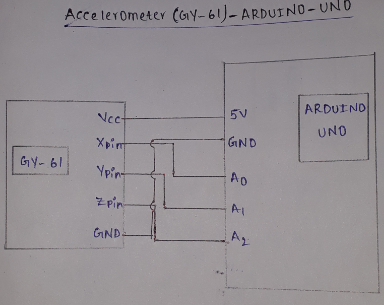
* Bluetooth protocal: Bluetooth Specification v2.0+EDR
* Frequency: 2.4GHz ISM band
* Modulation: GFSK(Gaussian Frequency Shift Keying)
* Emission power: ≤4dBm, Class 2
* Sensitivity: ≤-84dBm at 0.1% BER
* Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
* Security: Authentication and encryption
* Profiles: Bluetooth serial port
* Power supply: +3.3VDC 50mA
* Working temperature: -20 ~ +75Centigrade
* Dimension: 26.9mm x 13mm x 2.2 mm

METHODOLOGY:

1. Connect the 2-bluetooth to 2-arduino’s.



1. Connect the accelerometer to one of the arduino and connect the motor to another one.

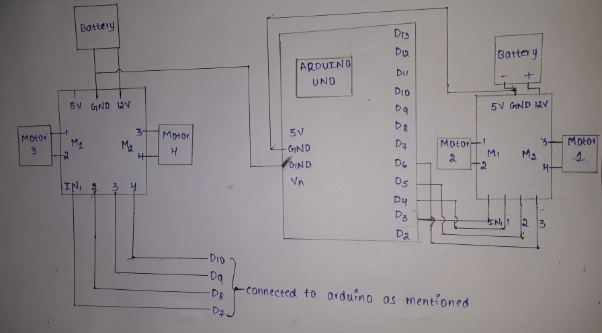


1. Make one of the Bluetooth module to slave and other to master using AT commands.

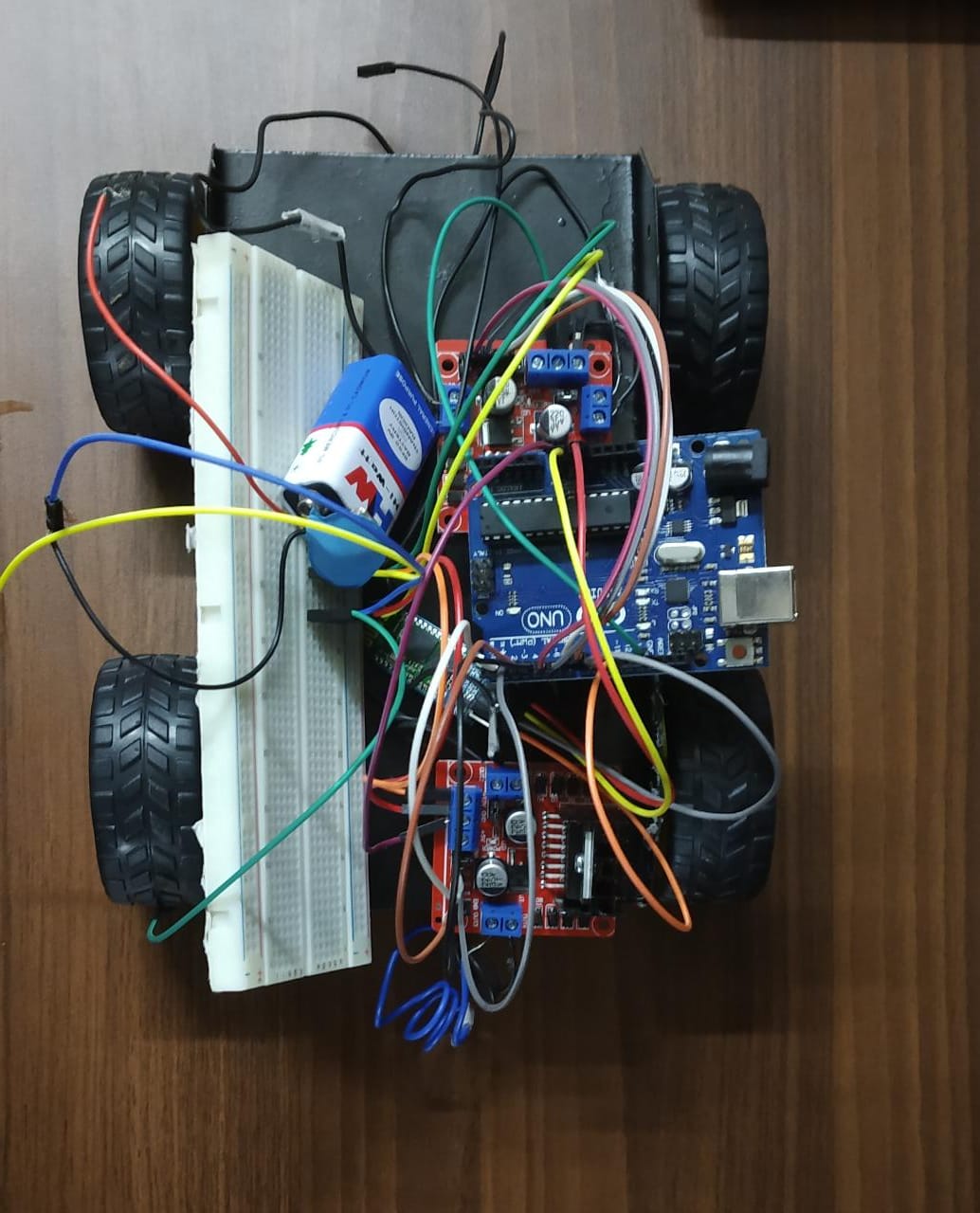
Here are some of the AT commands:

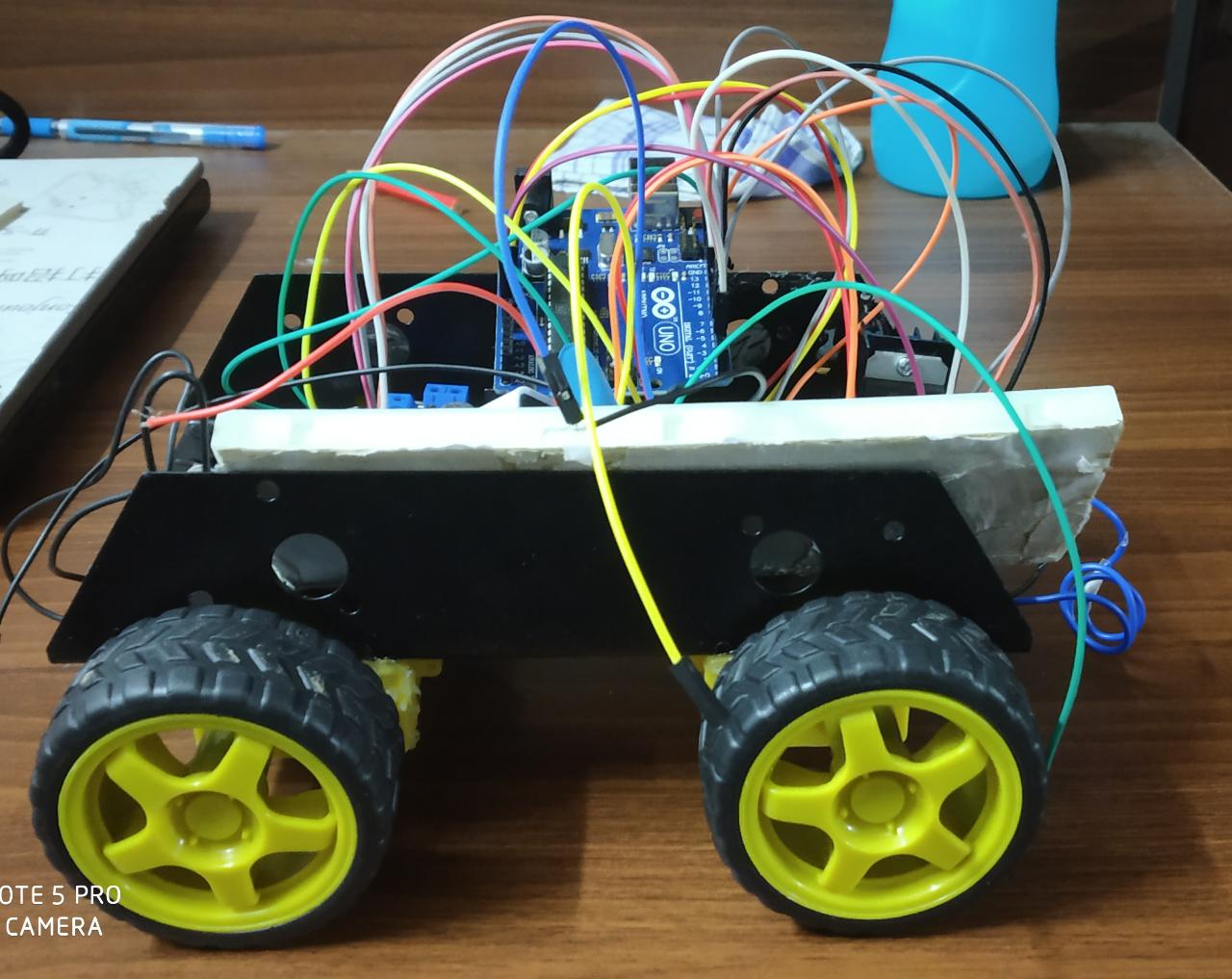
AT : Ceck the connection.  
AT+NAME : See default name  
AT+ADDR : see default address  
AT+VERSION : See version  
AT+UART : See baudrate  
AT+ROLE: See role of bt module(1=master/0=slave)  
AT+RESET : Reset and exit AT mode  
AT+ORGL : Restore factory settings  
AT+PSWD: see default password

1. Connect the motor drivers and motors to the arduino which is used as a slave.



Result:





Appendix

CODE FOR MASTER AND CAR:

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

int xpin = A3; // x-axis of the accelerometer

int ypin = A2; // y-axis

int zpin = A1; // z-axis (only on 3-axis models)

int xval;

int yval;

int zval;

void setup() {

Serial.begin(9600);

BTSerial.begin(38400);

}

void loop() {

Serial.print(analogRead(xpin));

Serial.print("\t");

Serial.print(analogRead(ypin));

Serial.print("\t");

Serial.print(analogRead(zpin));

Serial.println();

xval=analogRead(xpin);

yval=analogRead(ypin);

zval=analogRead(zpin);

if( ((xval>=300 && xval<=350)) && ((yval>=305) && (yval<=335))&&((zval>=385)&&(zval<=415))) //stationary or stop(transmitter parallel to ground)

{

Serial.println("stop");

BTSerial.write("stop");

digitalWrite(13,LOW);

}

else if ((xval>=310 && xval<=340) && (yval>=255 && yval<=285)&&(zval>=350&&zval<=385)) //forward(transmitter tilted forward)

{

Serial.println("forward");

BTSerial.write("forward");

digitalWrite(13,HIGH);

}

else if ((xval>=310 && xval<=340) && (yval>=370 && yval<=400)&&(zval>=355&&zval<=385)) //backward(transmitter tilted backward)

{

Serial.println("backward");

BTSerial.write("backward");

digitalWrite(13,HIGH);

}

else if ((xval>=375 && xval<=405) && (yval>=315 && yval<=355)&&(zval>=350&&zval<=380)) //left(transmitter tilted to left)

{

Serial.println("left");

BTSerial.write("left");

digitalWrite(13,HIGH);

}

else if ((xval>=250 && xval<=280) && (yval>=330 && yval<=350)&&(zval>=340&&zval<=390))//right(transmitter tilted to right)

{

Serial.println("right");

BTSerial.write("right");

digitalWrite(13,HIGH);

}

delay(100);

}

CODE FOR ACCELEROMETER AND ARDUINO:

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

String readdata;

void setup()

{

pinMode(13, OUTPUT);

Serial.begin(9600);

BTSerial.begin(38400); // HC-05 default speed in AT command more

pinMode(2,OUTPUT);//1-+

pinMode(3,OUTPUT);//1-

pinMode(4,OUTPUT);//2+

pinMode(5,OUTPUT);//2-

pinMode(5,OUTPUT);//3+

pinMode(6,OUTPUT);//3-

pinMode(7,OUTPUT);//4+

pinMode(8,OUTPUT);//4-

}

void loop()

{

while (BTSerial.available()) {

delay(10);

char c = BTSerial.read();

readdata += c;

}

if (readdata.length() > 0) {

if (readdata == "stop")

{

digitalWrite(13, LOW);

Serial.println(readdata);

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

//delay(100);

}

else if (readdata == "forward")

{

digitalWrite (13, HIGH);

Serial.println(readdata);

digitalWrite(2,LOW);

digitalWrite(3,HIGH); //FORWARD

digitalWrite(4,LOW);

digitalWrite(5,HIGH);

digitalWrite(6,LOW);

digitalWrite(7,HIGH);

digitalWrite(8,LOW);

digitalWrite(9,HIGH);

//delay (100);

}

else if (readdata == "backward")

{

digitalWrite (13, HIGH);

Serial.println(readdata);

digitalWrite(2,HIGH);

digitalWrite(3,LOW);

digitalWrite(4,HIGH);

digitalWrite(5,LOW); //BACKWARD

digitalWrite(6,HIGH);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

//delay (100);

}

else if (readdata == "right")

{

digitalWrite (13, HIGH);

Serial.println(readdata);

digitalWrite(2,LOW);

digitalWrite(3,HIGH);

digitalWrite(4,LOW); //RIGHT

digitalWrite(5,LOW);

digitalWrite(6,LOW);

digitalWrite(7,HIGH);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

//delay (100);

}

else if (readdata == "left")

{

digitalWrite (13, HIGH);

Serial.println(readdata);

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,HIGH); //LEFT

digitalWrite(6,LOW);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,HIGH);

//delay (100);

}

readdata = "";

}

}