

KARABÜK ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

2015710213007 ASIM HABO

2014010213022 MERVE ERDEM

NAME OF THE PROJECT :

Bluetooth Controlled Car

LECTURER :

Ferhat ATASOY

Our robot is a car that has 3 Success criteria:

1. Bluetooth controlled car.

2. Red , Orange and Green Traffic light control.

3. Adjust speed of the car according to the distance with the leading car.

The components:

1. Car chassis.

2. Arduino Mega.

3. Adafruit arduino motor shield.

4. HC06 Bluetooth model.

5. Ultrasonic sensor HC-SR04.

6. Color sensor.

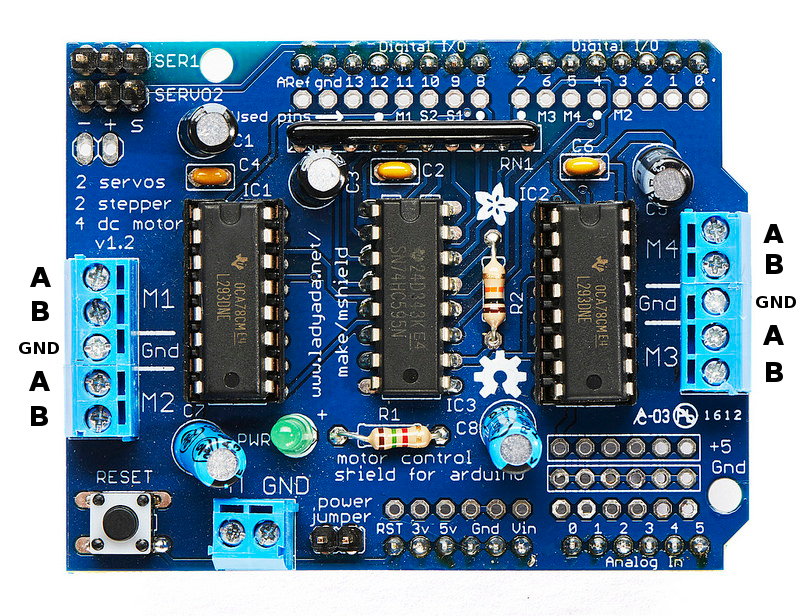
7. Batteries.

8. Power Bank.

9. Male to male and male to female Wires .

How to Attach DC motors of wheels to Adafruit motor shield?

We have 2wd car and for each wheel we have a DC motor, so in total there is 2 DC motors which will be connected to our motor shield.



Adafruit motor shield can handle 4 dc motors, and that motors can be connected to M1 , M2 , M3 and M4 with positive and negative signals ( A And B Pins ) .

We connect our 2 DC motors to M1 and M2 , 2 wires are attached to each DC motor , one is positive and the other is negative , so that 2 wires of each DC motor are connected to A and B pins . in total 4 wires are connected to A and B pins of M1 as well to A and B pins of M2 .

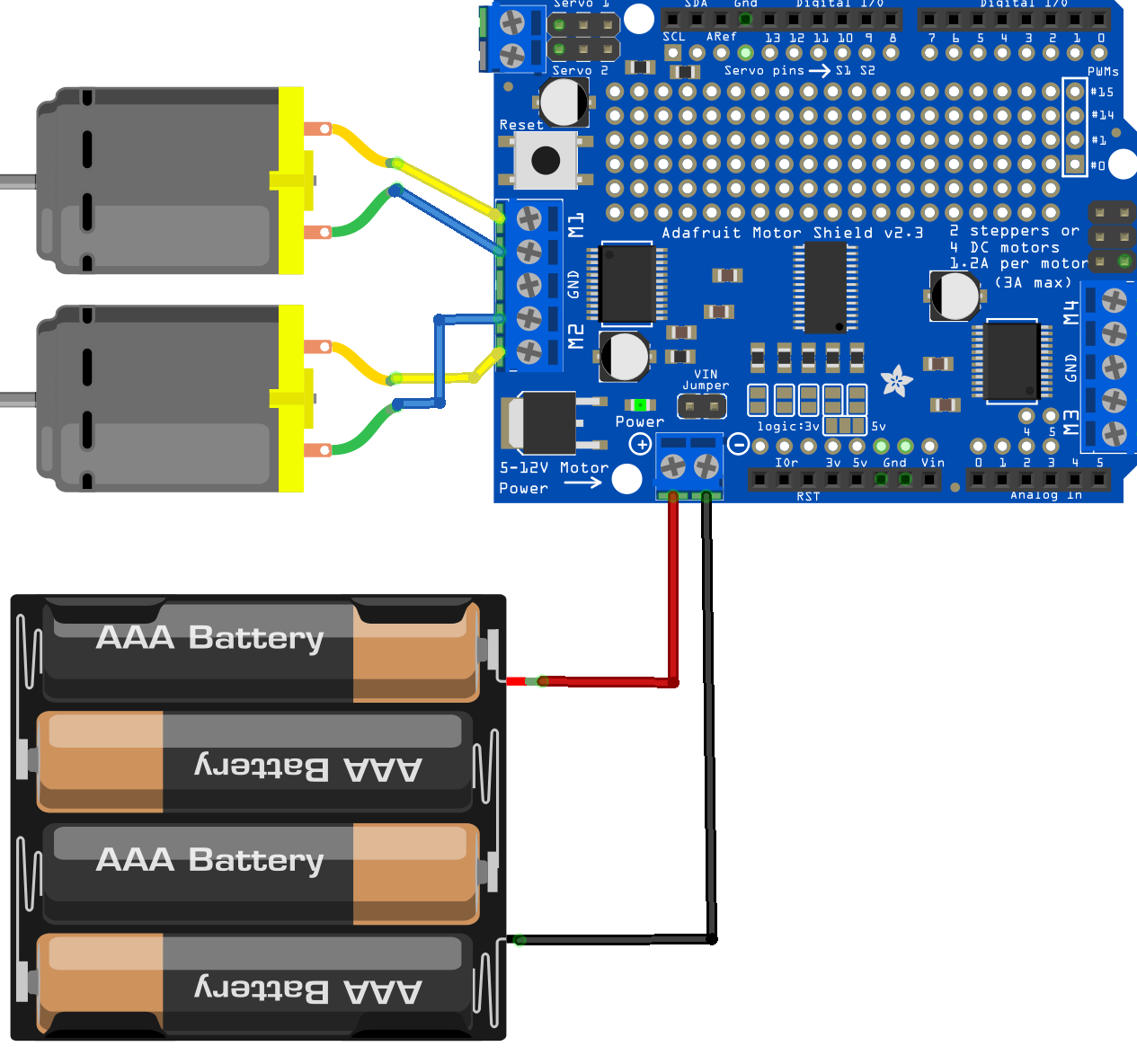
How to power Adafruit motor shield ?

We use 4 batteries to power our motor shield and as well DC motors,

Each battery is 1.5V then totally give powers by 6V , as we know Adafruit shield can only take power between 5V and 12V .

That batteries are put in battery holder and connected to Adafruit motor shield with 2 wires, one to 5V and the other is ground ( GND ) .

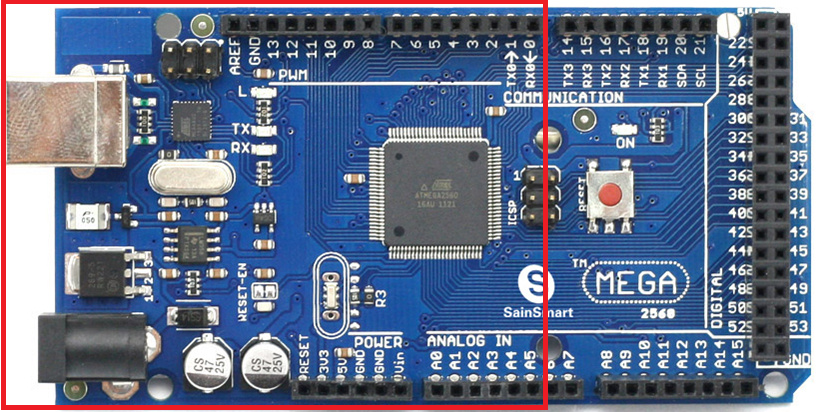
And it's the final diagram of Adafruit motor shield and the attached components:



Attaching Adafruit motor shield to Arduino board :

No need to wires for connecting our motor shield to Arduino , Adafruit can be attached directly to Arduino board .

Headers of The motor shield are attached to Arduino pins , the figure below shows which pins will hold the headers of the motor shield .



Testing Motor Shield with Arduino:

We wanted to test if the motor shield works with our DC motors , so we started with installing Adafruit motor shield library to Arduino software and then writing a code .

#include <AFMotor.h>

//motor sheild

**AF\_DCMotor** motor1(1); //Create an object motor1 and define the first motor to M1

**AF\_DCMotor** motor2(2); //Create an object motor2 and define the second motor to M2

void setup() {

motor1.**setSpeed**(255); // set the speed of the first motor to 250 (Max is 255 and Min is 150)

motor2.**setSpeed**(255); //set the speed of second motor to 250 (Max is 255 and Min is 150 )

}

void loop() {

//run command is to power the DC motors from Adafruit motor shield

motor1.**run**(FORWARD); // let the first motor run forward

motor2.**run**(FORWARD); // let the second motor run forward

delay(1000); // after running forward wait for 1s

motor1.**run**(BACKWARD); // let the first motor run backward

motor2.**run**(BACKWARD); // let the second motor run backward

delay(1000); // wait again for 1s

motor1.**run**(RELEASE); // let the first motor stop

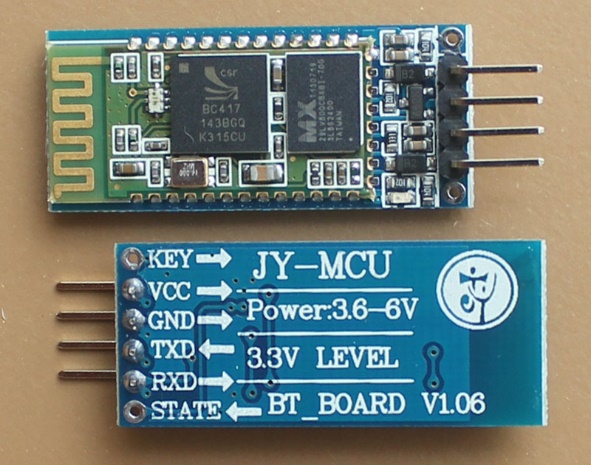
motor2.**run**(RELEASE); // let the second motor stop

delay(1000);

}

We wrote this code and uploaded to Arduino and the code runs as long as uploading is completed , so DC motors started to move forward once then wait for 1s after that to move backward then wait again for 1 sec and then stop completely before the loop runs again and again .

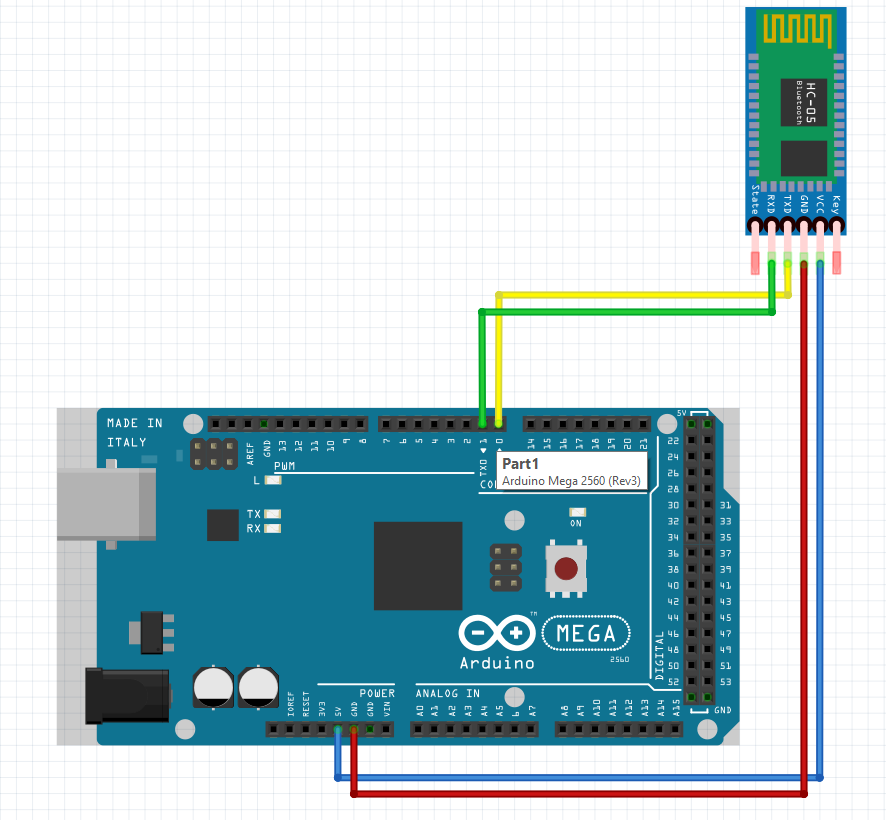
How does Bluetooth work?



Bluetooth module HC – 06 has 4 pins which are VCC, GND, TXD and RXD. We connect them with male to female wires to Arduino mega.

VCC means the power of Bluetooth which must be between 3.6V and 6V so we connect VCC pin to 5V from Arduino mega, In addition GND pin will be connected to GND of Arduino mega , TX(1) and RX(0) communicates on digital pins with the computer or other devices.

We connect RXD of Bluetooth to TX of Arduino, and TXD of Bluetooth to RX of Arduino so like this the pins are all connected to Arduino board.



As we see , Bluetooth wires will be connected to some common pins with the motor shield which will place above the Arduino mega , so firstly we connect the wires to that pins of Arduino then attach the motor shield above that wires .

Testing Bluetooth model:

We wanted to test the Bluetooth model and control the car with a smart mobile, after assembling wheels and attaching them to the car chassis, as well placing our Arduino on the car chassis, it’s the time to test whether it works with Bluetooth.

We wrote a Bluetooth code and uploaded to Arduino :

#include <AFMotor.h>

**AF\_DCMotor** motor1(1); //Create an object motor1 and define the first motor to M1.

**AF\_DCMotor** motor2(2); //Create an object motor2 and define the second motor to M2.

char bt='S'; // assign a char variable to S serial which comes from the application in the smart mobile.

void setup()

{

Serial.**begin**(9600); // starting the serial

motor1.**setSpeed**(250); // set the speed of the first motor to 250

motor2.**setSpeed**(250); //set the speed of second motor to 250

Stop(); // call Stop function }

void loop() {

bt=Serial.read(); // read the serial that coming from the application and assign it to bt variable.

if(bt=='F') // if it reads F from the application , F means we press forward button in the application.

{

forward(); // call forward function

}

if(bt=='B') // if it reads B from the application , B means we press backward button in the application

{

backward(); // call backward function

}

if(bt=='L') // if it reads L from the application , L means we press Left button in the application

{

left(); // call left function

}

if(bt=='R') // if it reads R from the application , R means we press right button in the application

{

right(); // call right function

}

if(bt=='S') // if it reads S from the application , S means we press stop button in the application

{

Stop(); // call stop function

}

} // end of loop

//Functions

void forward() // in forward function , the car moves forward

{

motor1.**run**(FORWARD);

motor2.**run**(FORWARD);

}

void backward() // in backward function , the car moves backward

{

motor1.**run**(BACKWARD);

motor2.**run**(BACKWARD);

}

void left() // in left function , the car moves to left

{

motor1.**run**(FORWARD);

motor2.**run**(BACKWARD); // when the first motor run forward and the second motor run backward , this creates a movement to left

}

void right() // in right function , the car moves to right

{

motor1.**run**(BACKWARD);

motor2.**run**(FORWARD); // when the first motor run backward and the second motor run forward , this creates a movement to right

}

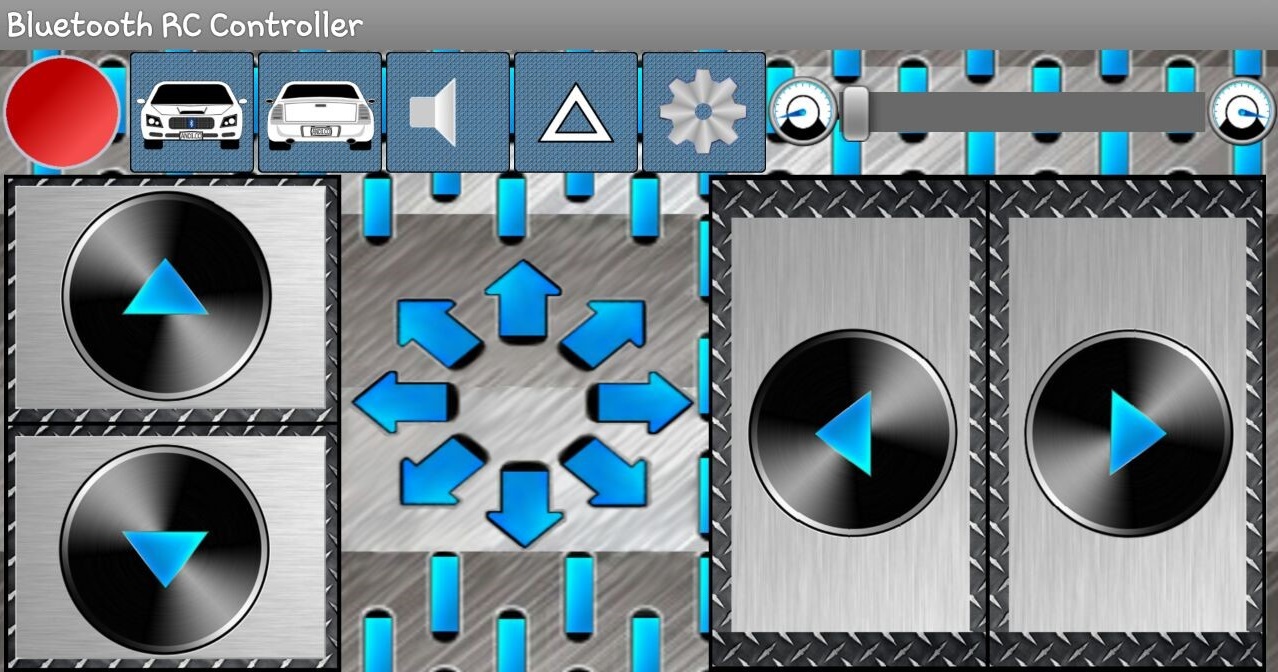
void Stop() //in stop function , the car stops

{

motor1.**run**(RELEASE);

motor2.**run**(RELEASE);}

After uploading this code to Arduino, we make a connection between our smart mobile and Bluetooth model then run the application, the application name is Bluetooth RC Controller:



We connected our Arduino to a power bank that gives continuous power to Arduino via USB input , then we started to test it and all worked well .

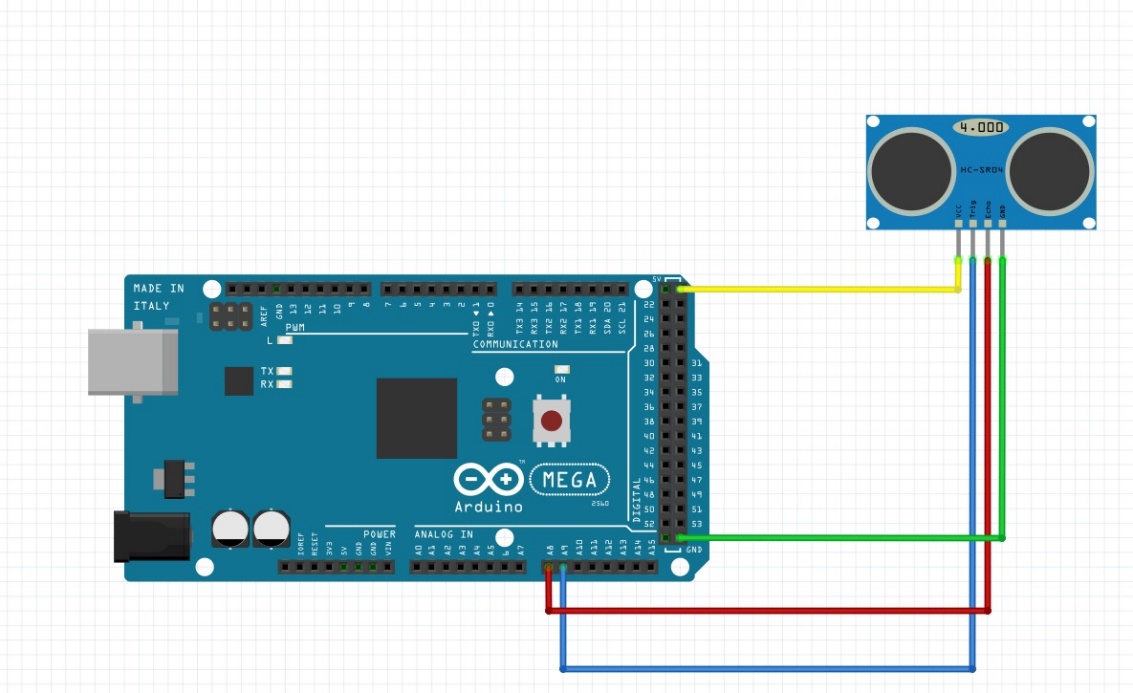
Connecting Distance Sensor to Arduino:

Distance Sensor HC-SR04 has 4 pins which are VCC , GND , Trig and Echo .

It has two Transducer , one Transducer sends an ultrasonic signal and this signal hits a block and comes back to the other Transducer , with applying this operation it determines the distance with the duration of reaching a block that makes the signal go back to the other Transducer .

The Transducer which sends the signal is called by Transmitter (T) and the Transducer which receives that signal is called by Receiver (R) .

Vcc is connected to 5V of Arduino , Gnd is connected to Gnd of Arduino , Trig pin is connected to an analog pin of Arduino , that trig pin prompts Transmitter to send the signals and the last pin is Echo is also connected to an analog pin of Arduino , that Echo receives the waves that coming back after hitting a block .



Testing Distance Sensor with Arduino and Bluetooth :

#include <AFMotor.h>

//motor sheild

AF\_DCMotor motor1(1);

AF\_DCMotor motor2(2);

const int trigPin = A9; // Trig pin to an analog pin of Arduino

const int echoPin = A8; //Echo Pin to an analog pin of Arduino

long duration; // a variable to store the duration according to the time

int distance; // a variable to store the distance between the sensor and a block

//Bluetooth

char bt='S';

void setup() {

Serial.begin(9600);

Stop();

//Distance setup

pinMode(trigPin,OUTPUT); // Sender

pinMode(echoPin,INPUT); // Receiver

}

void loop() {

bt=Serial.read();

if(bt=='F')

{

forward();

}

if(bt=='B')

{

backward();

}

if(bt=='L')

{

left();

}

if(bt=='R')

{

right();

}

if(bt=='S')

{

Stop();

}

}

//Functions

//Bluetooth

void forward()

{

digitalWrite(trigPin, LOW); // put Trigpin off for a while

digitalWrite(trigPin, HIGH); // put Trig pin on to motivate the transmitter to send signals

digitalWrite(trigPin, LOW); // put it off again to stop sending signals

duration = pulseIn(echoPin, HIGH); // put echo pin on to receive the signals that comes back after hitting a block , then store the duration of the signal's trip until a block

distance= duration\*0.034/2; //distance = time \* speed , here we the speed is constant which it's sound speed , we divide the answer by 2 , because we need only one round trip

if(distance<=10) // if the determined distance is less than 10

{

Stop(); // call stop function to not make the car move

}

else if(distance>=11 && distance<=20) // if the distance between 11 and 20

{

motor1.setSpeed(100); // reduce the speed of DC motors and set it as 100

motor2.setSpeed(100);

motor1.run(FORWARD); // then let it move forward

motor2.run(FORWARD);

}

else if(distance>=21 && distance<=25) // if the distance between 21 and 25

{

motor1.setSpeed(150); // reduce the speed of DC motors and set it as 150

motor2.setSpeed(150);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

else // if there is no block in front of the sensor

{

motor1.setSpeed(250); // set the speed as max and let the car move

motor2.setSpeed(250);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

}

void backward()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(BACKWARD);

motor2.run(BACKWARD);

}

void left()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(FORWARD);

motor2.run(BACKWARD);

}

void right()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(BACKWARD);

motor2.run(FORWARD);}

void Stop()

{

motor1.run(RELEASE);

motor2.run(RELEASE);}

Connecting Color Sensor To Arduino :

Color sensor has 4 led that reads RGB colors , on other hand it gives data for RGB ,that data is numbers for every color of RGB , so we take that data and compare it with the other data that the car will read it while moving , and with comparing the data of RGB we can detect the colors .

It has 8 pins , there is VCC and GND pins that is connected respectively to 5V and GND of Arduino , the other pins are connected to digital pins of Arduino Mega .

Our color sensor light has an 8 x 8 array of photodiodes , The photodiodes have three different color filters. Sixteen of them have red filters, another 16 have green filters, another 16 have blue filters and the other 16 photodiodes are clear with no filters.

Each 16 photodiodes are connected in parallel, so using the two control pins S2 and S3 we can select which of them will be read , and The sensor has two more control pins, S0 and S1 which are used for scaling the output frequency. The frequency can be scaled to three different preset values of 100 %, 20 % or 2%. This frequency-scaling function allows the output of the sensor to be optimized for various frequency counters or microcontrollers.

How to read RGB colors :

// the used PINs

const int s0=22;

const int s1=23;

const int s2=24;

const int s3=25;

const int sensorData=26;

// Global variable to store data

int data=0;

void setup()

{

// PINs configuration

pinMode(s0,OUTPUT);

pinMode(s1,OUTPUT);

pinMode(s2,OUTPUT);

pinMode(s3,OUTPUT);

pinMode(sensorData,INPUT);

// Serial setup

Serial.begin(9600);

// Frequency scale

digitalWrite(s0,HIGH);

digitalWrite(s1,HIGH); // S0 = High and S1 = High means 100% of Frequency counters .

}

void loop()

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* getting the RED COLOR data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Set S2 and S3 to LOW in order to sense the RED color

digitalWrite(s2,LOW);

digitalWrite(s3,LOW);

// store data in the Global varibale

data=pulseIn(sensorData,LOW);

// Display data through the Serial monitor

Serial.print("Red data= ");

Serial.print(data);

Serial.print(" ");

delay(20);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* getting the GREEN COLOR data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Set S2 and S3 to HIGH in order to sense the Green color

digitalWrite(s2,HIGH);

digitalWrite(s3,HIGH);

data=pulseIn(sensorData,LOW);

Serial.print("Green data= ");

Serial.print(data);

Serial.print(" ");

delay(20);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* getting the BLUE COLOR data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Set S2 to low and S3 to HIGH in order to sense the Blue color

digitalWrite(s2,LOW);

digitalWrite(s3,HIGH);

data=pulseIn(sensorData,LOW);

Serial.print("Blue data= ");

Serial.print(data);

Serial.println(" ");

delay(20);

delay(2000);

}

With this code we read RGB colors by color sensor , After reading RGB colors of the objects we want to read , we can compare RGB colors of that objects while the car is moving .

The final code :

Now all our components are ready and connected to Arduino .

The code:

#include <AFMotor.h>

//motor sheild

AF\_DCMotor motor1(1);

AF\_DCMotor motor2(2);

const int trigPin = A9;

const int echoPin = A8;

long duration;

int distance;

//Bluetooth

char bt='S';

//Color sensor

const int s0=22;

const int s1=23;

const int s2=24;

const int s3=25;

const int sensorData=26;

int dataR=0;

int dataG=0;

int dataB=0;

void setup() {

Serial.begin(9600);

Stop();

//Distance setup

pinMode(trigPin,OUTPUT); // Sender

pinMode(echoPin,INPUT); // Receiver

//Color setup

pinMode(s0,OUTPUT);

pinMode(s1,OUTPUT);

pinMode(s2,OUTPUT);

pinMode(s3,OUTPUT);

pinMode(sensorData,INPUT);

digitalWrite(s0,HIGH);

digitalWrite(s1,HIGH);

}

void loop() {

bt=Serial.read();

//Detect Color

digitalWrite(s2,LOW);

digitalWrite(s3,LOW);

dataR=pulseIn(sensorData,LOW);

digitalWrite(s2,HIGH);

digitalWrite(s3,HIGH);

dataG=pulseIn(sensorData,LOW);

digitalWrite(s2,LOW);

digitalWrite(s3,HIGH);

dataB=pulseIn(sensorData,LOW);

if(isRed() || isYellow() || isGreen()){ // if the color sensor detect one of this Color

if(isRed()){ // if the detected color and data is belonged to Red

Stop(); // stop the DC motors

}

else if (isYellow()){ // if the detected color and data is belonged to Yellow

motor1.setSpeed(100); // set the speed at lower speed

motor2.setSpeed(100);

motor1.run(FORWARD); // run the car forward

motor2.run(FORWARD);

delay(1000);

motor1.run(BACKWARD); // run the car backward ,

motor2.run(BACKWARD);

Stop(); // then stop , it means show that the car is ready to move before the green color .

}

else if(isGreen()){

forward();

}

}

if(bt=='F')

{

forward();

}

if(bt=='B')

{

backward();

}

if(bt=='L')

{

left();

}

if(bt=='R')

{

right();

}

if(bt=='S')

{

Stop();

}

}

//Functions

//Bluetooth

void forward()

{

digitalWrite(trigPin, LOW);

digitalWrite(trigPin, HIGH);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance= duration\*0.034/2;

if(distance<=10)

{

Stop();

}

else if(distance>=11 && distance<=20)

{

motor1.setSpeed(100);

motor2.setSpeed(100);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

else if(distance>=21 && distance<=25)

{

motor1.setSpeed(150);

motor2.setSpeed(150);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

else

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(FORWARD);

motor2.run(FORWARD);

}

}

void backward()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(BACKWARD);

motor2.run(BACKWARD);

}

void left()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(FORWARD);

motor2.run(BACKWARD);

}

void right()

{

motor1.setSpeed(250);

motor2.setSpeed(250);

motor1.run(BACKWARD);

motor2.run(FORWARD);

}

void Stop()

{

motor1.run(RELEASE);

motor2.run(RELEASE);

}

//Detect Red color

bool isRed()

{

if((dataR<38 && dataR>25) && (dataG<68 && dataG>50) && (dataB<55 && dataB>38))

{

return true;

}

else

{

return false;

}

}

//Detect Green color

bool isGreen()

{

if((dataR<50 && dataR>36) && (dataG<53 && dataG>35) && (dataB<52 && dataB>40))

{

return true;

}

else

{

return false;

}

}

bool isYellow()

{

if((dataR<31 && dataR>19) && (dataG<48 && dataG>28) && (dataB<48 && dataB>35))

{

return true;

}

else

{

return false;

}

}

The Final Diagram :

