Department of Computer Science and Engineering

Project: **Cybot Warrior Cyber Security Robot Car**

Course Title: Peripheral & Interfacing Lab

Course Code: CSE- 316

Submission Date: 19-06-2019

Submitted To:

Abdullah Al Omar

B.Sc. in CSE,

University of Asia pacific.

Submitted By:

Name : Md Mehedi Hasan	Name: Md All Mamun	Name: Arafat Bin Kashem
ID: 16201073	ID: 16201071	ID: 16201066
Sec: A2	Sec: A2	Sec: A2

Contents

1.INTRODUCTION:

Cybot Warrior Cyber Security Robot is a Car model robot which has lots of cyber security features that can be used for ethical hacking and penetration testing purpose and also it is an automatic Robot car.

2.FEATURES:

- Automatic Obstacle Avoidance Robot Car with Fastest Speed and High Accuracy
- Automatic and Manual Controlled Robot Accident Avoidance Car with Fastest Speed and High Accuracy
- Bluetooth Controlled by Android Smart Phone
- Wi-Fi Controlled by Wireless Keyboard and Touch Pad
- Wi-Fi Jamming
- Wi-Fi Routing
- Display Support
- Kali Linux Hacking Operating System with Single Board Computing

3.Device, Equipment, Software and Sensors:

- Arduino UNO
- Arduino Connecting Cable
- Arduino IDE
- L239D Motor Controller
- 4 Yellow DC Motor
- 4 Yellow Wheel
- 1 Tower Pro Servo Motor
- Ultrasonic SR-04 Sensor
- Bluetooth Module HC-05
- Switch
- Home Made Card Board Chassis
- Jumper Wire or Connecting Wire Set
- 4 Rechargeable LIPO Battery 3.7 Volt
- Battery Charger
- Battery Holder
- 5 Volt Power Bank
- Android Smart Phone
- Raspberry pi 3 b+
- 3.5-inch touch display wave share
- Mini Wireless Keyboard with Touch Pad

FULL INFORMATION:

Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 Microprocessor. It has 14 digital input/output pins (of which 6 can be used as PWM outputs),6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller. simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip.

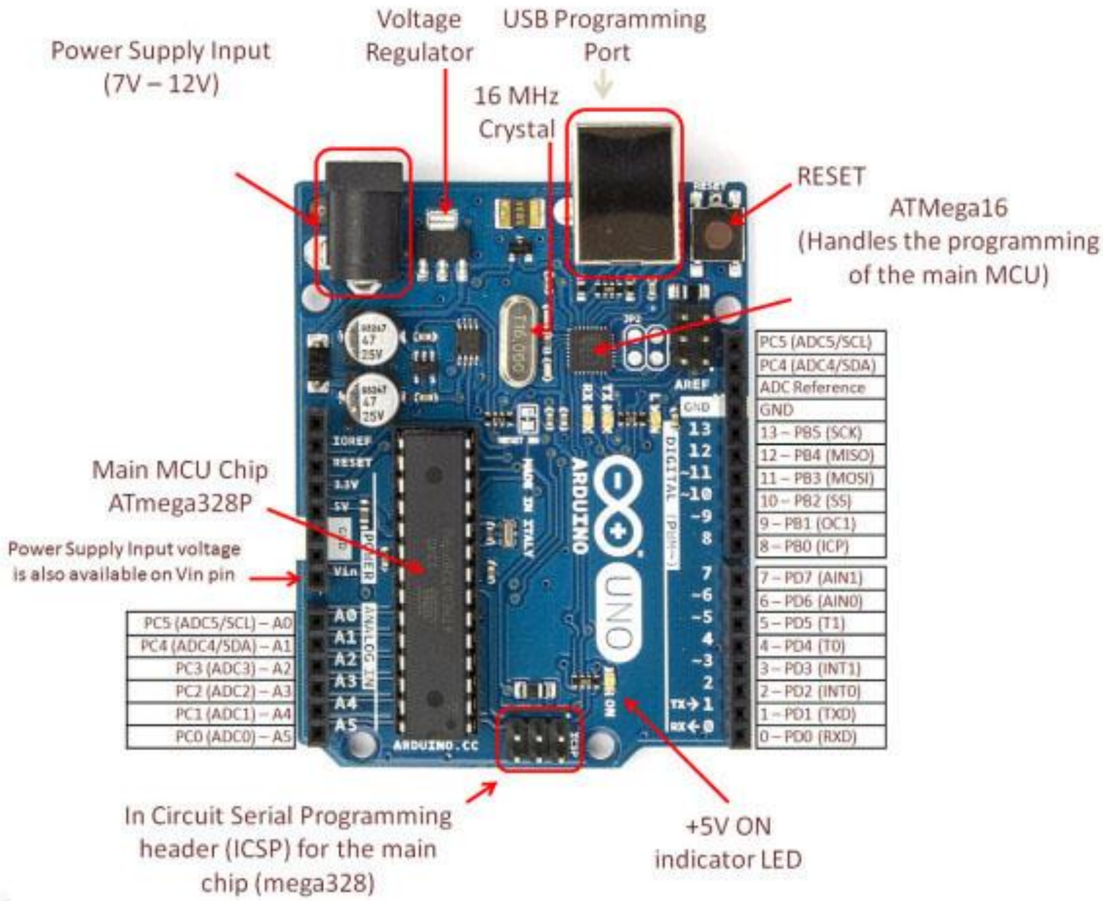


Figure 1: Arduino UNO with Pin Configuration and information

Arduino Uno Connecting Cable:

This is a standard USB 2.0 cable. This is the most common A to B Male/Male type peripheral cable, the kind that's usually used for printers. Compatible with most SFE designed USB boards as well as USB Arduino boards like the Uno. Using this connecting cable can be upload Arduino Uno board also can be use Supply power (5V)

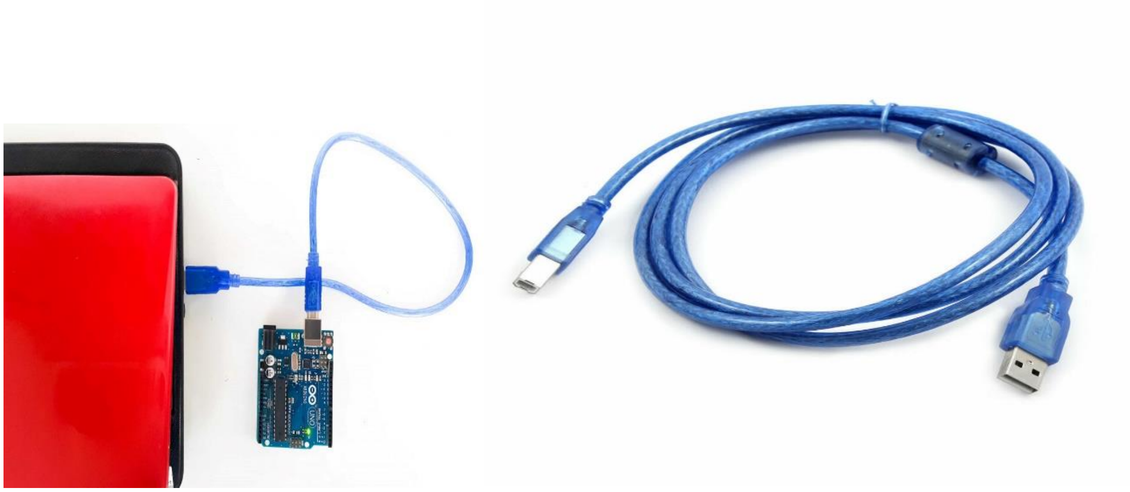


Figure 2: Using Arduino UNO Cable Connect Laptop with Arduino

Arduino IDE:

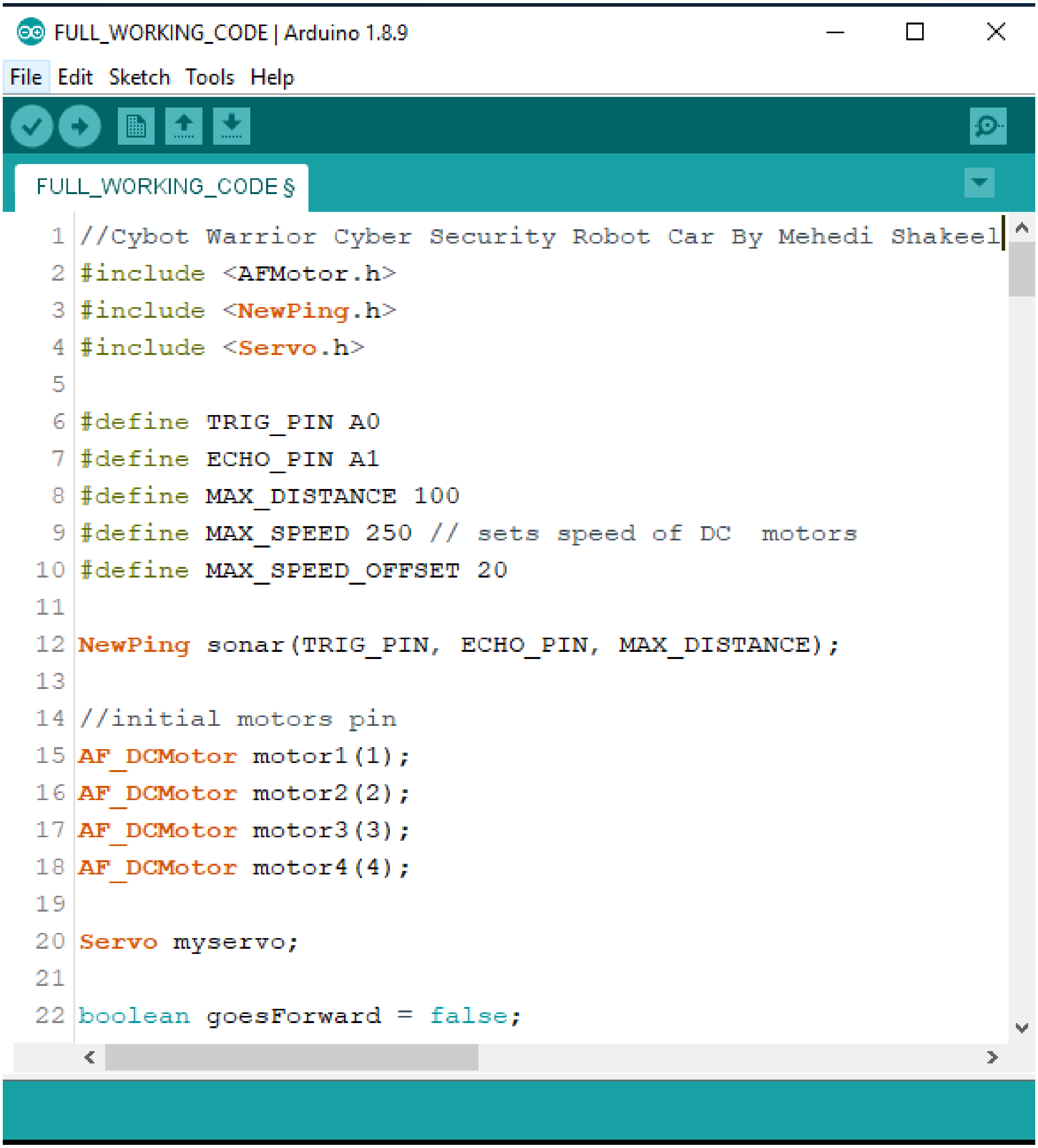
Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

Arduino IDE Windows Download : https://www.arduino.cc/download_handler.php?f=/arduino-1.8.9-windows.exe

Arduino IDE Linux Download : https://www.arduino.cc/download_handler.php?f=/arduino-1.8.9-linux64.tar.xz

Arduino IDE MAC OS X Download : https://www.arduino.cc/download_handler.php?f=/arduino-1.8.9-linux64.tar.xz

Arduino IDE Example:


FULL_WORKING_CODE | Arduino 1.8.9
File Edit Sketch Tools Help
FULL_WORKING_CODE §

```
//Cybot Warrior Cyber Security Robot Car By Mehedi Shakeel
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>

#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 100
#define MAX_SPEED 250 // sets speed of DC  motors
#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

//initial motors pin
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

Servo myservo;

boolean goesForward = false;
```

Figure 3: Arduino Ide Example

L239D Motor Controller:

L293D is a dual H-bridge 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.

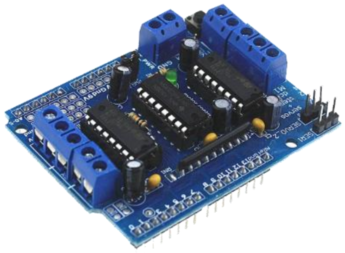


Figure 4: L293D Motor Driver

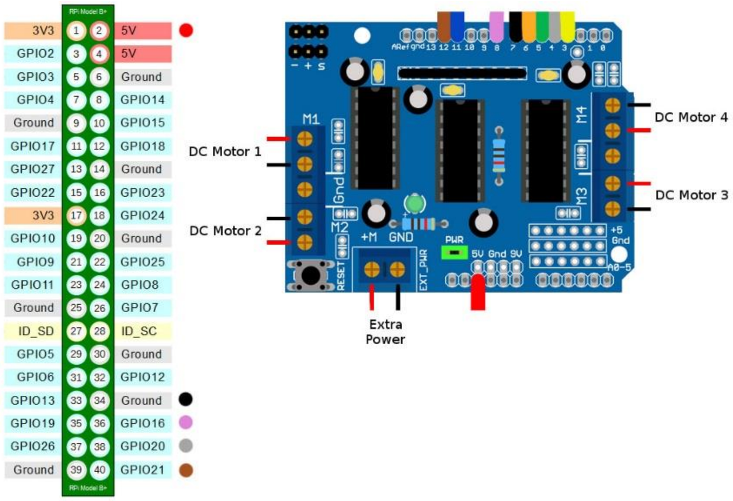


Figure 5: L293D Motor Driver Pin Configuration

Yellow DC Motor & Wheel:

DC gear motor and wheel set for making robots! These motors are light weight, high torque and low RPM. They can climb hills and have excellent traction, plus you can mount the wheel on either side of the motor with its double-sided output shaft. The 6V, 83rpm Micro Yellow DC Motor with Back Shaft is ideal for DIY enthusiasts. This motor is inexpensive, small, easy to install, and ideally suited for use in a mobile robot car. We recommend this L2398 Motor shield to driver this motor. There is a wheel set in here.

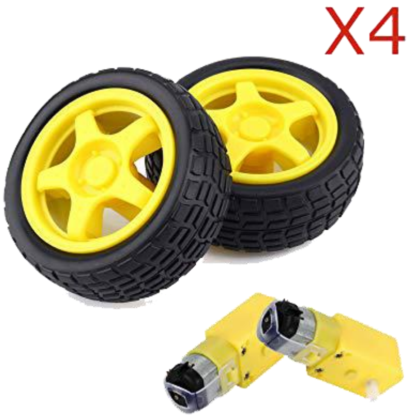


Figure 6: Yellow DC Motor and Wheel

Specifications:

- Motor Voltage: 3 – 12 V
- Motor Current: 70 mA (typical) – 250 mA (max)
- Speed: up to 170 RPM
- Torque: up to 0.8 Kg
- Gear Ration: 1:48
- Wheel Diameter: 66 mm
- Wheel Width: 27 mm
- No-load speed(3V): 50RPM
- No-load speed(5V): 83RPM
- Torque (3V): 0.8Kgcm
- Torque (5V): 1.0Kgcm
- Size: 69mm * 37mm * 23mm

- Weight:30g

Tower Pro Servo Motor:

Servo Motor is tiny and lightweight with high output power motor. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

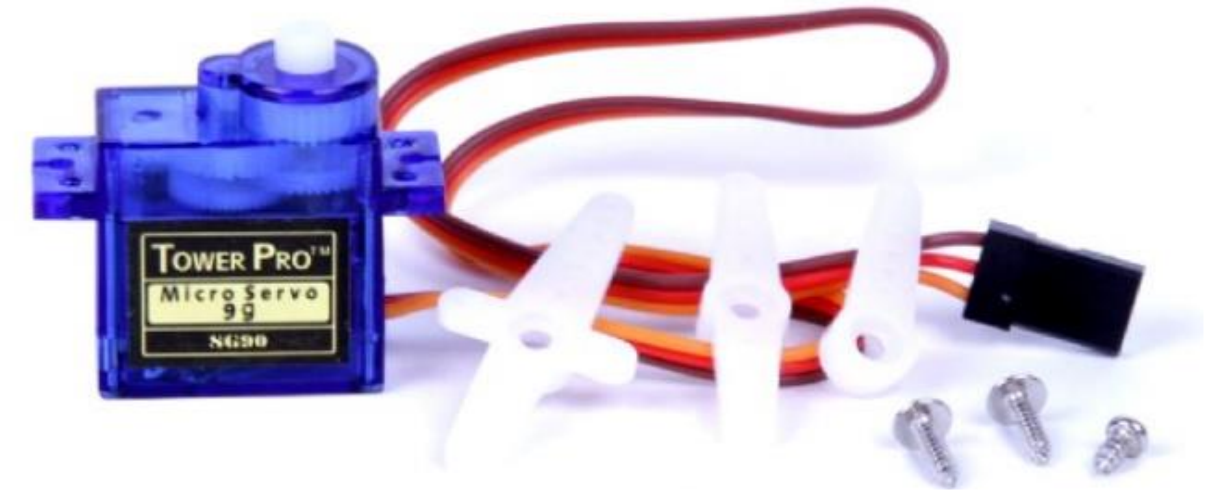


Figure 7: Tower Pro Servo Motor

Tower Pro Servo Connection Diagram:

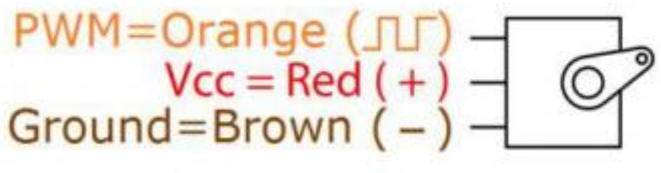


Figure 8: Servo Motor Connection

- PWM connect with any PWM GPIO Pin
- VCC Connect With 5-volt GPIO Pin
- Ground/GND Connect with any GPIO GND Pin

Ultrasonic HC-SR04:

HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

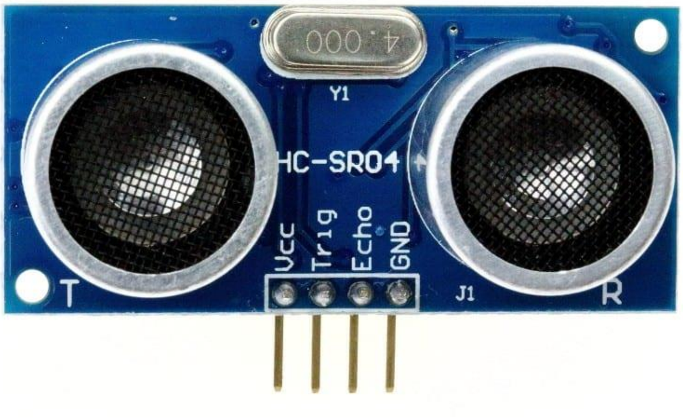


Figure 9: Ultrasonic HC-SR04 Sensor

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

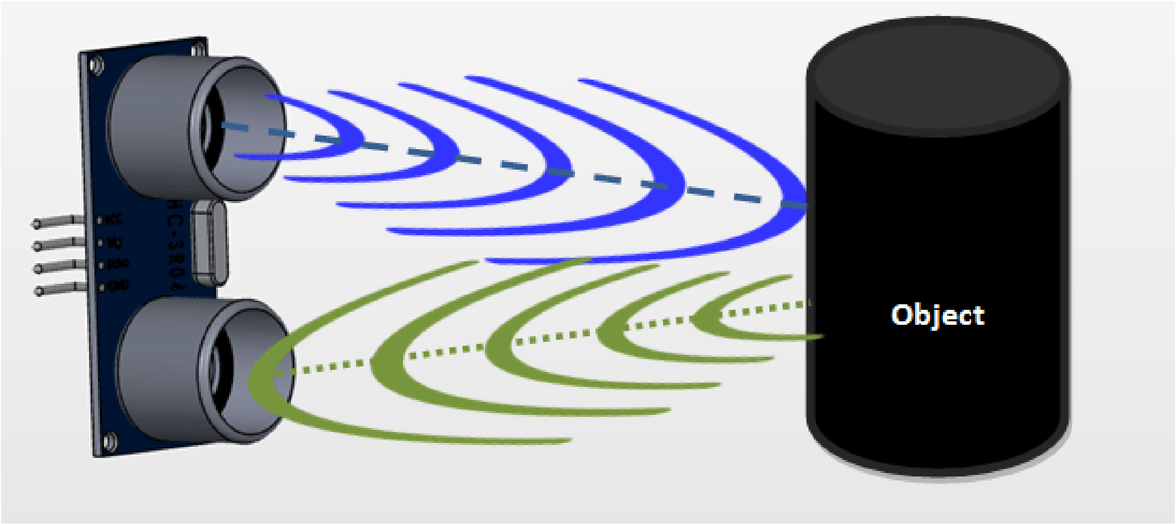


Figure 10: Ultrasonic Sensor Working Principle

Bluetooth Module HC-05:

The HC-05 Bluetooth Module is a very cool module which can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rates hence it is easy to interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. So, if you looking for a Wireless module that could transfer data from your computer or mobile phone to microcontroller or vice versa then this module might be the right choice for you. However, do not expect this module to transfer multimedia like photos or songs; you might have to look into the CSR8645 module for that.

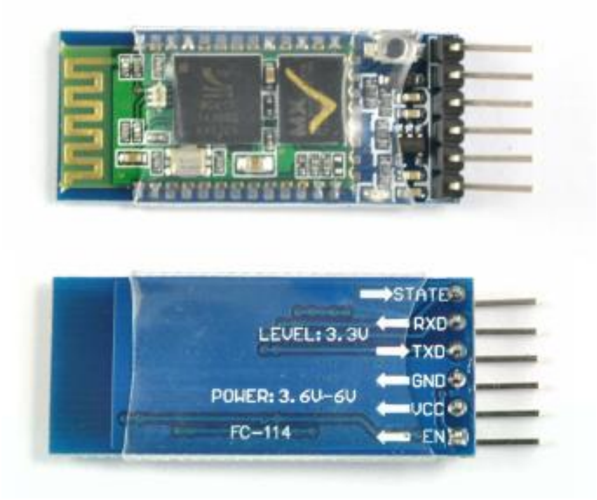


Figure 11: HC-05 Bluetooth Module

Pin Diagram of HC-05 Bluetooth Module:

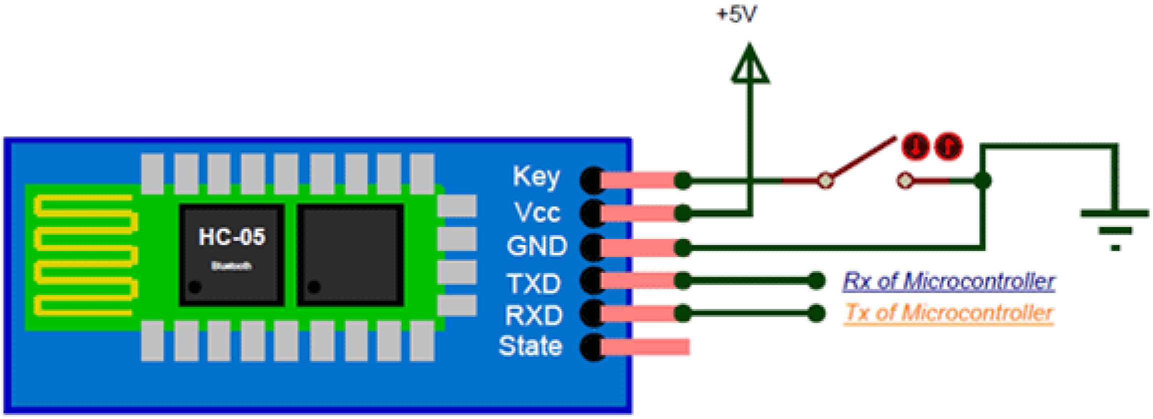


Figure 11: HC-05 Bluetooth Module Pin Diagram

Switch:

Its use for controlling power ON/OFF supply from battery or power source to Arduino boards

Figure 12: Switch

Home Made Car Chassis:

Using Card board we can make a base of our robot car where all the equipment and sensors will be attached. Also, we can buy readymade chassis from shop

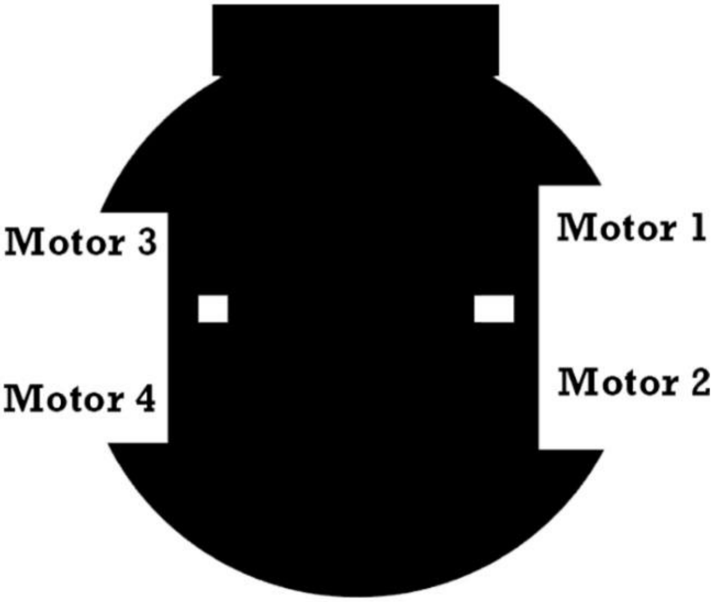


Figure 13: Home Made Robot Car Chassis

Jumper Wire or Connecting Wire Set:

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

It's uses to build the circuit or connection between Arduino boards and other sensors to communicate with each hardware equipment

Figure 13: Jumper wire

Rechargeable LIPO Battery 3.7 Volt or Power Bank:

Rechargeable LIPO battery is the best way to give power supply to any kinds of electronics hardware Arduino projects or robotics projects because this is rechargeable so that it can re-use.

Figure 14: Rechargeable LIPO Battery

Battery Charger:

For Recharging this lipo battery, we need a special battery charger. During the project making it can be discharge so we need this special battery charger below

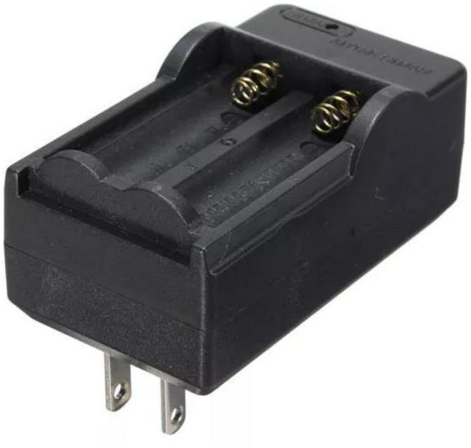


Figure 15: Lipo Battery Charger

Battery Holder:

To attached the battery, we need a battery holder so that we can easily add those battery to our robot car chassis.

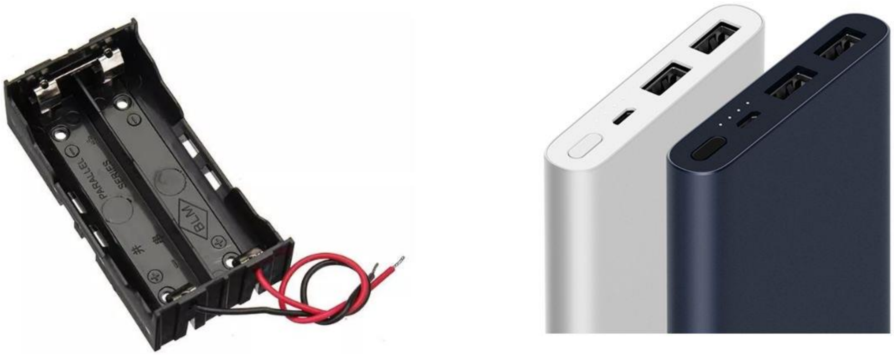


Figure 16: Lipo Battery Holder and Power bank

Also, battery holder can protect the battery from outside damages like heat. Also, we can use power bank as a power supply but I prefer to use separate power source for different parts of your project. It will help us to give recommended power supply to our projects and make it faster to work.

Android Smart Phone:

There are lots of android apps are available on play store to control Arduino based robotics project using Bluetooth or Wi-Fi module to get connected with your robotics projects we need to choose app according to our project here is the app I use to control my robot below

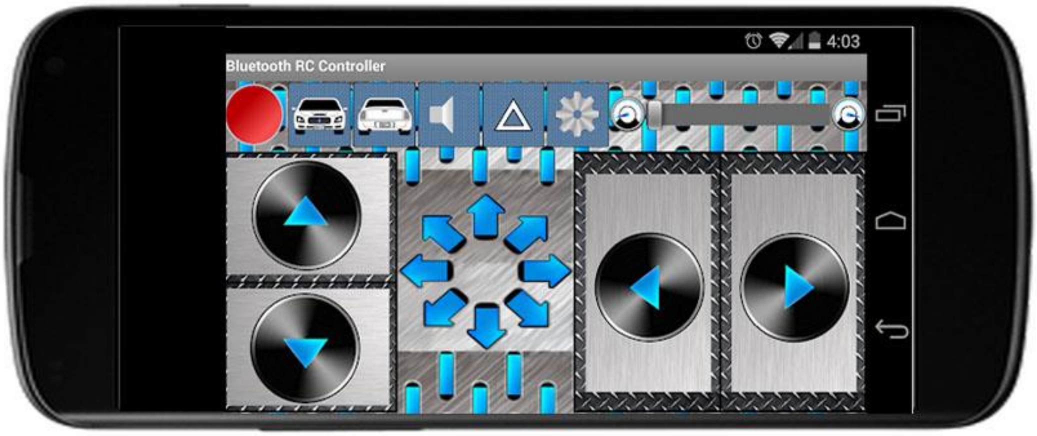


Figure 17: Robot Controller using Android App

Mobile App Link : https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller

Raspberry pi 3 b+:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. We use raspberry pi 3 b+ to run Kali Linux Hacking Operating system on it to add almost all cyber security features.

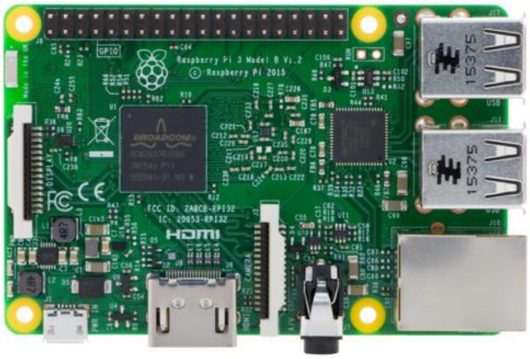


Figure 18: Raspberry Pi 3B+

Kali Linux OS Img for Raspberry Pi Download Link : https://images.offensive-security.com/arm-images/kali-linux-2019.2-rpi3-nexmon-64.img.xz

3.5-inch Touch Display:

We use 3.5-inch tft touch screen display to raspberry pi as a monitor or screen of our kali Linux operating system. So that we can see what is happening in our kali Linux attacks and the outputs. We can use it as a WIFI jammer, WIFI router and different types of cyber security attacks.

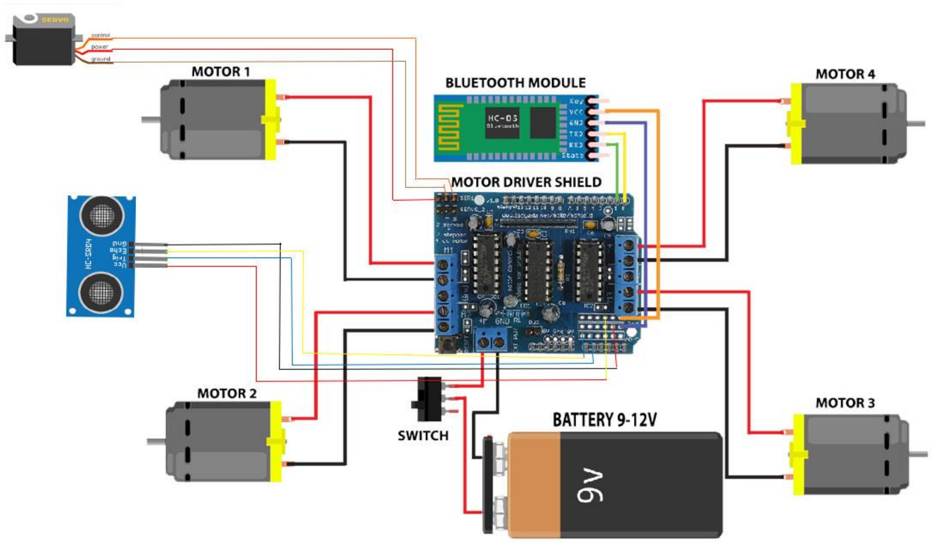
Figure 19: Kali Linux OS with Touch Screen

Mini Wireless Keyboard with Touch Pad:

This mini wireless keyboard with touch pad helps us to control and use our kali Linux operating system as like a single board computer.

Figure 20: Wireless Keyboard with Touchpad

Circuit Diagrams to Build the Robot Car Body with All the Equipment’s:



Circuit Diagram For Cybot Warrior Robot Car

5. Steps of Making the robot

- ✓ Attach the L239d Motor controller using the gpio pins of Arduino board it fit well on Arduino board we do not need any kind of connecting wire or soldering,
- ✓ Connect 4x DC Motor two the motor controller M1, M2, M3 & M4
- ✓ Connect the lipo battery to the motor controller power in ports with a switch
- ✓ Connect the HC-05 Bluetooth Module two the Arduino gpio pin TX & RX using soldering.
- ✓ Two give power to the Bluetooth module we have to use +5 volt from motor driver GPIO pins and GND Pins
- ✓ Then Connect The servomotor with the motor driver shield servo connector section
- ✓ After that attached the ultrasonic sensor with the servo robot using ultrasonic sensor holder with glue.
- ✓ Now attached all the equipment on your home-made chassis according to your design.
- ✓ Then install kali Linux arm images on a SD card and insert it to raspberry pi and attached it to the robot chassis and using the power bank give power to the raspberry pi.
- ✓ Then connect the wireless receiver to the usb port of raspberry pi and power on the mini wireless keyboard with touchpad using the wireless connection.
- ✓ Full Video Instruction on YouTube : https://www.youtube.com/watch?v=DINk1eWiKu8&list=PLoAx5AQlvczW3r1IadJF2QM48WNyI54Rw
- ✓ Full Project Github Repositories : https://github.com/mehedishakeel/CyBot-Warrior---DIY-Robot-Cyber-Security-Warrior-By-Mehedi-Shakeel---SSTECTUTORIALS

6.Code for CYbot Warrior Robot :

```
//Cybot Warrior Robot Car Arduino Code By Mehedi Shakeel
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>
#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 100
#define MAX_SPEED 250
#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

//initial motors pin
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

Servo myservo;

boolean goesForward = false;
int distance = 100;
int speedSet = 255;

char command;

void setup()
{
  Serial.begin(9600);
  myservo.attach(10);
  myservo.write(115);
  delay(200);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);

}

void loop() {

  int a = Serial.available();


  if ( a > 0) {
    command = Serial.read();
    Serial.println(command);
```

```
Stop();

switch (command) {
  case 'F':
    forward();
    break;
  case 'B':
    back();
    break;
  case 'L':
    left();
    break;
  case 'R':
    right();
    break;
  case 'S':
    moveStop();
    break;
  case 'D':

    a = 0;
    int distanceR = 0;
    int distanceL =  0;
    delay(40);
    distance = readPing();

    if (distance)
    {
      moveStop();
      delay(100);
      moveBackward();
      delay(300);
      moveStop();
      delay(200);
      distanceR = lookRight();
      delay(200);
      distanceL = lookLeft();
      delay(200);

      if (distanceR >= distanceL)
      {
        turnRight();
        moveStop();
      } else
      {
        turnLeft();
        moveStop();
      }
      moveForward();
    }
    else
    {
      moveForward();
    }
```

```
        distance = readPing();
        break;
    }
  }

  else {

    int distanceR = 0;
    int distanceL =  0;
    delay(40);

    if (distance <= 15)
    {
      moveStop();
      delay(100);
      moveBackward();
      delay(300);
      moveStop();
      delay(200);
      distanceR = lookRight();
      delay(200);
      distanceL = lookLeft();
      delay(200);

      if (distanceR >= distanceL)
      {
        turnRight();
        moveStop();
      } else
      {
        turnLeft();
        moveStop();
      }
    } else
    {
      moveForward();
    }
    distance = readPing();



  }
  a = 0;
}
int lookRight()
{
  myservo.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(115);
  return distance;
}
```

```
int lookLeft()
{
  myservo.write(170);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(115);
  return distance;
  delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if (cm == 0)
  {
    cm = 250;
  }
  return cm;
}

void moveStop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}

void moveForward() {

  if (!goesForward)
  {
    goesForward = true;
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2) // slowly bring
the speed up to avoid loading down the batteries too quickly
    {
      motor1.setSpeed(speedSet);
      motor2.setSpeed(speedSet);
      motor3.setSpeed(speedSet);
      motor4.setSpeed(speedSet);
      delay(5);
    }
  }
}

void moveBackward() {
  goesForward = false;
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2)
  {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);
    motor3.setSpeed(speedSet);
    motor4.setSpeed(speedSet);
    delay(5);
  }
}

void turnRight() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}

void turnLeft() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
void forward()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
}

void back()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
}

void left()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
}

void right()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
}

void Stop()
{
  motor1.setSpeed(0);
  motor1.run(RELEASE);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
  motor3.setSpeed(0);
  motor3.run(RELEASE);
  motor4.setSpeed(0);
  motor4.run(RELEASE);
}
```

7. Pros and Cons

Pros:

- Portable and powerful robot with multiple beneficiary features.
- It can avoid obstacles efficiently that comes in its way.
- It can work as autonomous as well as manually.
- It can avoid accident while operated manually.
- Faster speed than other Arduino based robot cars.
- It can be used for ethical hacking, and other security purposes.
- Include built-in display with Linux OS controlled by raspberry pi.
- Powerful Wi-Fi router with untraceable IP by Tor network.
- It can Jam all available Wi-Fi networks in 10 meters.
- Open for future development and additional feature implements.

Cons:

- We have to use two different devices for controlling the movement and computing.
- Requires very powerful batteries due to the higher weight and additional features.
- It can be used for unethical purposes by the vulgar.
- For efficient and faster movement, we have to change the chassis for multiple times.

8. Conclusion

The Cybot Warrior Cyber Security Robot Car project is one of the very few outstanding and unique Arduino based projects in Bangladesh. If future development and implement is possible, it can be used to solve various real-life problems such as preventing data lose, safe internet browsing, Penetration testing, and much more purposes.