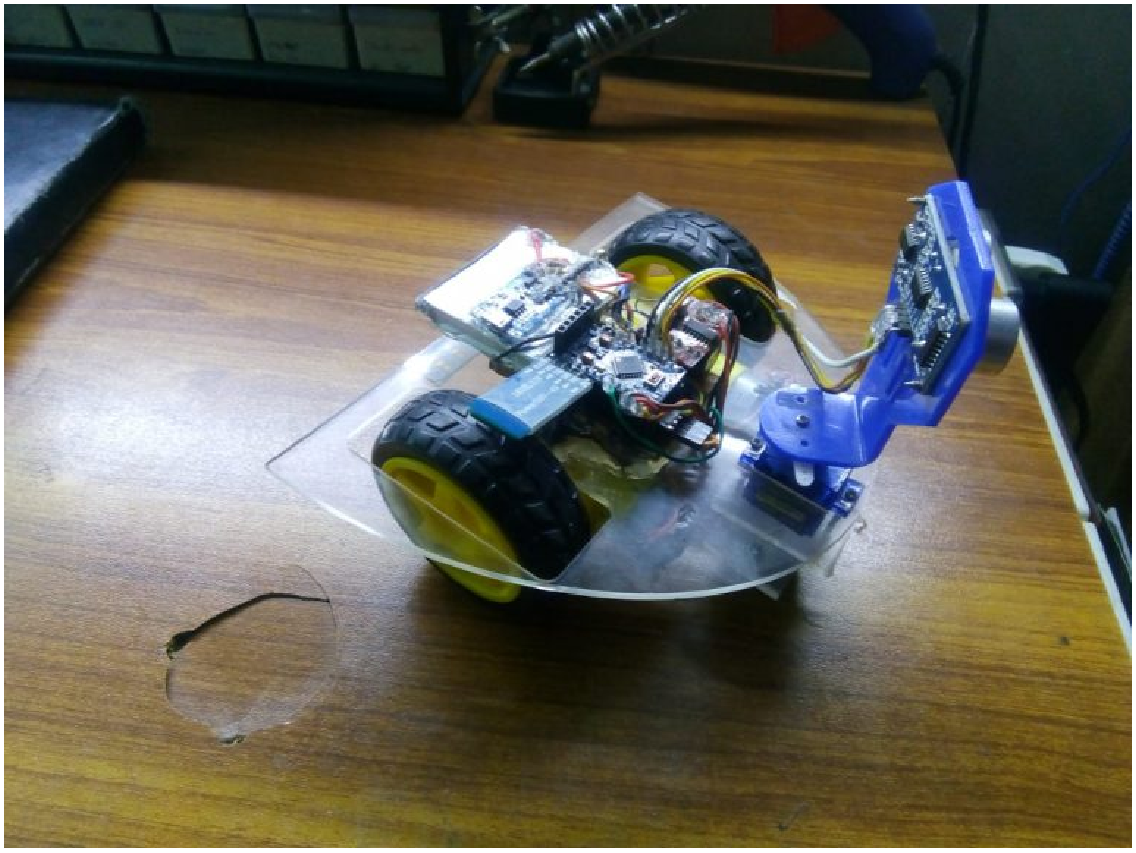
Robot Car

By **abdulsamad abdulsamad** - March 14, 2019



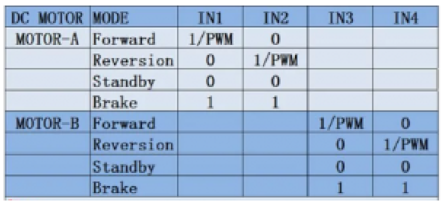
Download as PDF Print

Today lets build a robot car
First we will decide what functions we want in our car. I want following functions;

1. **Manual Control**
2. **Line following**
3. **Obstacle avoidance**
4. **Colorful LEDs animation**
5. **Buzzer**
6. **Mobile control using Bluetooth Communication**
7. **Rechargeable battery**

Manual control

Lets start with simple design of car and controlling it. For the movement of car I will be using two dc gear motor. I will use a H Bridge to control the speed and direction of motor.
Truth table for this H-Bridge(L2983D) is as follow;

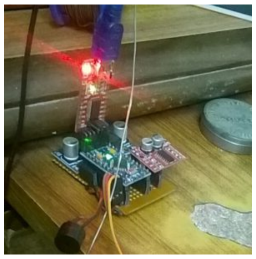


DC MOTOR	MODE	IN1	IN2	IN3	IN4
MOTOR-A	Forward	1/PWM	0		
	Reversion	0	1/PWM		
	Standby	0	0		
	Brake	1	1		
MOTOR-B	Forward			1/PWM	0
	Reversion			0	1/PWM
	Standby			0	0
	Brake			1	1

We will give this signal to H-Bridge using arduino pro mini. We will connect H-bridge to arduino pins 3,5,6 and 11 as they have pwm which we need for speed control.
In arduino IDE we will first define the pins and create a function to convert joystick values to H-bridge value according to its truth table.

Buzzer

For buzzer we will simply connect a mini buzzer to pin 7 of arduino.
It will look something like this.



Colorful RGB animation.

For RGB lights i will be using neopixel Ws281b rgb led strips. It only uses one data pin to control the entire strip. I am using pin 10 to control RGB strip. I am using adafruit neopixel library to control ths strips.

Line following

For line following i will be using two IR proximity sensor. I have a tutorial on how these sensor works, check them out if dont how its working.

https://stem.pk/what-is-infrared-sensor-ir-sensor/

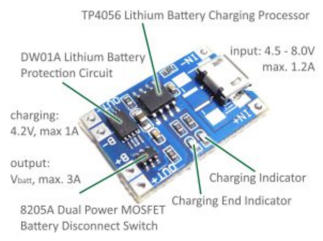
You can buy them commercially but i made my own. I will hookup them to pin 4 and 8. when one sensor is on black line it will turn in one way and when the other sensor is on black it will turn in the other way, otherwise it will go straight.

Rechargeable battery

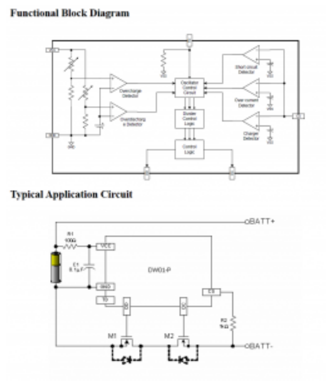
For powering the car i will be using Li-ion battery of 2000mAH and 3.7v which will be boosted to 5v using MT3608 boost converter. This battery can last for quite a long time .
For charging I will use TP4056 charging module. This module charges the battery and as well as protects the battery from over-charging, over-discharging,over-current and short circuit protection. You can use any 5v mobile charger to charge battery.
I am also adding 10k resistor between battery + terminal and analog 0 pin of arduino to later get remaining battery.

There is a small problem, when you start the car from rest position and accelerate it quickly. A huge surge current passes through this module which consider it as over-current and cut of the power, to address this problem we have to see how this module is detecting over-current.

The charging module have following functional circuit;


TP4056 Lithium Battery Charging Processor
DW01A Lithium Battery Protection Circuit
input: 4.5 - 8.0V max. 1.2A
charging: 4.2V, max 1A
output: Vbatt, max. 3A
Charging Indicator
Charging End Indicator
8205A Dual Power MOSFET Battery Disconnect Switch

By looking at these pictures we came to know that charging module have Dw01A protection chip, which senses voltage drop across power mosfets at CS pin. So we will add a 1k resistor at CS pin GND. This will make a voltage divider and will lower the sensing voltage at higher currents.
You can also check this forum.



https://electronics.stackexchange.com/questions/272947/remove-over-current-protection-of-battery-protection-circuit

Obstacle avoiding

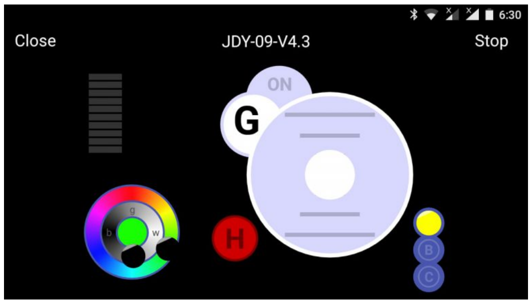
To detect obstacle we will need a ultrasonic sensor which uses sonar to detect objects. We will mount it on servo motor so that we get 180° of view. In this mode normally car will move straight until it detects some object. when it detects some object, it will stop look left and right and will turn to the position it see no object.

Another problem occurs when we use servo. The problem is servo does not work with neopixel. Why that's is happening? for that we have to see how these libraries work. Both libraries uses 16 bit timer of the arduino to make time interrupts which are required for normal operation of servo and neo pixel and unfortunately arduino pro mini which uses Atmega328p only have one 16 bit timer *i.e* timer 1 that means both libraries conflict with each other. To solve this problem we have to use TicoServo library for servo which uses pwm mode of timer 1 to generate signal required for servo.

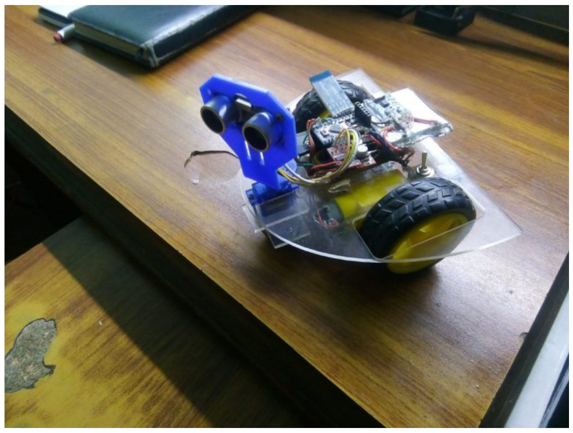
If you want details on this topic you can check this link.
https://learn.adafruit.com/neopixels-and-servos/overview

Mobile control

For mobile control we want an app to communicate to arduino using bluetooth. I will be using RemoteXY for that. You can find it on Play Store.
GUI of app will be made on RemoteXY website. We will add joystick to control car, bar to show remaining battery, slide switch to select current driving mode, a button for buzzer and finally a color wheel for RGB lights. Of course you can add according to your taste



and our beautiful car is ready.



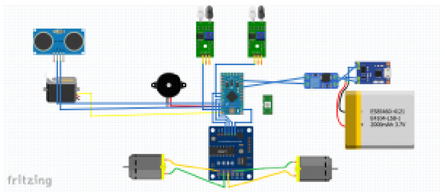
Complete code is here;

```
#include <Ultrasonic.h>
#include <Adafruit_TiCoServo.h>
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN       10
int irsensor1 = 4;
int irsensor2 = 8;
//define right motor control pins
#define int1 3
#define int2 5

//define left motor control pins
#define int3 6
#define int4 11

//define two arrays with a list of pins for each motor
uint8_t RightMotor[2] = {int1, int2};
uint8_t LeftMotor[2] = {int3, int4};

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      11

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send
signals.
// Note that for older NeoPixel strips you might need to change the third parameter--see the
strandtest
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

//////////////////////////////////////////////
//        RemoteXY include library          //
//////////////////////////////////////////////

// RemoteXY select connection mode and include library
#define REMOTEXY_MODE__HARDSERIAL

#include <RemoteXY.h>

// RemoteXY connection settings
#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 9600


// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,7,0,1,0,46,0,8,24,0,
  5,52,45,9,43,43,31,209,24,3,
  3,88,35,10,26,94,14,6,0,3,
  33,25,25,24,14,1,0,36,41,12,
  12,35,33,72,0,66,0,4,4,9,
  21,77,24 };

// this structure defines all the variables of your control interface
struct {

    // input variable
  int8_t joystick_x; // =-100..100 x-coordinate joystick position
  int8_t joystick_y; // =-100..100 y-coordinate joystick position
  uint8_t slider; // =0 if select position A, =1 if position B, =2 if position C, ...
  uint8_t rgb_r; // =0..255 Red color value
  uint8_t rgb_g; // =0..255 Green color value
  uint8_t rgb_b; // =0..255 Blue color value
  uint8_t button; // =1 if button pressed, else =0

    // output variable
  int8_t battery; // =0..100 level position

    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)

```

```
////////////////////////////////////////////
//           END RemoteXY include          //
////////////////////////////////////////////

#define SERVO_PIN    9
#define SERVO_MIN 550 // 1 ms pulse
#define SERVO_MAX 2150 // 2 ms pulse
Adafruit_TiCoServo servo;

int n=0;
long unsigned battery_volts=0;
int battery_pin = A3;
#define PIN_BUTTON 7

Ultrasonic ultrasonic(12, 13);
int distancel;
int distancer;
int distancef;
int stuck = false;

void setup()
{
  RemoteXY_Init ();
  pixels.begin();
  servo.attach(SERVO_PIN, SERVO_MIN, SERVO_MAX);
  pinMode (PIN_BUTTON, OUTPUT);
  pinMode (int1, OUTPUT);
  pinMode (int2, OUTPUT);
  pinMode (int3, OUTPUT);
  pinMode (int4, OUTPUT);
  pinMode (irsensor1,INPUT);
  pinMode (irsensor2,INPUT);
  RemoteXY.battery = map(analogRead(battery_pin), 500 , 890 ,0 , 100); //read battery voltage
}

void loop()
{
  RemoteXY_Handler ();   // use the RemoteXY structure for data transfer
  digitalWrite(PIN_BUTTON, (RemoteXY.button==0)?LOW:HIGH);   //turn buzzer on or off
  servo.write(1350); //set servo postion at centre
  //taking multiple readings for average readings of battery voltage
  battery_volts = analogRead(battery_pin) + battery_volts;
  n++;
  if(n==1000){
  battery_volts = battery_volts/1000;
  n=0;
  RemoteXY.battery = map(battery_volts, 500 , 890 ,0 , 100);
  battery_volts=0;
  }
  //setting rgb color
  for(int i=0;i<NUMPIXELS;i++){
    // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
    pixels.setPixelColor(i, pixels.Color(RemoteXY.rgb_r,RemoteXY.rgb_g,RemoteXY.rgb_b));
    pixels.show(); // This sends the updated pixel color to the hardware.
  }
   //selecting modes for driving
    if(RemoteXY.slider == 0){
                  if(RemoteXY.joystick_y > 0){
  jsteering (RightMotor, RemoteXY.joystick_y + RemoteXY.joystick_x);
  jsteering (LeftMotor, RemoteXY.joystick_y - RemoteXY.joystick_x);
                  }
        else if(RemoteXY.joystick_y < 0){
   jsteering (RightMotor, RemoteXY.joystick_y - RemoteXY.joystick_x);
  jsteering (LeftMotor, RemoteXY.joystick_y + RemoteXY.joystick_x);
          }
          else{
     digitalWrite (RightMotor [0], LOW);
     digitalWrite (RightMotor [1], LOW);
     digitalWrite (LeftMotor [0], LOW);
     digitalWrite (LeftMotor [1], LOW);
            }
  }
  else if(RemoteXY.slider == 1){
   irsteering();
    }
    else if(RemoteXY.slider == 2){
       ussteering();
      }
}
```

```

void jsteering (uint8_t * motor, int v) // v = motor speed, motor = pointer to an array of pins
{
  if (v > 100) v=100;
  if (v < -100) v=-100;
  if (v > 0){
    analogWrite (motor [0],  v * 2.55);
    digitalWrite (motor [1], LOW);
  }
  else if ( v<0 ){
    digitalWrite (motor [0], LOW);
    analogWrite (motor [1], (-v) * 2.55);
  }
  else{
    digitalWrite (motor [0], LOW);
    digitalWrite (motor [1], LOW);
    analogWrite (motor [2], 0);
  }
}

void irsteering(){
  if(digitalRead(irsensor1)==HIGH && digitalRead(irsensor2)==HIGH)
 {


     digitalWrite(int2, 0);
    digitalWrite(int4, 0);
    analogWrite(int1,100);
    analogWrite(int3,100);
 }
  else if(digitalRead(irsensor1)==HIGH)
 {
    digitalWrite(int2, 0);
    digitalWrite(int4, 0);
    analogWrite(int1,50);
    analogWrite(int3,100);
 }
 else if(digitalRead(irsensor2)==HIGH)
 {
  digitalWrite(int2, 0);
    digitalWrite(int4, 0);
    analogWrite(int1,100);
    analogWrite(int3,50);
 }
 else{
    digitalWrite(int2, 0);
    digitalWrite(int4, 0);
    digitalWrite(int1,0);
    digitalWrite(int3,0);
  }
}

void ussteering(){
   servo.write(1350);
   if(stuck == true){
      delay(500);
      stuck = false;
    }
  distancef = ultrasonic.read();
  if (distancef < 20){
    stuck = true;
    analogWrite(int1, 0);
    analogWrite(int2, 0);
    analogWrite(int3, 0);
    analogWrite(int4, 0);
      delay(500);
      servo.write(550);
      delay(500);
      distancel = ultrasonic.read();
      servo.write(2150);
      delay(500);
      distancer = ultrasonic.read();
      if(distancel > distancer && distancel > 20){
        digitalWrite(int1, LOW);
           digitalWrite(int4, LOW);
           analogWrite(int3,200);
           analogWrite(int2,200);
             delay(50);
        }
        else if(distancer > distancel && distancer > 20){
           digitalWrite(int2, LOW);
        digitalWrite(int3, LOW);
        analogWrite(int1,200);
        analogWrite(int4,200);
        delay(50);
          }
          else{
            digitalWrite(int2, LOW);
            digitalWrite(int3, LOW);
            analogWrite(int1,200);
            analogWrite(int4,200);
            delay(100);
            }
  }
  else if (distancef > 20){
    analogWrite(int2, 0);
    analogWrite(int4, 0);
    analogWrite(int1,200);
    analogWrite(int3,200);
    }
  }
```

schematic looks something like this



Here is where i got my parts from

2xDC gear motors https://hallroad.org/smart-robot-car-plastic-tire-tyre-wheel-dc-3-6v-gear-motor-set.html
2xIR sensor https://hallroad.org/ir-infrared-obstacle-avoidance-sensor-module-in-pakistan.html
Buzzer https://hallroad.org/12v-5v-piezo-buzzer-active-buzzer-continous-beep.html
Servo SG90 https://hallroad.org/towerpro-sg90-sg-90-180-degree-degree-servo-motor-in-pakistan.html
Neopixel RGB strip https://hallroad.org/dc5v-ws2812b-5-meter-60-led-strip-addressable-leds-in-pakistan-en.html
Ultrasonic sensor https://hallroad.org/hc-sr04-hc-sr04-ultrasonic-sensor-ultrasonic-distance-sensor-in-pakistan.html
Ultrasonic sensor bracket https://hallroad.org/ultrasonic-distance-sensor-mounting-bracket-in-pakistan.html
Arduino pro mini https://hallroad.org/arduino-pro-mini-in-pakistan.html
Mini H-Bridge https://hallroad.org/1.5a-mini-dual-channel-dc-motor-driver-module-l298n-pwm-speed-controller-in-pakistan.html
Battery 2000mAh 3.7v https://hallroad.org/2000mah-3.7v-lithium-ion-battery-li-ion-battery-with-jst2.0-connector-in-pakistan.html
MT3608 Boost converter https://hallroad.org/mt3608-2a-max-dc-dc-step-up-power-module-booster-power-module-in-pakistan.html
TP4056 charging and protection module https://hallroad.org/3.7v-li-ion-charger-module-1a-with-battery-protection-tp4056-in-pakistan.html
Bluetooth HC-05 https://hallroad.org/hc-05-bluetooth-module-in-pakistan.html
Toggle switch https://hallroad.org/toggle-switch-spdt-on-off-in-pakistan.html
Caster Wheel https://hallroad.org/metal-ball-caster-wheel-20mm-metal-ball-for-robot.html
Robot Car chassis https://hallroad.org/2wd-round-robot-car-chassis-in-pakistan.html

abdulsamad abdulsamad