

Arduino

Introduction

What is Arduino?

Arduino is an open-source platform. It consists of hardware (a development board) and a piece of software (IDE) intended for anyone making interactive projects. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

Arduino IDE

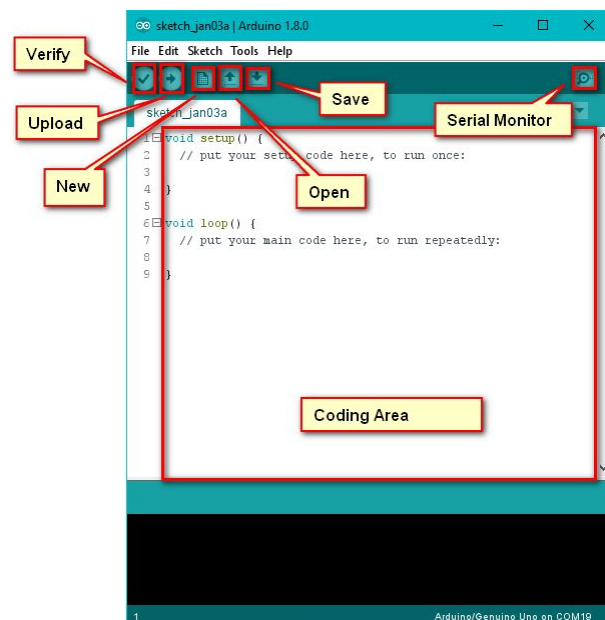
Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text.

Arduino Program Structure:

```
void setup()
{
    // runs once
}

void loop()
{
    // repeats
}
```

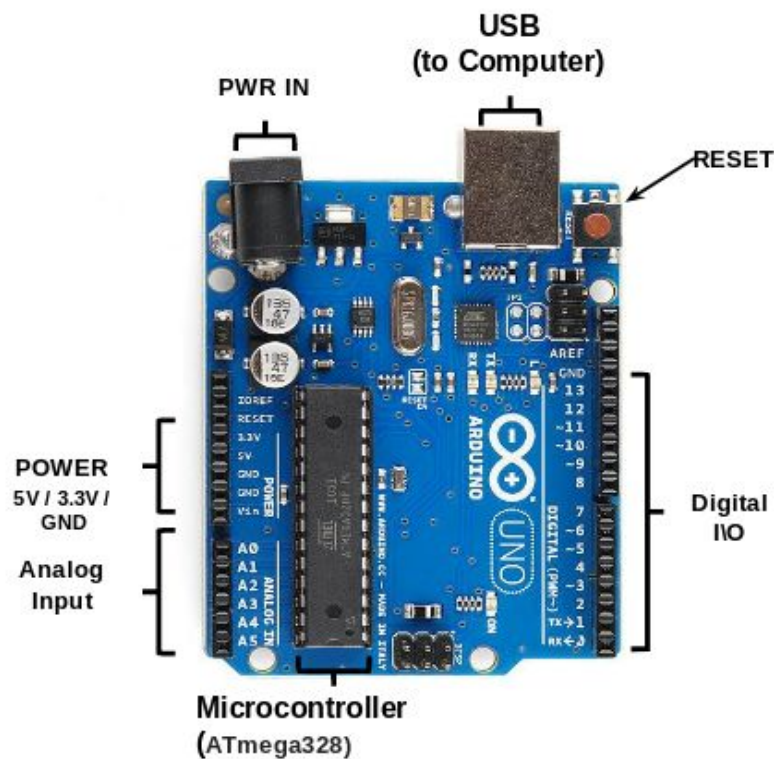
■ ■ ■



Arduino

Arduino Board

The Board



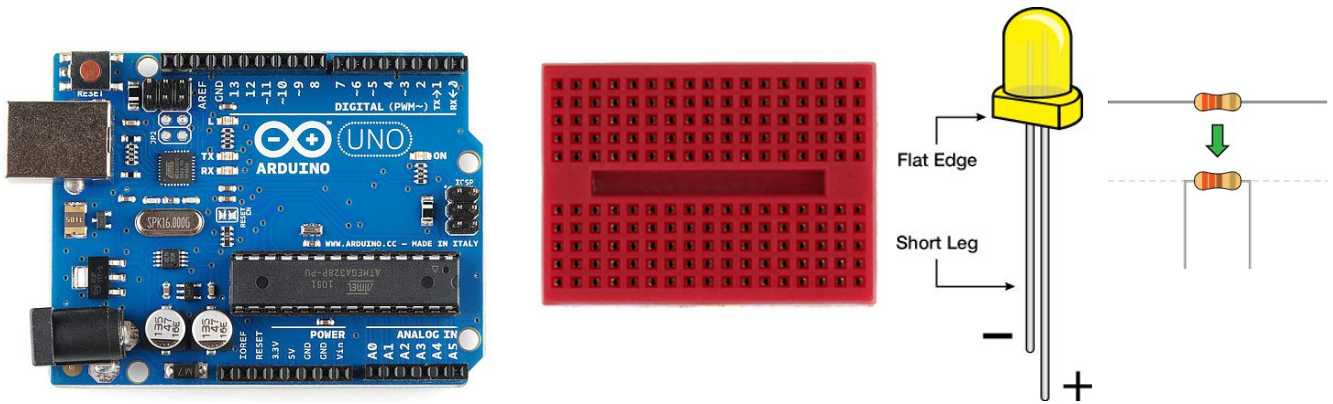
The board is a printed circuit board designed to facilitate work with a particular microcontroller. A microcontroller is a small computer on a single chip containing a processor, memory and input/output.



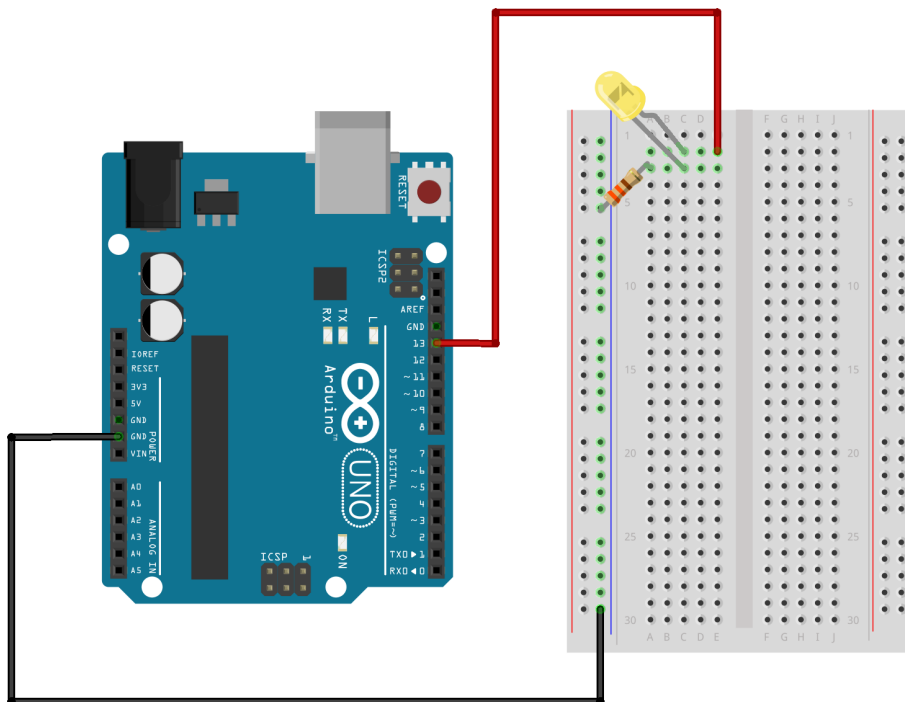
Arduino

LED Blinking

Things Needed



Circuit Diagram



The Code



```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

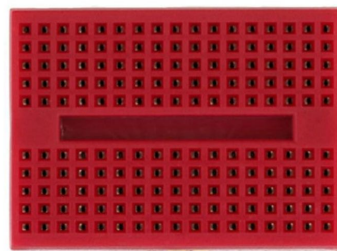
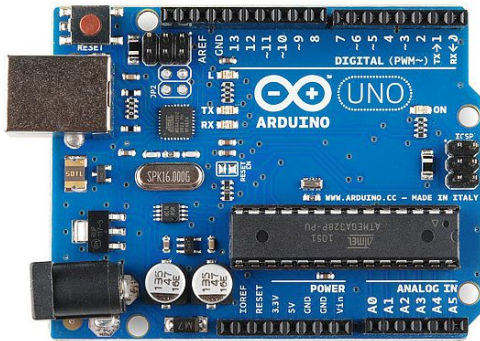
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

20 Arduino/Genuino Uno on /dev/cu.usbmodem1411

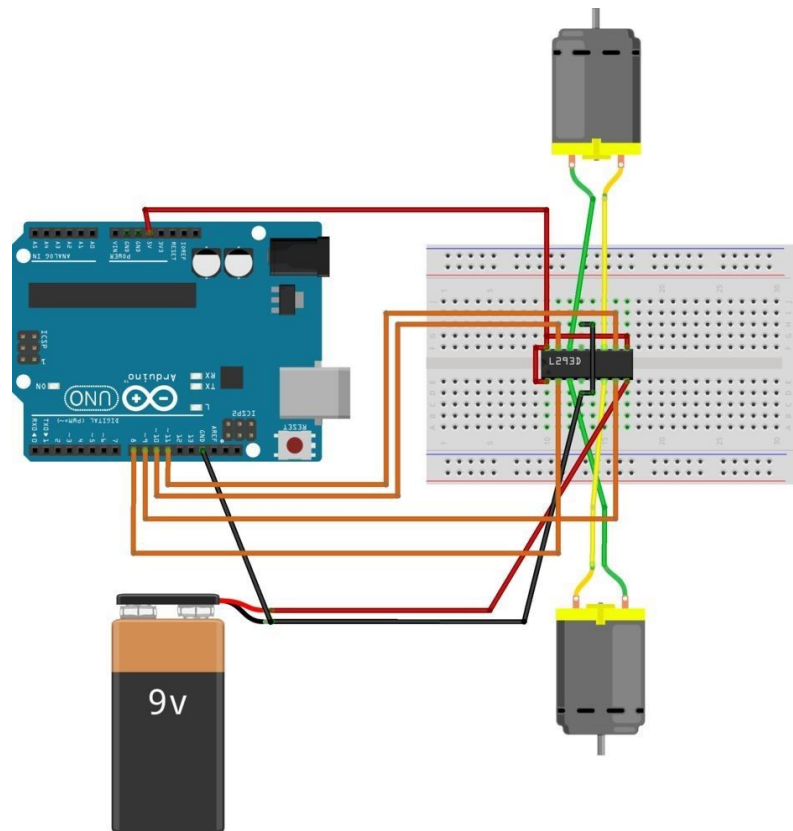
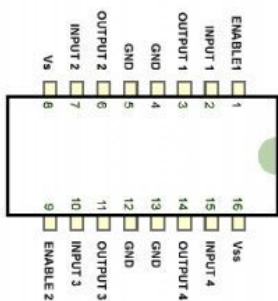
Arduino

Motor Movement

Things Needed



Circuit Diagram



fritzing

The Code

```
Thu Jul 18, 16:52:30
L293D_IC | Arduino 1.8.8
File Edit Sketch Tools Help
L293D_IC $
const int leftForward = 8;
const int leftBackward = 9;
const int rightForward = 10;
const int rightBackward = 11;

void setup()
{
  pinMode(leftForward , OUTPUT);
  pinMode(leftBackward , OUTPUT);
  pinMode(rightForward , OUTPUT);
  pinMode(rightBackward , OUTPUT);
}

void loop()
{
  //go forward
  digitalWrite(leftForward , HIGH);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , HIGH);
  digitalWrite(rightBackward , LOW);
  delay(5000);

  //go backward for 5sec
  digitalWrite(leftForward , LOW);
  digitalWrite(leftBackward , HIGH);
  digitalWrite(rightForward , LOW);
  digitalWrite(rightBackward , HIGH);
  delay(5000);

  //go Left for 5sec
  digitalWrite(leftForward , HIGH);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , LOW);
  digitalWrite(rightBackward , LOW);
  delay(5000);

  //go Right for 5sec
  digitalWrite(leftForward , LOW);
  digitalWrite(leftBackward , HIGH);
  digitalWrite(rightForward , HIGH);
  digitalWrite(rightBackward , LOW);
  delay(5000);
}
```

```
Thu Jul 18, 16:52:37
L293D_IC | Arduino 1.8.8
File Edit Sketch Tools Help
L293D_IC $
const int leftForward = 8;
const int leftBackward = 9;
const int rightForward = 10;
const int rightBackward = 11;

void setup()
{
  pinMode(leftForward , OUTPUT);
  pinMode(leftBackward , OUTPUT);
  pinMode(rightForward , OUTPUT);
  pinMode(rightBackward , OUTPUT);
}

void loop()
{
  //go forward
  digitalWrite(leftForward , HIGH);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , HIGH);
  digitalWrite(rightBackward , LOW);
  delay(5000);

  //go backward for 5sec
  digitalWrite(leftForward , LOW);
  digitalWrite(leftBackward , HIGH);
  digitalWrite(rightForward , LOW);
  digitalWrite(rightBackward , HIGH);
  delay(5000);

  //go Left for 5sec
  digitalWrite(leftForward , HIGH);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , LOW);
  digitalWrite(rightBackward , LOW);
  delay(5000);

  //go Right for 5sec
  digitalWrite(leftForward , LOW);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , HIGH);
  digitalWrite(rightBackward , LOW);
  delay(5000);

  //Stop for 5sec
  digitalWrite(leftForward , LOW);
  digitalWrite(leftBackward , LOW);
  digitalWrite(rightForward , LOW);
  digitalWrite(rightBackward , LOW);
  delay(5000);
}
```

■ ■ ■

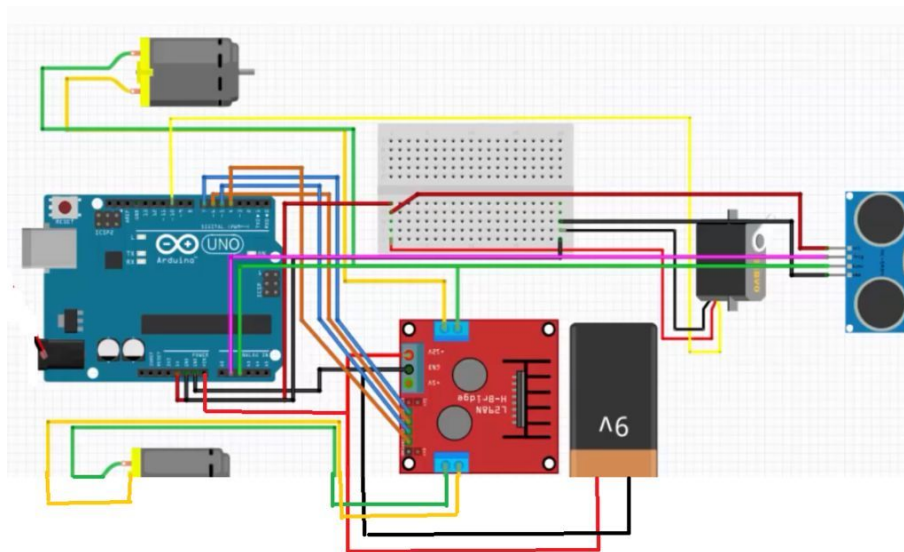
Arduino

Obstacle Avider

Things Needed

1. Arduino UNO
2. Smart robot car chassis with 2 x toy car wheels and 1 x Universal wheel (or ball casters)
3. Two DC motors
4. L298n motor driver
5. HC-SR04 Ultrasonic Sonar sensor
6. TowerPro micro servo 9g
7. Mini breadboard
8. Ultrasonic sonar sensor mounting bracket
9. Hot glue gun(optional)

Circuit Diagram



For more details on the circuit diagram and the making, please refer:
<https://www.instructables.com/id/Obstacle-Avoiding-Robot-Arduino-1/>

The Code

Download and paste NewPing library (Ultrasonic sensor function library) file to the Arduino libraries folder.

- Download NewPing here - <https://github.com/JRodrigoTech/Ultrasonic-HC-SR04...>
- Paste files to the path - C:\Arduino\libraries



```
obstacle_avoiding | Arduino 1.8.8
File Edit Sketch Tools Help
obstacle_avoiding

#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup(){
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  servo_motor.attach(10); //our servo pin

  servo_motor.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}

1 Arduino/Genuino Uno on /dev/ttyACM0
```



```
obstacle_avoiding | Arduino 1.8.8
File Edit Sketch Tools Help
obstacle_avoiding

distance = readPing();
delay(100);
}

void loop(){
  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 20){
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);

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

int lookRight(){
  servo_motor.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
}

1
```



```
obstacle_avoiding

int lookRight(){
  servo_motor.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
}

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

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

void moveStop(){
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){
  if(!goesForward){
    goesForward=true;
  }
}
```

```
obstacle_avoiding

void moveForward(){
  if(!goesForward){
    goesForward=true;
    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);
    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
  }
}

void moveBackward(){
  goesForward=false;
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);
}

void turnRight(){
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}

void turnLeft(){
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
```

```
obstacle_avoiding

digitalWrite(RightMotorForward, LOW);
}

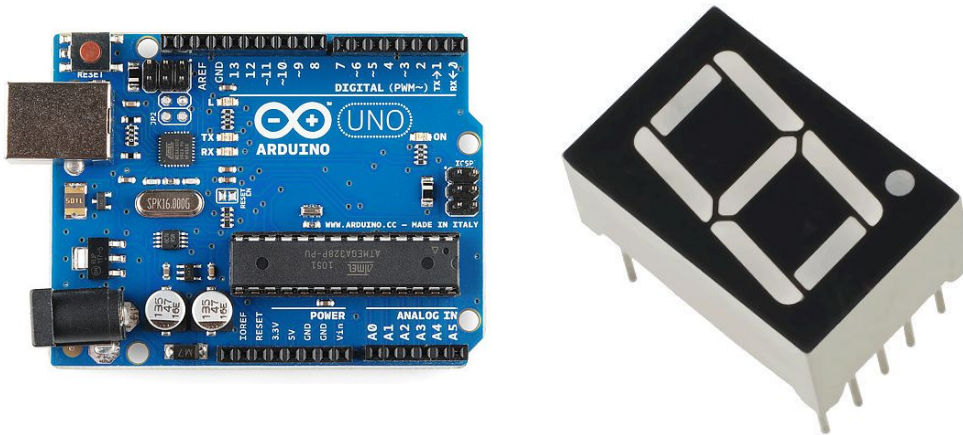
void turnRight(){
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}

void turnLeft(){
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
```

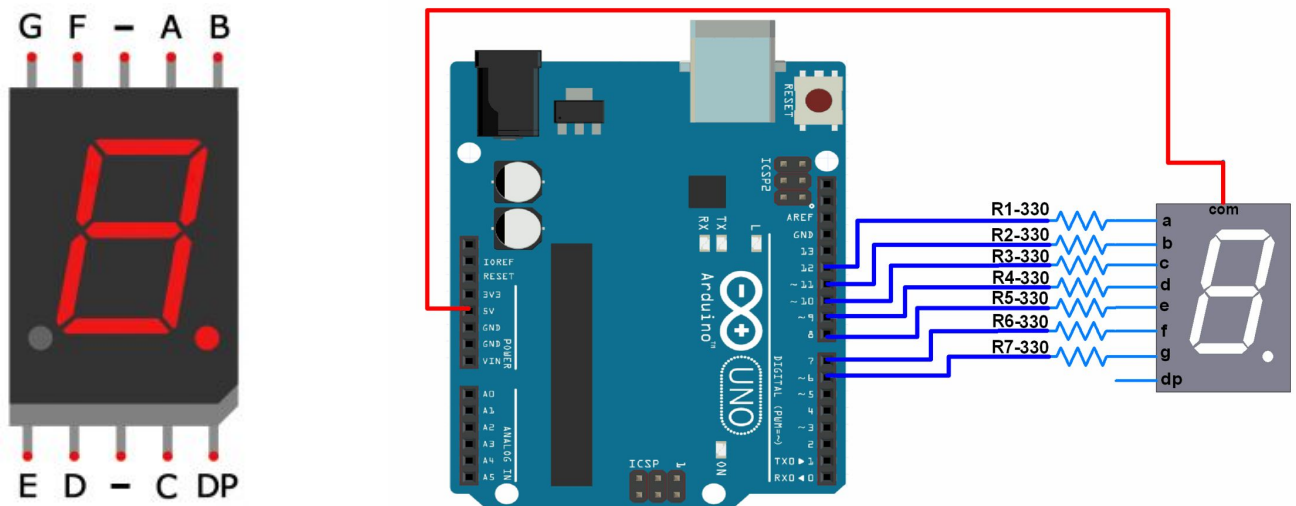
Arduino

7 Segment Display

Things Needed



Circuit Diagram



The Code

```
7_segment_display $
int disp_pin[7]; /* array for a-g pins of 7-Segment display */

void define_segment_pins(int a, int b, int c, int d, int e, int f, int g)
/* Assigns 7-segment display pins to board */
{
    disp_pin[0] = a;
    disp_pin[1] = b;
    disp_pin[2] = c;
    disp_pin[3] = d;
    disp_pin[4] = e;
    disp_pin[5] = f;
    disp_pin[6] = g;
}

void display_number(int num) /* Function for displaying number (0-9) */
{
    switch(num)
    {
        case 0:
            digitalWrite(disp_pin[0], LOW); /* Drive disp_pin[0] to LOW */
            digitalWrite(disp_pin[1], LOW);
            /* Driving LOW turns on LED segment for common anode display */
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], LOW);
            digitalWrite(disp_pin[4], LOW);
            digitalWrite(disp_pin[5], LOW);
            digitalWrite(disp_pin[6], HIGH);
            break;
        case 1:
            digitalWrite(disp_pin[0], HIGH); /* Drive disp_pin[0] to HIGH */
            digitalWrite(disp_pin[1], LOW);
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], HIGH);
            /* Driving HIGH turns off LED segment for common anode display */
            digitalWrite(disp_pin[4], HIGH);
            digitalWrite(disp_pin[5], HIGH);
            digitalWrite(disp_pin[6], HIGH);
            break;
        case 2:
            digitalWrite(disp_pin[0], LOW);
            digitalWrite(disp_pin[1], LOW);
            digitalWrite(disp_pin[2], HIGH);
            digitalWrite(disp_pin[3], LOW);
            digitalWrite(disp_pin[4], HIGH);
            digitalWrite(disp_pin[5], LOW);
            digitalWrite(disp_pin[6], HIGH);
            break;
        case 3:
            digitalWrite(disp_pin[0], LOW);
            digitalWrite(disp_pin[1], LOW);
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], LOW);
            digitalWrite(disp_pin[4], HIGH);
            digitalWrite(disp_pin[5], HIGH);
            digitalWrite(disp_pin[6], LOW);
            break;
        case 4:
            digitalWrite(disp_pin[0], HIGH);
            digitalWrite(disp_pin[1], LOW);
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], HIGH);
            digitalWrite(disp_pin[4], HIGH);
            digitalWrite(disp_pin[5], LOW);
            digitalWrite(disp_pin[6], LOW);
            break;
        case 5:
            digitalWrite(disp_pin[0], LOW);
            digitalWrite(disp_pin[1], HIGH);
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], LOW);
            digitalWrite(disp_pin[4], HIGH);
            digitalWrite(disp_pin[5], LOW);
            digitalWrite(disp_pin[6], LOW);
            break;
        case 6:
            digitalWrite(disp_pin[0], LOW);
            digitalWrite(disp_pin[1], HIGH);
            digitalWrite(disp_pin[2], LOW);
            digitalWrite(disp_pin[3], LOW);
            digitalWrite(disp_pin[4], LOW);
            digitalWrite(disp_pin[5], LOW);
            digitalWrite(disp_pin[6], LOW);
            break;
    }
}
```

```
7_segment_display $
case 1:
case 2:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], HIGH);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], HIGH);
    digitalWrite(disp_pin[6], LOW);
    break;
case 3:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], HIGH);
    digitalWrite(disp_pin[5], HIGH);
    digitalWrite(disp_pin[6], LOW);
    break;
case 4:
    digitalWrite(disp_pin[0], HIGH);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], HIGH);
    digitalWrite(disp_pin[4], HIGH);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], LOW);
    break;
case 5:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], HIGH);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], HIGH);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], LOW);
    break;
case 6:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], HIGH);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], LOW);
    break;
}
```

```
7_segment_display 8
case 6:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], HIGH);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], LOW);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
case 7:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], HIGH);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], HIGH);
digitalWrite(disp_pin[6], HIGH);
break;
case 8:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], LOW);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
case 9:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
default:
digitalWrite(disp_pin[0], HIGH);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], LOW);
```

```
7_segment_display 8
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
default:
digitalWrite(disp_pin[0], HIGH);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], LOW);
digitalWrite(disp_pin[5], HIGH);
digitalWrite(disp_pin[6], LOW);
break;
}
}

void setup() {
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
define_segment_pins{12,11,10,9,8,7,6}; /* a-g segment pins to Arduino */
}

void loop() {
int i;
for(i = 9; i>=0; i--)
{
display_number(i);
delay(1000);
}
for(i = 0; i<=9; i++)
{
display_number(i);
delay(1000);
}
}
```

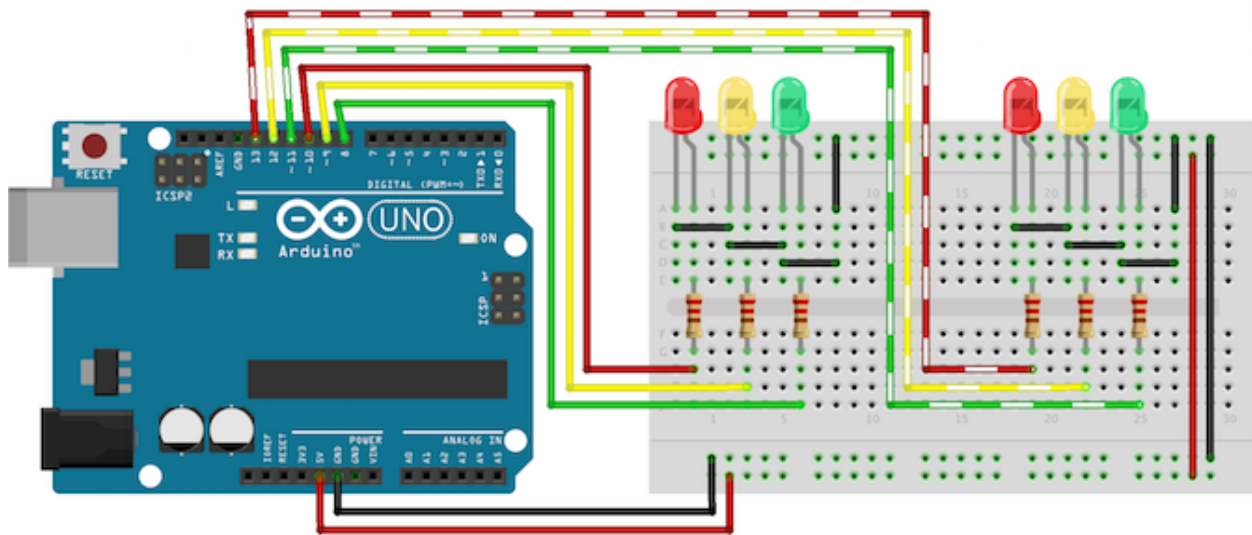
Arduino

Traffic Light

Summary

1. Arduino Uno
2. LED Lights: Green x2, Yellow x2, Red x2
3. Resistors

Circuit Diagram



The Code

```
traffic_light

// light one
int red1 = 10;
int yellow1 = 9;
int green1 = 8;

// light two
int red2 = 13;
int yellow2 = 12;
int green2 = 11;

void changeLights(){
  // turn both yellows on
  digitalWrite(green1, LOW);
  digitalWrite(green2, LOW);
  digitalWrite(yellow1, HIGH);
  digitalWrite(yellow2, HIGH);
  digitalWrite(red1, LOW);
  digitalWrite(red2, LOW);
  delay(5000);

  // turn both yellows off, and opposite green and red
  digitalWrite(yellow1, LOW);
  digitalWrite(red1, HIGH);
  digitalWrite(yellow2, LOW);
  digitalWrite(red2, LOW);
  digitalWrite(green2, HIGH);
  delay(5000);

  // both yellows on again
  digitalWrite(yellow1, HIGH);
  digitalWrite(yellow2, HIGH);
  digitalWrite(green2, LOW);
  digitalWrite(red1, LOW);
  delay(3000);

  // turn both yellows off, and opposite green and red
  digitalWrite(green1, HIGH);
  digitalWrite(yellow1, LOW);
  digitalWrite(red1, LOW);
  digitalWrite(yellow2, LOW);
  digitalWrite(red2, HIGH);
  delay(5000);
}
```

```
traffic_light

// turn both yellows off, and opposite green and red
digitalWrite(yellow1, LOW);
digitalWrite(red1, HIGH);
digitalWrite(yellow2, LOW);
digitalWrite(red2, LOW);
digitalWrite(green2, HIGH);
delay(5000);

// both yellows on again
digitalWrite(yellow1, HIGH);
digitalWrite(yellow2, HIGH);
digitalWrite(green2, LOW);
digitalWrite(red1, LOW);
delay(3000);

// turn both yellows off, and opposite green and red
digitalWrite(green1, HIGH);
digitalWrite(yellow1, LOW);
digitalWrite(red1, LOW);
digitalWrite(yellow2, LOW);
digitalWrite(red2, HIGH);
delay(5000);
}

void setup(){
  // light one
  pinMode(red1, OUTPUT);
  pinMode(yellow1, OUTPUT);
  pinMode(green1, OUTPUT);

  // light two
  pinMode(red2, OUTPUT);
  pinMode(yellow2, OUTPUT);
  pinMode(green2, OUTPUT);
}

void loop(){
  changeLights();
  delay(15000);
}
```