



Tempus

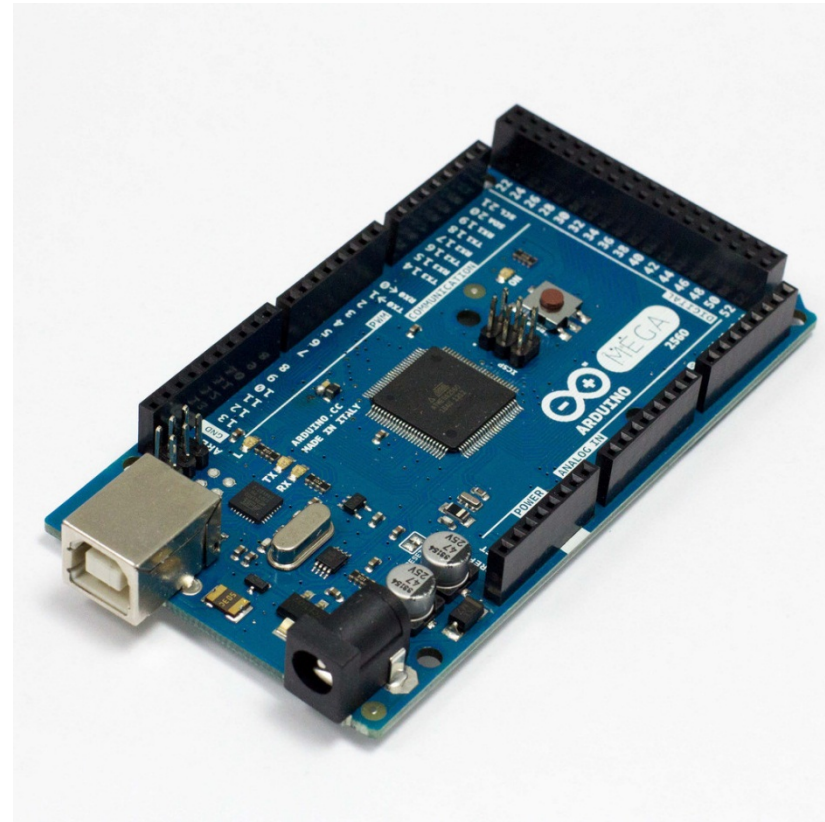
Development of Embedded System Courses with implementation
of Innovative Virtual approaches for integration of Research,
Education and Production in UA, GE, AM

Master classes for Arduino Programming



History of Arduino

Arduino is a flexible programmable hardware platform designed for artists, designers, tinkerers, and the makers of things. Arduino started its life in Italy, at Interaction Design Institute Ivera (IDI), a graduate school for interaction design. This is a specific school of design education that focuses on how people interact with digital products, systems, and environments and how they in turn influence us. This microcontroller comes from a company called Atmel and the chip is known as an AVR. It is slow in modern terms, running at only 16Mhz with an 8-bit core, and has a very limited amount of available memory, with 32 kilobytes of storage and 2 kilobytes of random access memory.





What is an Arduino?

Arduino is a tool, a community, and a way of thinking that is affecting how we use and understand technology. Arduino is tiny circuit board that has huge potential. One bit of technology that makes these dreams not only achievable but enjoyable is the Arduino, a small microcontroller board designed to be easy to learn and a breeze to program. It lets you operate motors and take input from sensors, allowing you to build the project you want to!





What is an Arduino?

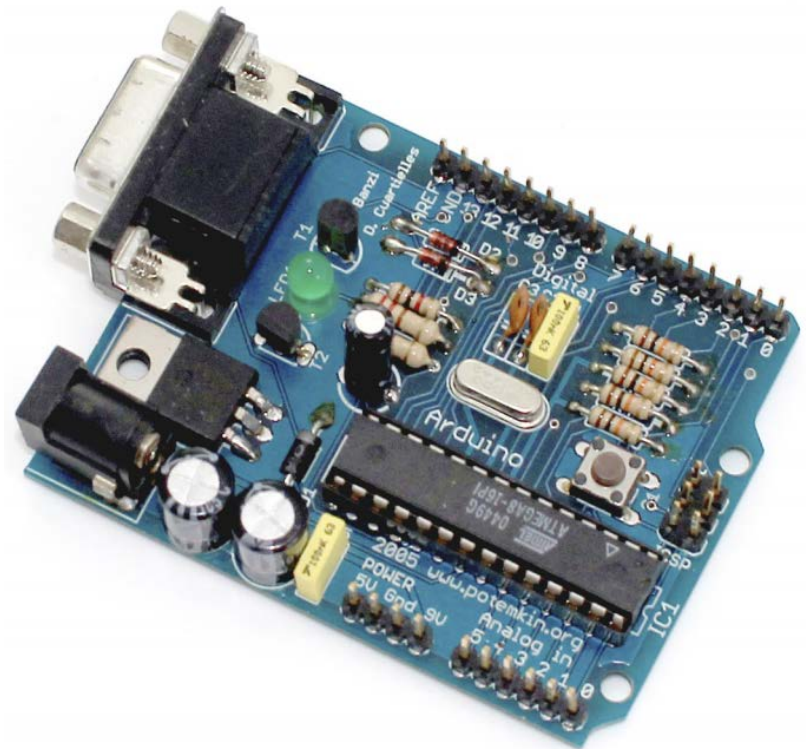
Arduino is an open-source physical computing platform designed to make experimenting with electronics more fun and intuitive. Arduino has its own unique, simplified programming language, a vast support network, and thousands of potential uses, making it the perfect platform for both beginner and advanced DIY enthusiasts.





What is an Arduino?

Arduino is made up of both hardware and software. The Arduino board is a printed circuit board (PCB) that is specifically designed to use a microcontroller chip as well as other input and outputs. It also has many other electronic components that are needed for the microcontroller to function or to extend its capabilities. Microcontrollers are small computers contained within a single, integrated circuit or computer chip, and they are an excellent way to program and control electronics. You write code in the Arduino software to tell the microcontroller what to do. For example, by writing a line of code, you can tell an LED to blink on and off. If you connect a pushbutton and add another line of code, you can tell the LED to turn on only when the button is pressed.





What is an Arduino?

Using just a few simple components, you can explore a variety of applications for Arduino and form a base on which you can build your own projects.

One of the keys to the Arduino's success is the minimal amount of time that it takes for a complete novice to move from opening the little box containing the Arduino interface board to having their first source code, also known as a sketch, up and running on that board.

The Arduino web site at www.arduino.cc provides easy-to-follow "Getting Started" tutorials and whenever you get stuck there is always the active, vocal, and generally helpful Arduino community that is willing to share its knowledge.

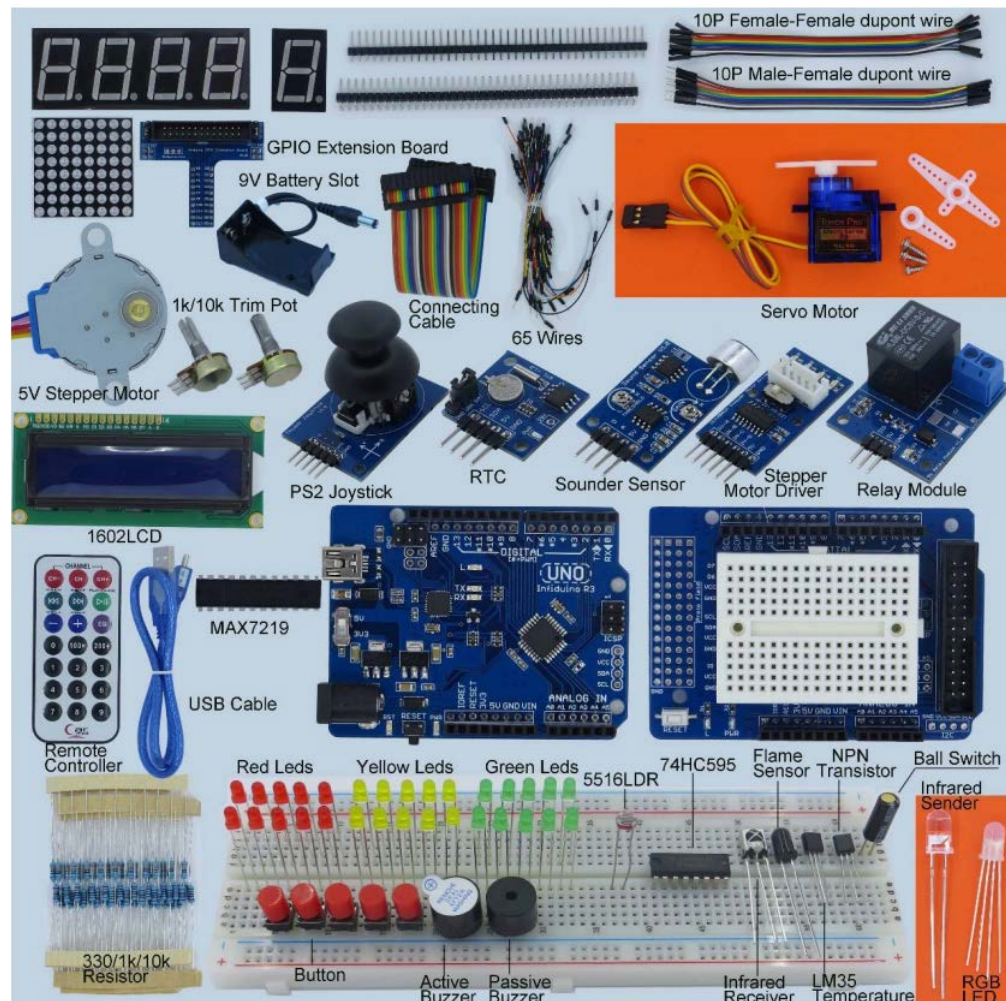




Tempus

Parts of ArduinoDesIRE

- ✓ Arduino Mega:
- ✓ USB A-B cable:
- ✓ LEDs:
- ✓ Resistors:
- ✓ Variable resistors:
- ✓ Diodes:
- ✓ Pushbuttons:
- ✓ Photo diodes:
- ✓ Temperature sensors:
- ✓ Piezo buzzer:
- ✓ Relays:
- ✓ Transistors:
- ✓ DC motors:
- ✓ Servo motors:

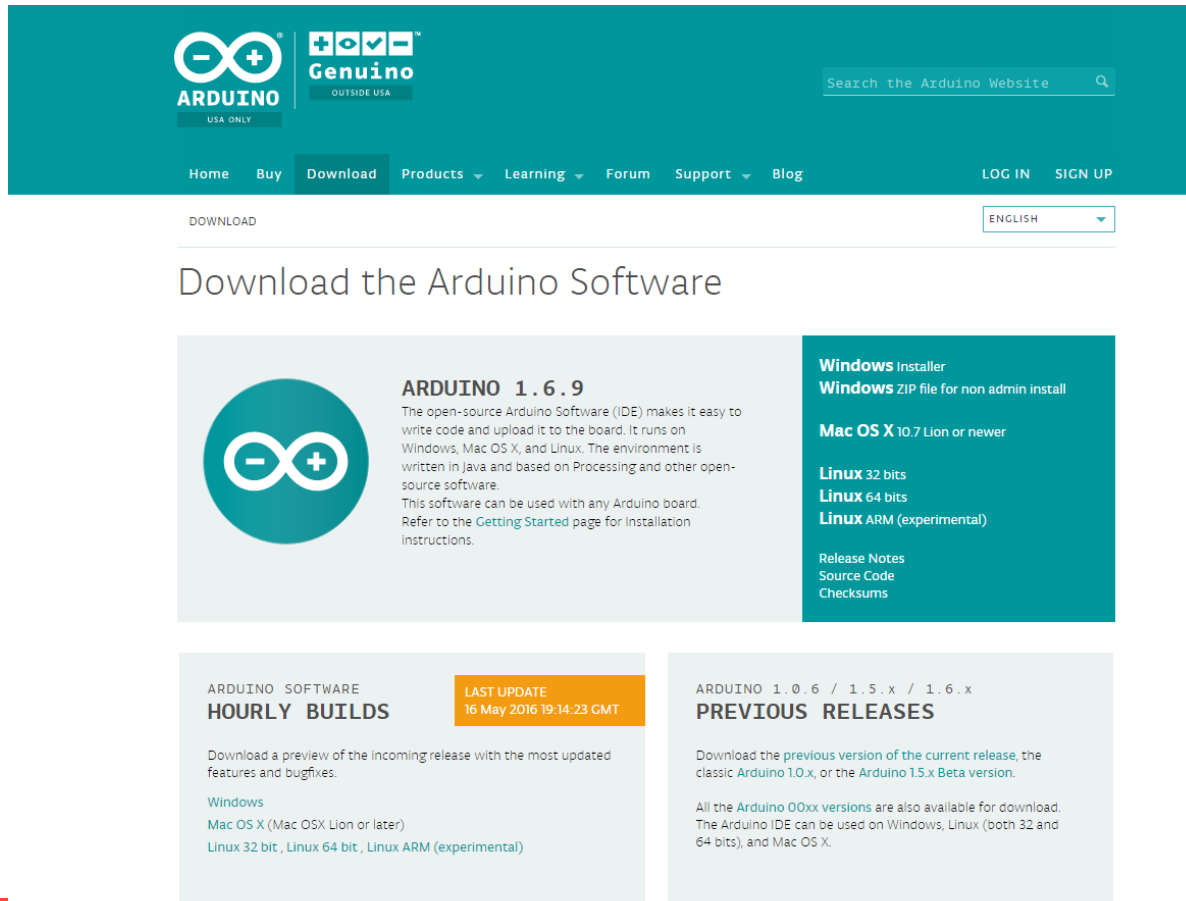




Tempus

Getting Started

website: <http://arduino.cc/>



The screenshot shows the Arduino website's download page. At the top, there's a teal header with the Arduino logo, the Genuino logo, and a search bar. Below the header, there's a navigation menu with links to Home, Buy, Download, Products, Learning, Forum, Support, and Blog. The main content area is titled "Download the Arduino Software". It features a large section for "ARDUINO 1.6.9" with a description of the software and a list of download links for Windows, Mac OS X, and Linux. Below this, there are two smaller sections: "ARDUINO SOFTWARE HOURLY BUILDS" and "ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES".

ARDUINO 1.6.9
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation Instructions.

Windows Installer
Windows ZIP file for non admin install
Mac OS X 10.7 Lion or newer
Linux 32 bits
Linux 64 bits
Linux ARM (experimental)
[Release Notes](#)
[Source Code](#)
[Checksums](#)

ARDUINO SOFTWARE HOURLY BUILDS
LAST UPDATE
16 May 2016 19:14:23 GMT
Download a preview of the incoming release with the most updated features and bugfixes.
[Windows](#)
[Mac OS X](#) (Mac OS X Lion or later)
[Linux](#) 32 bit, [Linux](#) 64 bit, [Linux](#) ARM (experimental)

ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES
Download the previous version of the current release, the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta](#) version.
All the [Arduino 00xx](#) versions are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.





Step 1: Hardware connection Connect your Arduino with PC through the USB cable.

Step 2: Install the USB driver When you do the above step for the first time, your PC should find a new hardware. Then your system will try to install the USB driver. Please browser to the folder named ".../drivers" of you Arduino IDE folder, and install the USB driver manually.

Step 3: Set up a new file. Open your Arduino IDE and set up a new file (through commands "File->New").

Step 4: Code writing The following lines shows the basic structure for writing a code in Arduino IDE using "Arduino programming language".

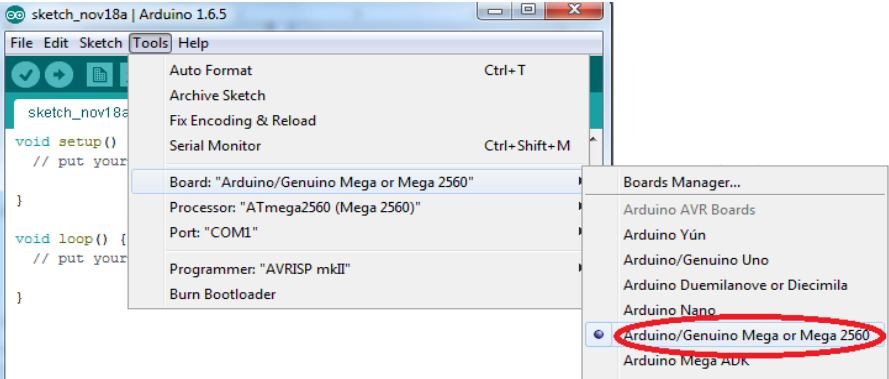
```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```



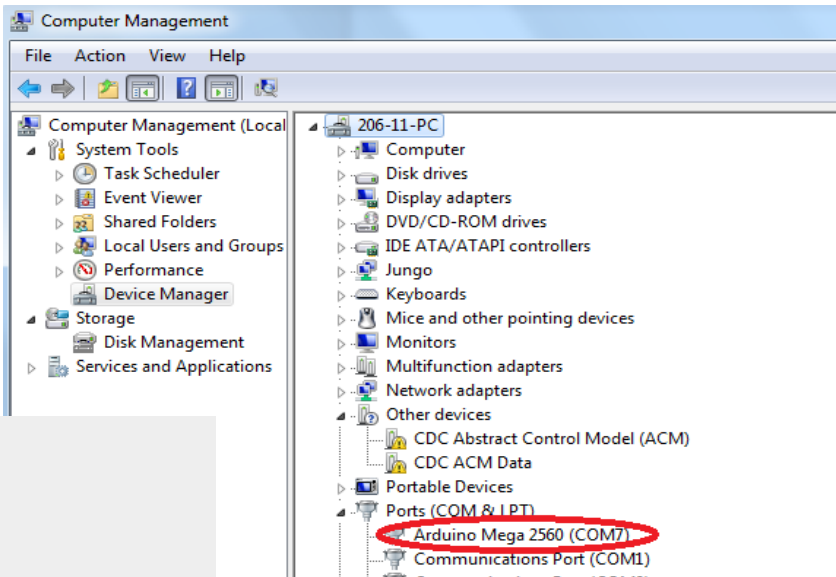
Step 5 Code downloading

Before downloading the code to Arduino you should do several things in Arduino IDE. - Select the right board. - Select the right COM port. - Compile the code you've written. - Press the download button. - Please see the following pictures for these steps.

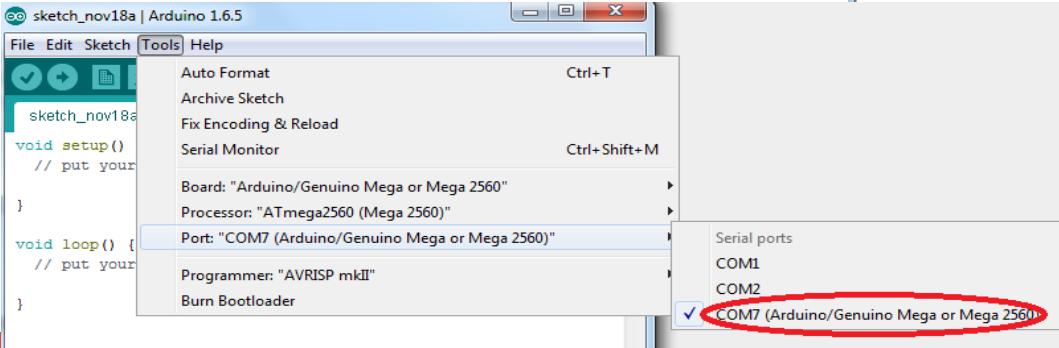
1



2



3





Components Needed:

1. Arduino board
2. 3 x LED
3. 1 x Breadboard
4. Wires

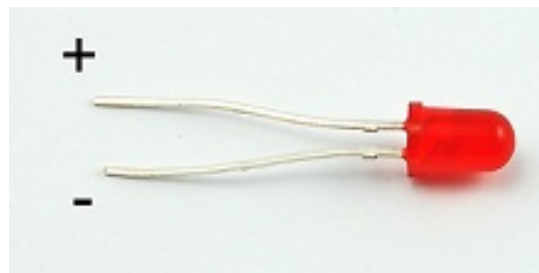


Knowledge Needed:

1. Can't be "INPUT" and "OUTPUT" at the same time. "pinMode()" function is used to set the IO as "INPUT" or "OUTPUT".
2. Arduino works under 5V. "HIGH" voltage - 5V, "Low" voltage - 0V.
3. For each digital IO port of Arduino, it can source at least 20mA current.
4. **LED** is the abbreviation of **light emitting diode**. It has two pins, one is anode, the other is cathode.

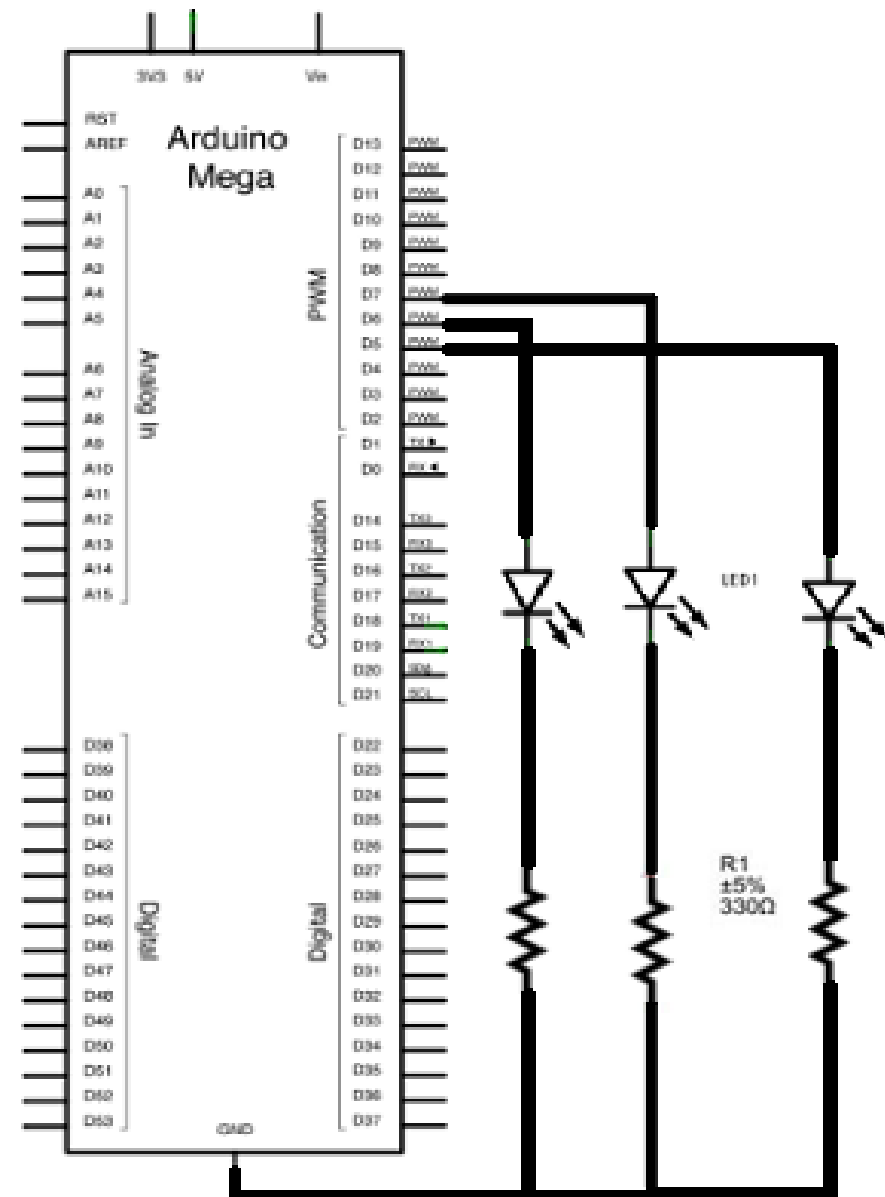
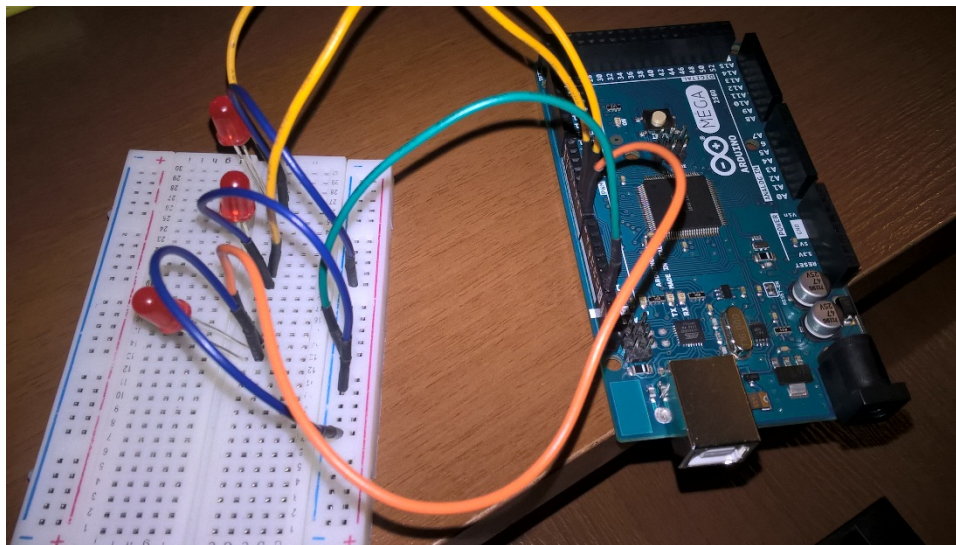
To make it light, anode should have a higher voltage than the cathode.

5. Because the IO port of Arduino can source at least 20mA current, So we can use the Digital IO to drive an LED directly. We just need 10mA current here.





Hardware Connection





Code Writing

```
void setup()
{
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop()
{
  for(int i=1; i<3; i++)
  {
    digitalWrite(5,HIGH);
    delay(1000);
    digitalWrite(6,HIGH);
    delay(1000);
    digitalWrite(7,HIGH);
    delay(1000);
  }

  for(int i=1; i<3; i++)
  {
    digitalWrite(5,LOW);
    delay(1000);
    digitalWrite(6,LOW);
    delay(1000);
    digitalWrite(7,LOW);
    delay(1000);
  }
}
```



When we push the button, the LED will be on. When the button is released, the LED will be off.

Components Needed

1. Arduino board
2. 1 x LED(Red color)
3. 1 x Button 5
4. 1 x Breadboard
5. 6. Wires

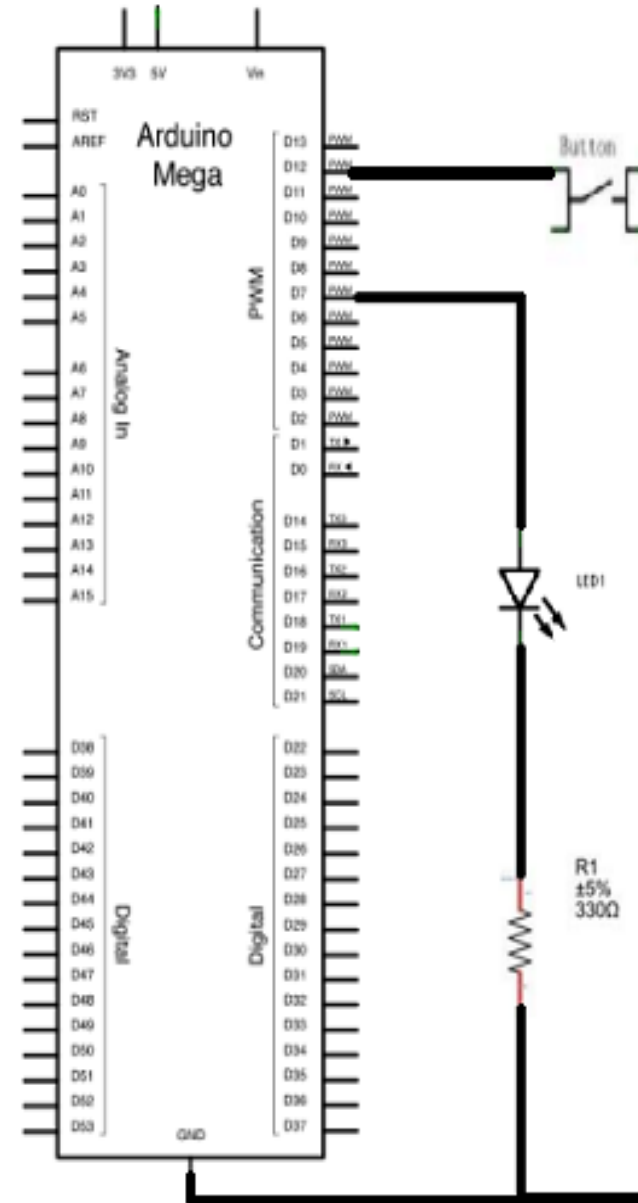
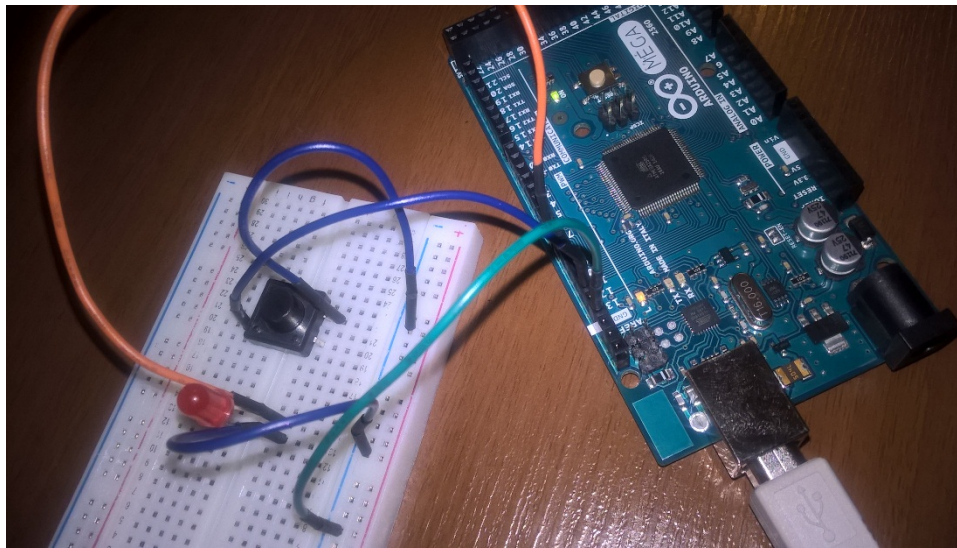


Knowledge Needed

1. An IO port of Arduino can not only output a "HIGH" or "LOW" voltage, but also it can sense the voltage level of outside. To use this function, we should set the IO as "INPUT", pinMode() function can do this task for us.
2. After the IO set as "INPUT", we can use digitalRead() function to read the external voltage level.



Hardware connection





Code Writing

```
void setup() {  
  pinMode(7, OUTPUT);  
  pinMode(12, INPUT_PULLUP);  
}  
  
void loop() {  
  int i=digitalRead(12);  
  if(i==0)  
    digitalWrite (7,HIGH);  
  else  
    digitalWrite (7,LOW);  
}
```




We will use the PWM (Pulse-with-modulation) method to change the lighting strength of an LED, make it light from dark to bright continuous.

Components Needed

1. Arduino board
2. 1 x LED(Red color)
3. 1 x Breadboard
4. Wires

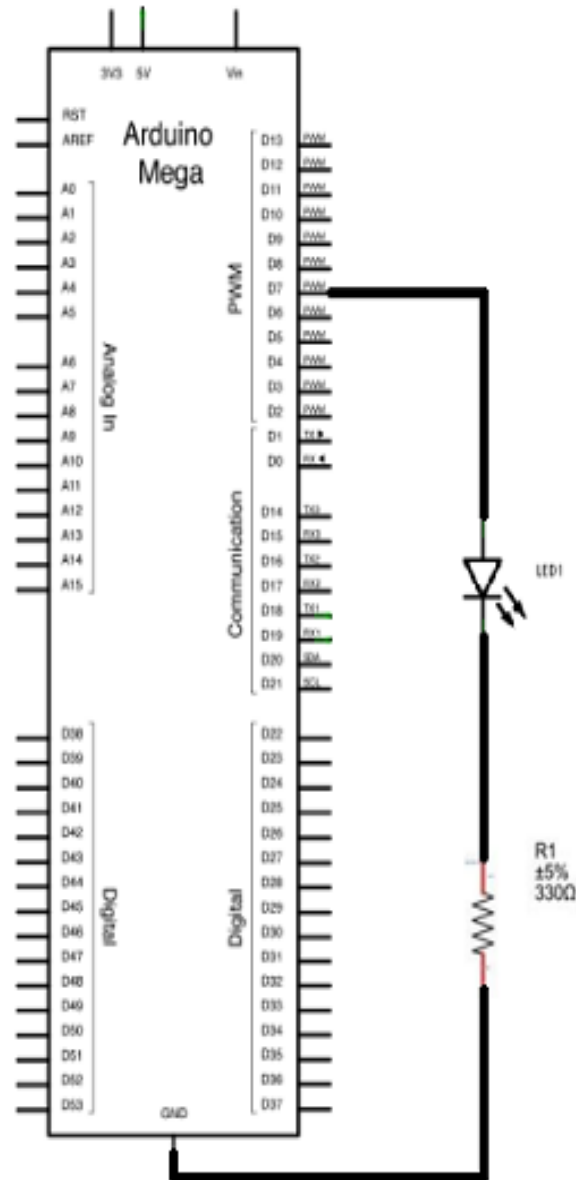
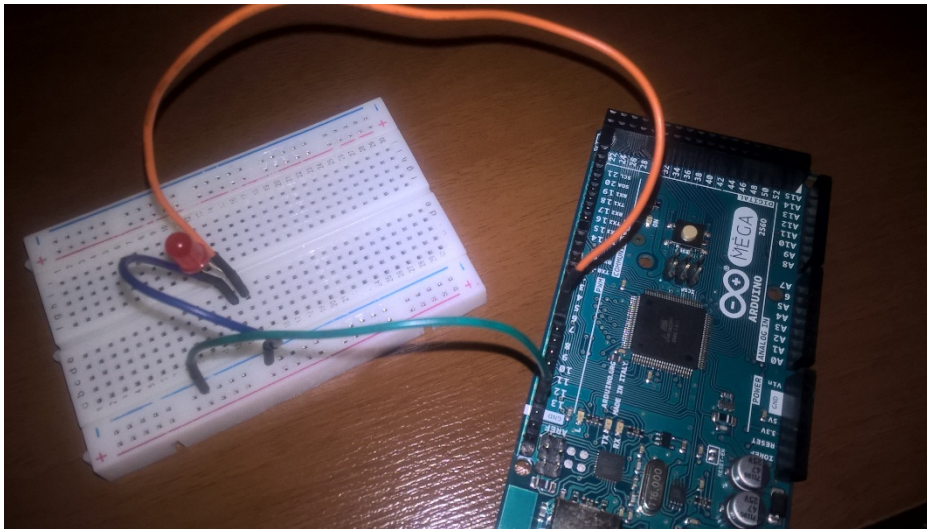


Knowledge Needed

It is a good method for getting analog results with digital means. For some power components like a resistor generating heat or an LED emitting light or other similar components, the PWM method will get a same result as when we exert an different analog voltage to the components. In this lesson we will use the `analogWrite()` to change the light strength of the LED.



Hardware Connection





Code Writing

```
void setup() {  
  pinMode(7, OUTPUT);  
}  
int i=0;  
void loop() {  
  
  if(i>=0 && i<250)  
  {  
    i+=50;  
    analogWrite(7,i);  
    delay(1000);  
  }  
  else{  
    i=0;  
    analogWrite(7,i);  
    delay(1000);  
  }  
}
```



Components Needed

1. Arduino board
2. 1 x Active Buzzer
3. 1 x Breadboard
4. Wires

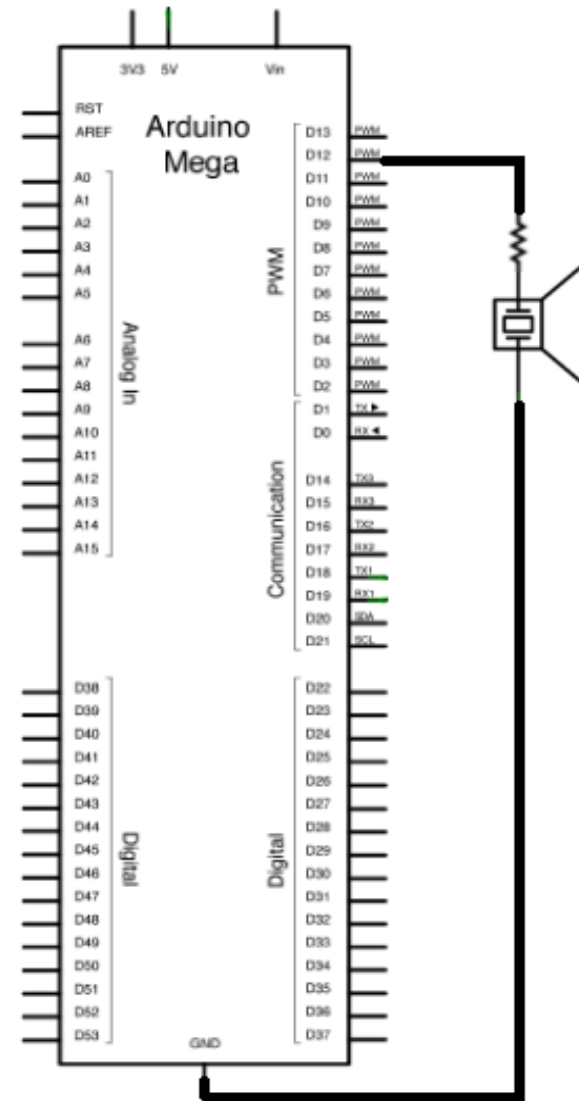
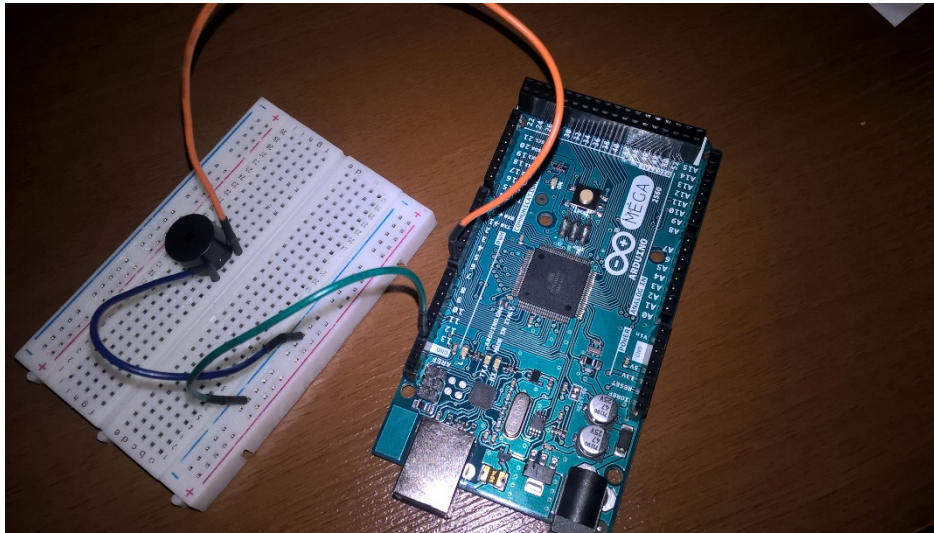


Knowledge Needed

Active Buzzer has oscillating circuit in it. When there is a rated voltage exerted on it, it will make a sound. The active buzzer in the kit has a rated voltage of 5V, and a rated current of no more than 30mA, usually an IO port can drive it directly.



Hardware Connection.





Code Writing

1

```
void setup() {  
  pinMode(12,OUTPUT);  
}  
void loop() {  
  tone(12, 277);  
  delay(500);  
}
```

2

```
void setup() {  
  pinMode(7,OUTPUT);  
  
}  
int i=0;  
  
void loop() {  
  if(i>=0 && i<250)  
  {  
    i+=25;  
    analogWrite(7,i);  
    delay(1000);  
  }  
  else{  
    i=0;  
    analogWrite(7,i);  
    delay(1000);  
  }  
  tone(7,277);  
  delay(500);  
}
```



Components Needed

1. Arduino board
2. 3 x LED(Red color)
3. 1 x Breadboard
4. Wires



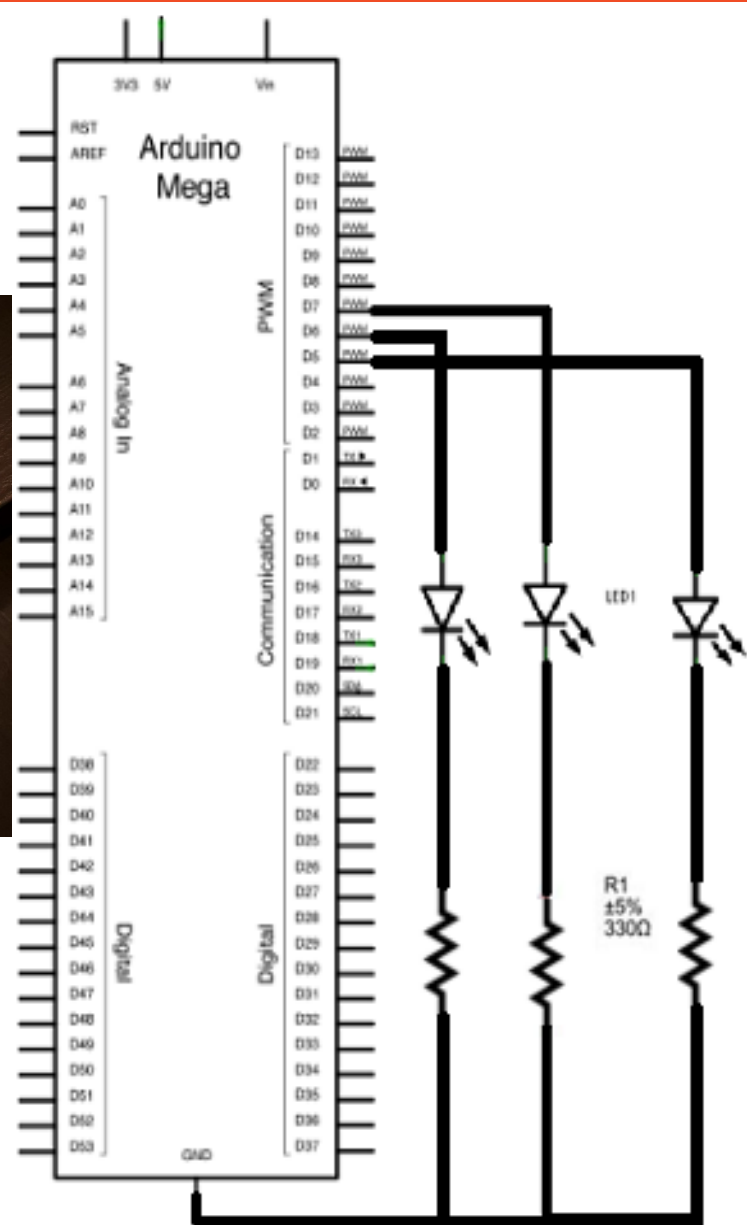
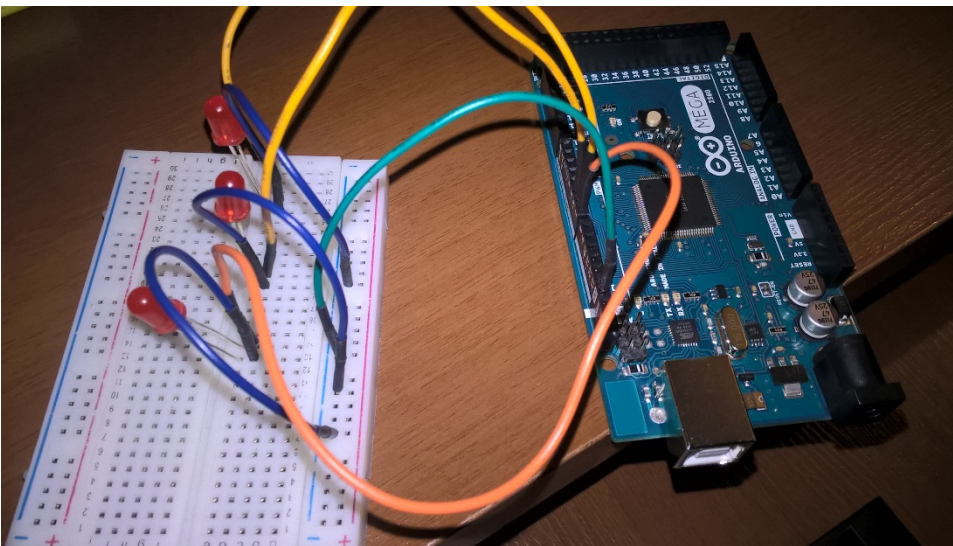
Knowledge Needed

Serial Monitor is a window, which is a part of the Arduino IDE software. Its job is to allow you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino.

The message “Enter LED Number 0 to 9 or 'x' to clear” has been sent by the Arduino, and it is telling us what commands we can send to the Arduino which is either to send the 'x' (to turn all the LEDs off) or the number of the LED you want to turn on (where 0 is the bottom LED, 1 is the next one up right up to 7 for the top LED).



Hardware Connection





Code Writing

```
void setup() {  
  pinMode(7,OUTPUT);  
  pinMode(6,OUTPUT);  
  pinMode(5,OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  if (Serial.available()) {  
    char c=Serial.read();  
    int n=c-'0';  
    if (n==1){  
      digitalWrite(7,HIGH);  
      digitalWrite(6,LOW);  
      digitalWrite(5,LOW);  
    }  
  }
```

```
else if(n==2){  
  digitalWrite(7,LOW);  
  digitalWrite(6,HIGH);  
  digitalWrite(5,LOW);  
}  
else if(n==3){  
  digitalWrite(7,LOW);  
  digitalWrite(6,LOW);  
  digitalWrite(5,HIGH);  
}  
else {  
  digitalWrite(7,LOW);  
  digitalWrite(6,LOW);  
  digitalWrite(5,LOW);  
}  
}
```




Components Needed

1. Arduino board
2. 1x LED(Red color)
3. 1x Button
4. 1x Buzzer
5. 1x Breadboard
6. Wires

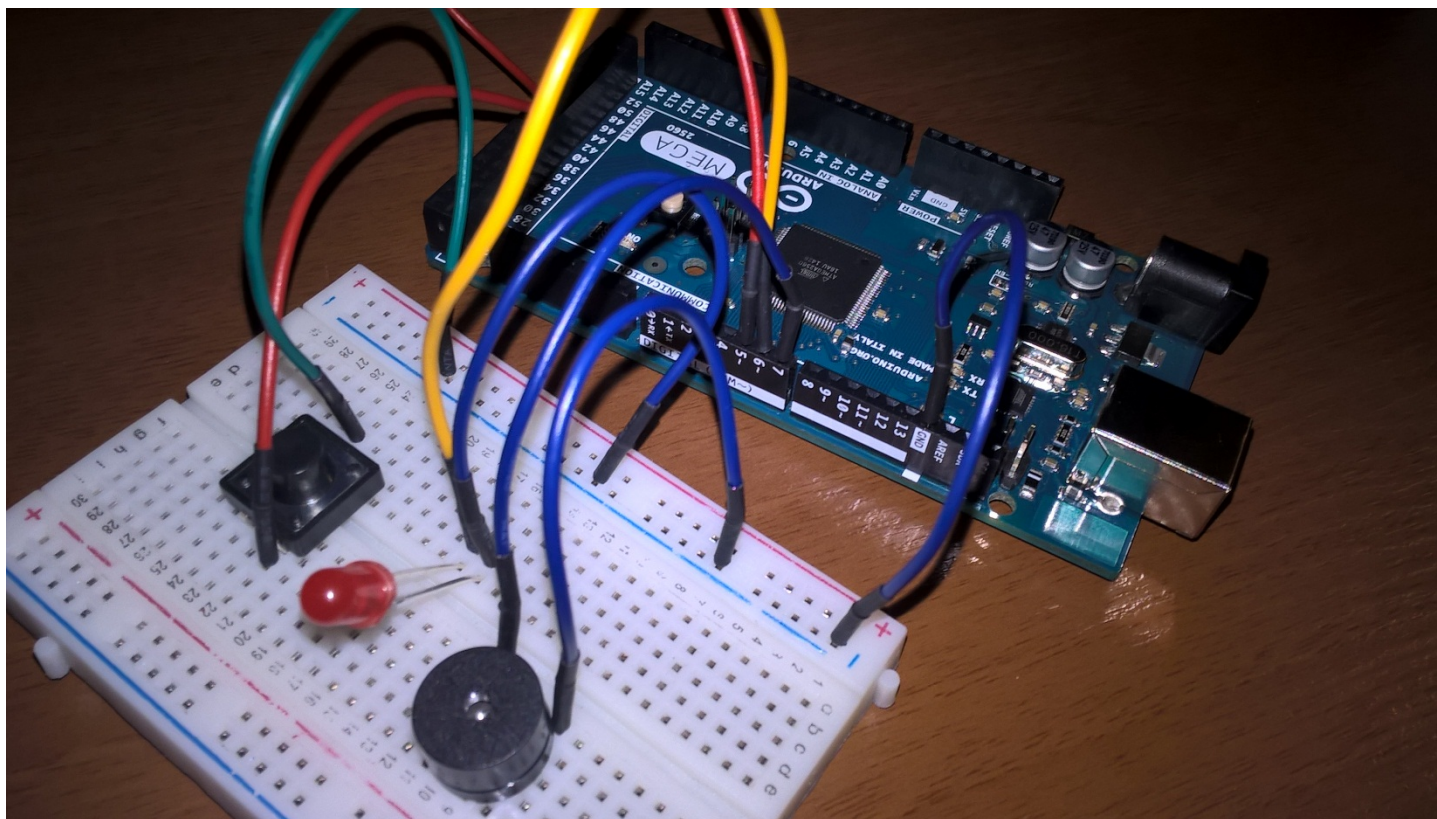


Knowledge Needed

We'll control the LEDs's and buzzer: the result of first push of button is LEDs' light and the result of second push of button is an action of buzzer.



Hardware Connection





Code Writing

```
void setup() {  
  pinMode(5,INPUT_PULLUP); //button  
  pinMode(6,OUTPUT); //LED  
  pinMode(7,OUTPUT); //buzzer  
  Serial.begin(9600);  
}  
int i=0;  
void loop() {  
  if (digitalRead(5)==0){  
    delay(1000);  
    i++; }  
  
  Serial.println(i);  
  if (i%2==1) {  
    digitalWrite(6, HIGH);  
    analogWrite(7, 0);  
  }  
  else {  
    digitalWrite(6, LOW);  
    analogWrite(7, 100);  
  }  
}
```



Components Needed

1. Arduino board
2. 1 x Potentiometer (1k or 10k)
3. 1x Breadboard
4. Wires



Knowledge Needed

A Potentiometer is a three-terminal resistor with a sliding contact that forms an adjustable voltage divider. As the following picture shows, when we connect the potentiometer as the picture shows, we'll get a voltage from the middle point. When we rotate the stick, the middle point will output a voltage between 0V-5V. We can use A0-A15 to read this voltage, then to control some external device based on this voltage message.

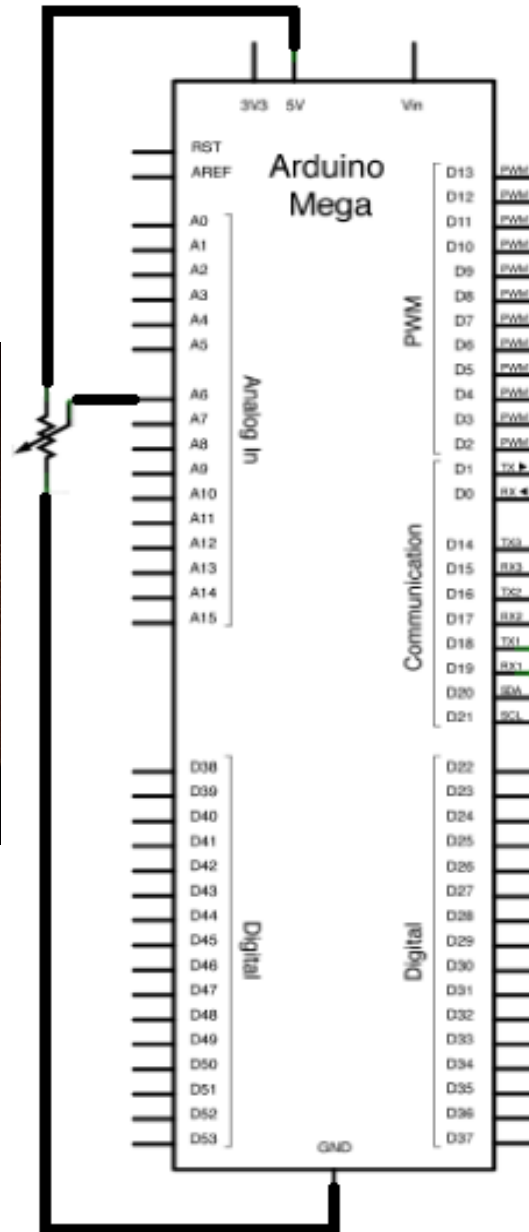
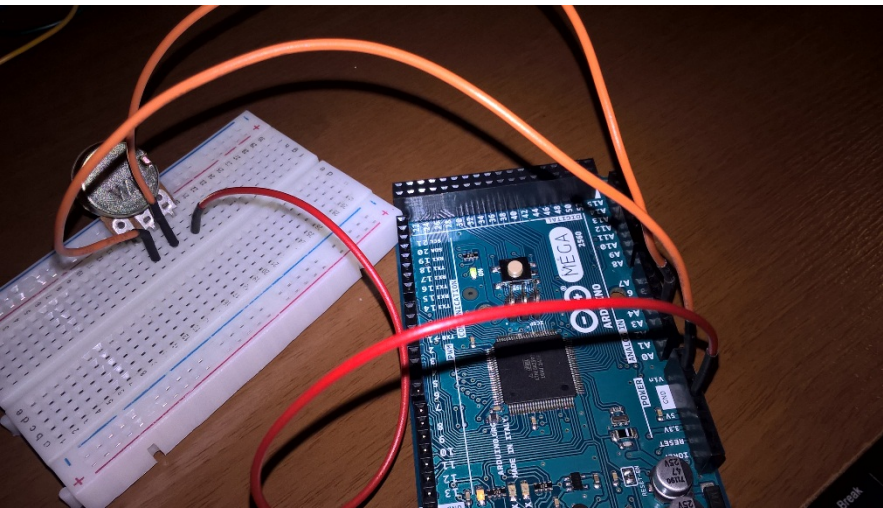




Tempus

DesIRE

Hardware Connection





Code Writing

```
void setup() {  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  Serial.println(analogRead(A2));  
  delay(1000);  
}
```




Components Needed

1. Arduino board
2. 2 x LED
3. 1 x Potentiometer (1k or 10k)
4. 1 x Breadboard
5. Wires

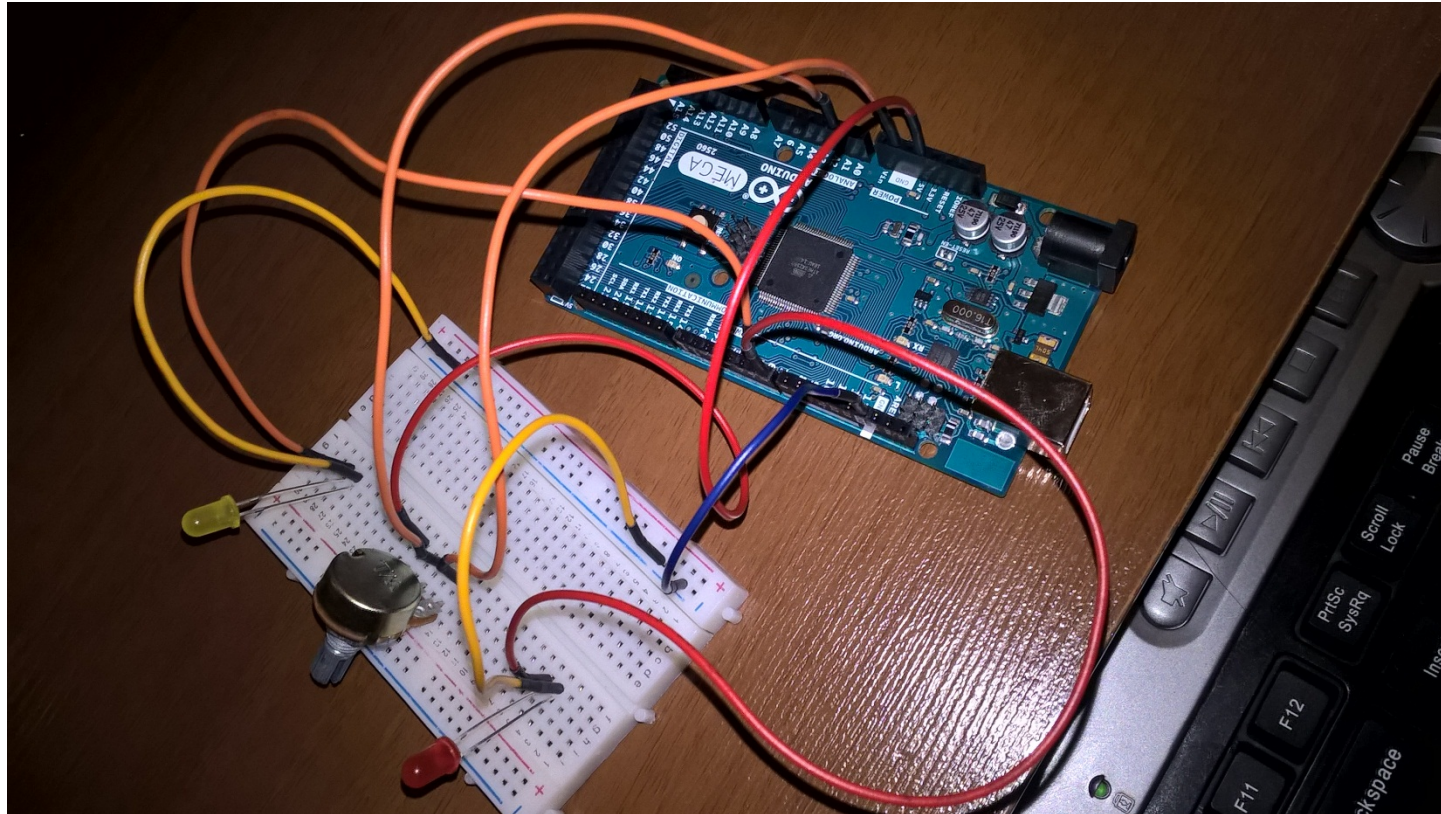


Knowledge Needed

We'll control the LEDs' light the following rule:
If we turn right the potentiometer, the right LED will be to turn on, another case will be to turn on the left LED.



Hardware Connection





Code Writing

```
void setup() {  
  pinMode(6,OUTPUT);//left LED  
  pinMode(7,OUTPUT);//right LED  
}
```

```
void loop() {  
  if (analogRead(A2)>=512){  
    digitalWrite(6, HIGH);  
    digitalWrite(7, LOW);  
  }  
  else {  
    digitalWrite(7, HIGH);  
    digitalWrite(6, LOW);  
  }  
}
```



Components Needed

1. Arduino board
2. 1 x Buzzer
3. 1 x Potentiometer (1k or 10k)
4. 1 x Breadboard
5. Wires



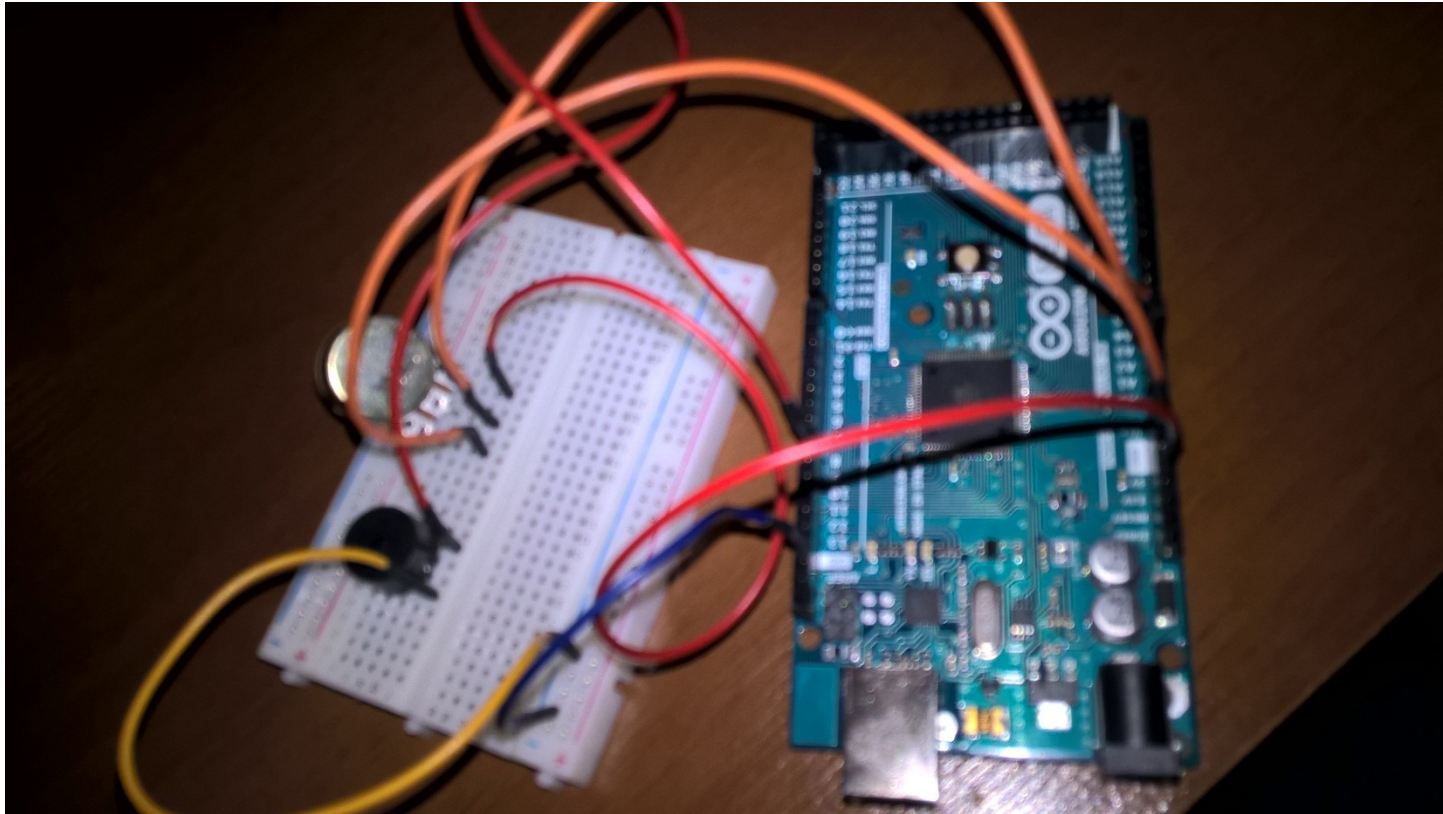
Knowledge Needed

We'll control the Buzzer volume the following rule:

If we turn right the potentiometer, the volume of the buzzer will be to turn up, another case will be to turn down.



Hardware Connection





Code Writing

```
void setup() {  
  pinMode(8,OUTPUT);  
  
}  
  
void loop() {  
  tone(8,analogRead(A2));  
}
```



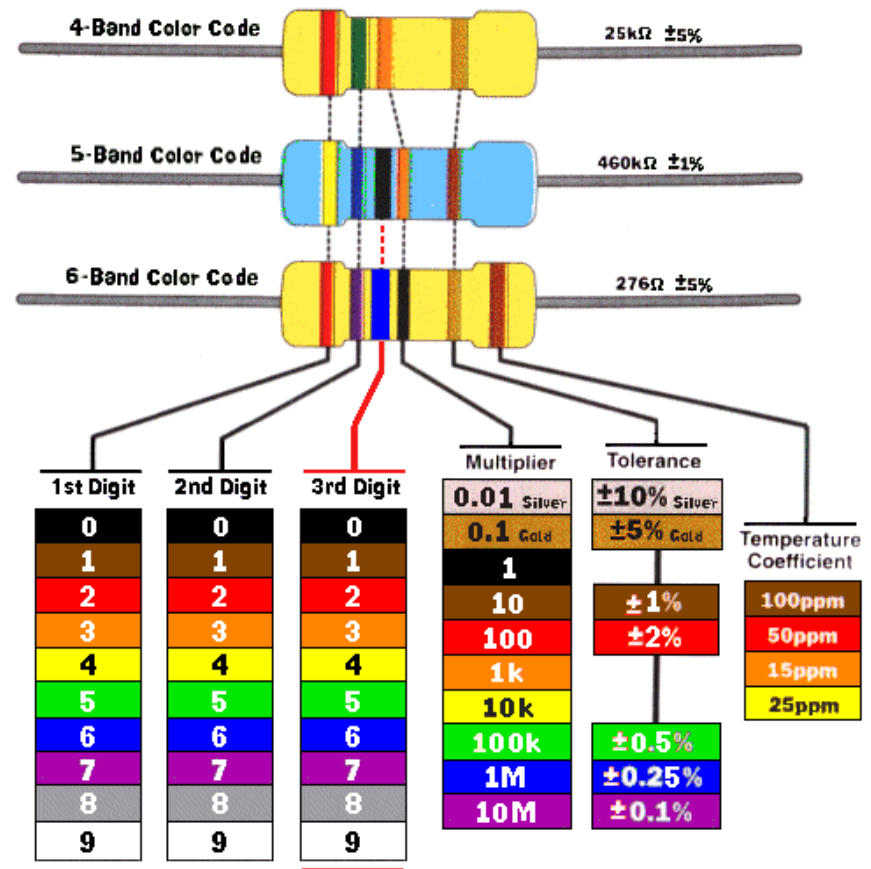

Components Needed

1. Arduino board
2. 1 x 5516LDR
3. 1 x 10k ohm resistor
4. 1 x LED(Red color)
5. 1 x Breadboard
6. Wires



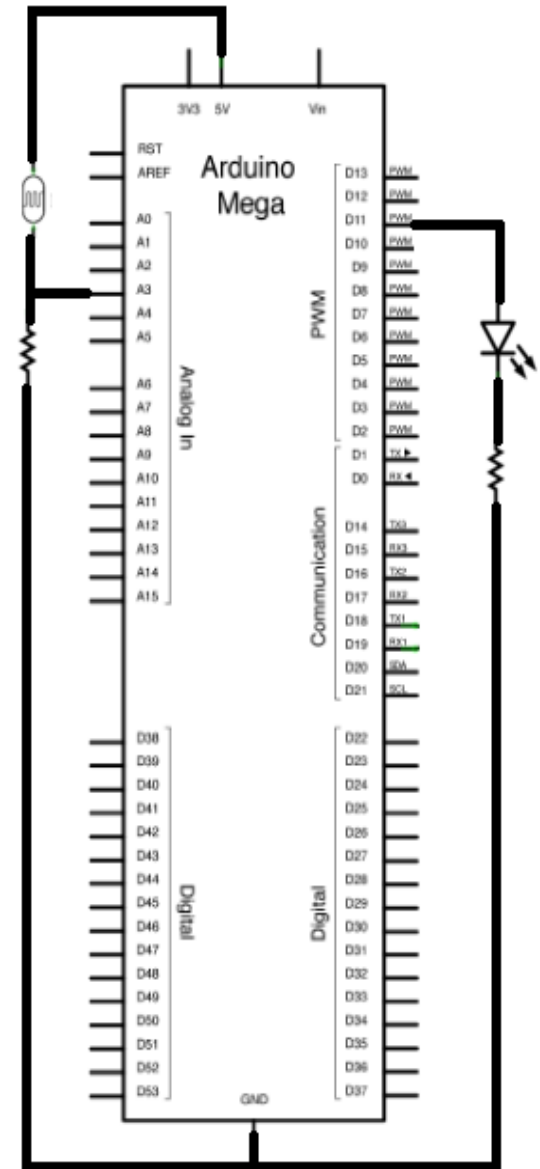
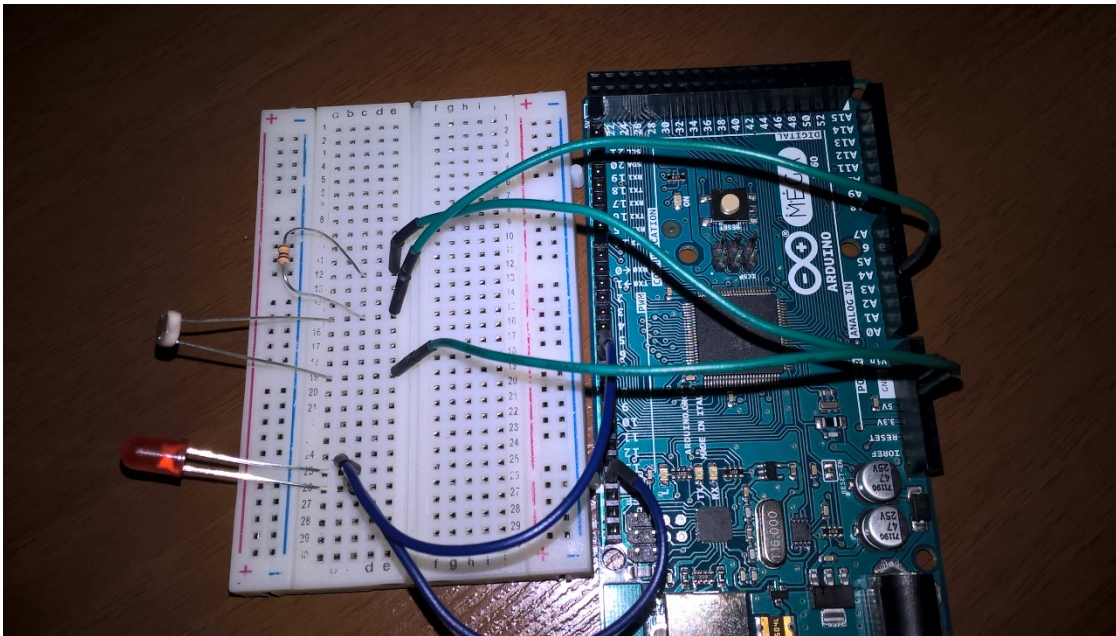
Knowledge Needed

LDR, also is Light Dependent Resistor. It's a resistor, but not a normal one. Its resistor value can be changed by the external light strength. The stronger the light strength of the outside environment, the smaller the resistor value of the LDR.





Hardware Connection





Code Writing

```
void setup() {  
  Serial.begin(9600);  
  pinMode(7, OUTPUT);  
  
}  
  
void loop() {  
  Serial.println(analogRead(A3));  
  if (analogRead(A3)>600)  
    digitalWrite(7,HIGH);  
  else  
    digitalWrite (7,LOW);  
  
}
```



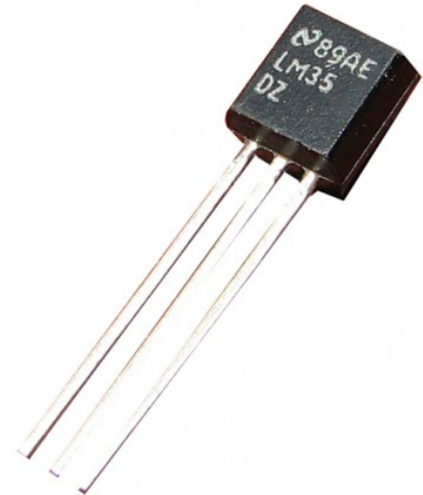
Components Needed

1. Arduino board
2. 1X LM35DZ
3. 1 x Breadboard
4. Wires



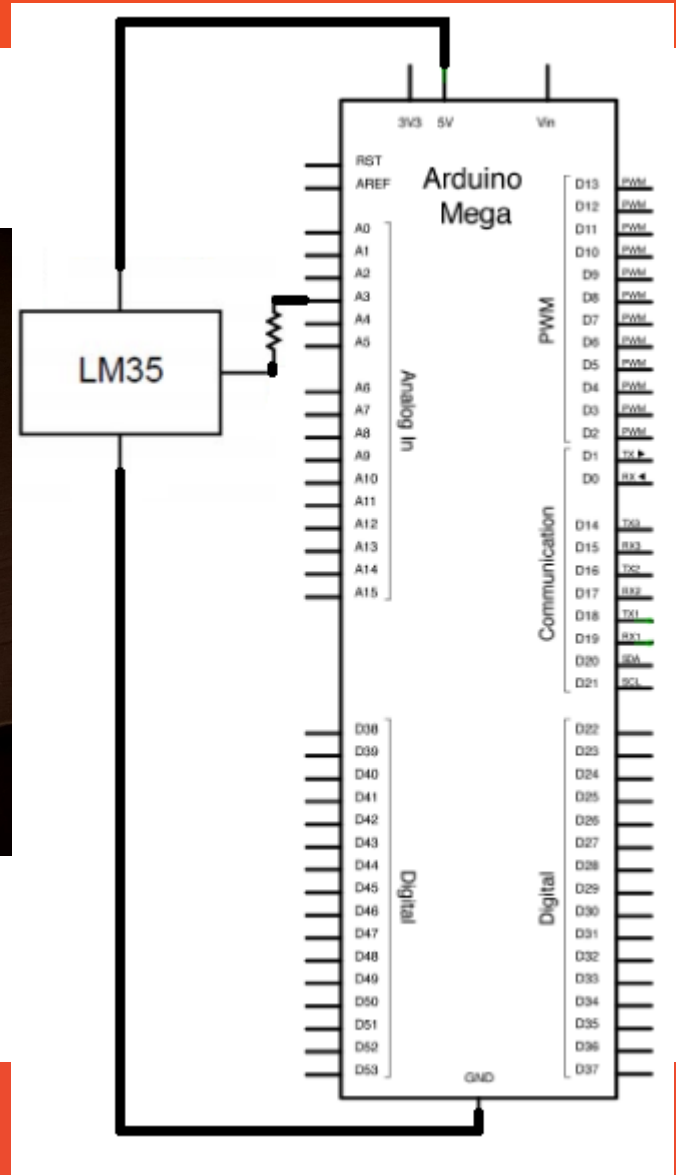
Knowledge Needed

LM35DZ is a sensor which can measure the room temperature. It is very easy to use. You can measure the temperature between 0°C to +100°C.





Hardware Connection





Code Writing

```
void setup() {  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  Serial.print("The room  
  temperature degrre is: ");  
  
  Serial.println(((analogRead(A3)/10  
  23)*5*100+32)*0.5);  
  
}
```




Components Needed

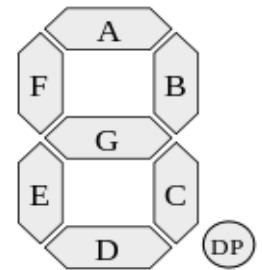
1. Arduino board
2. 1 x 1 Digital 8-Segment LED Display Tube
3. 1 x Breadboard
4. Wires



Knowledge Needed

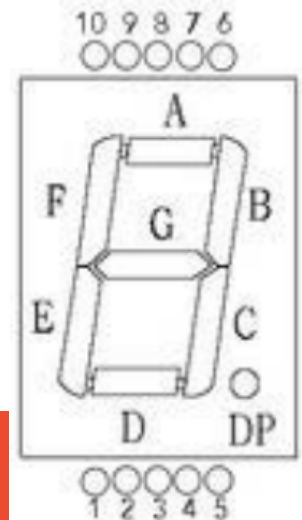
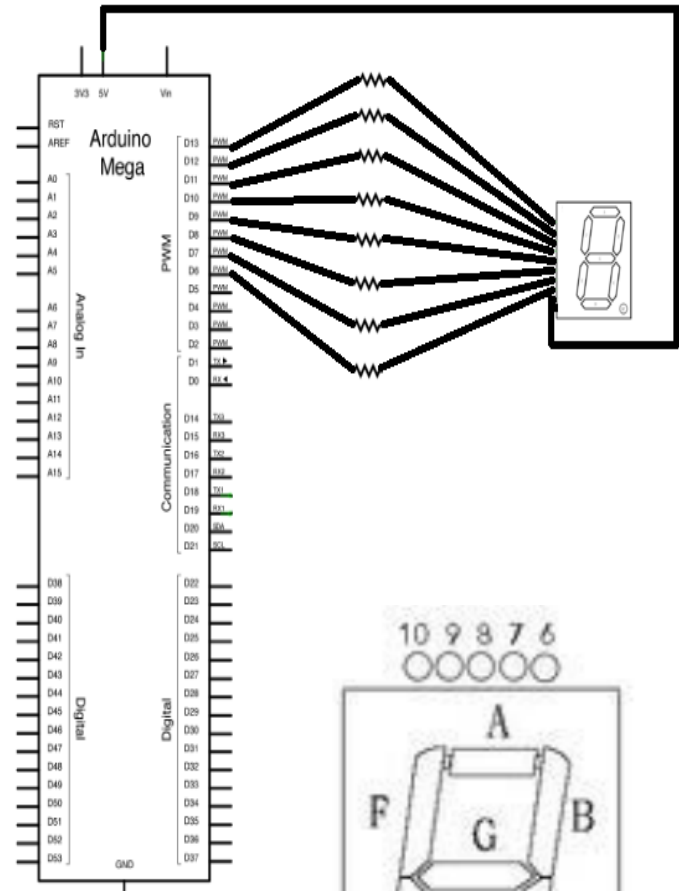
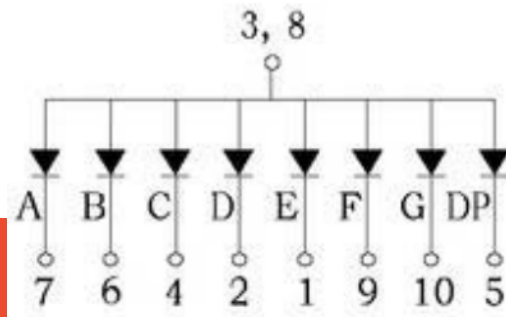
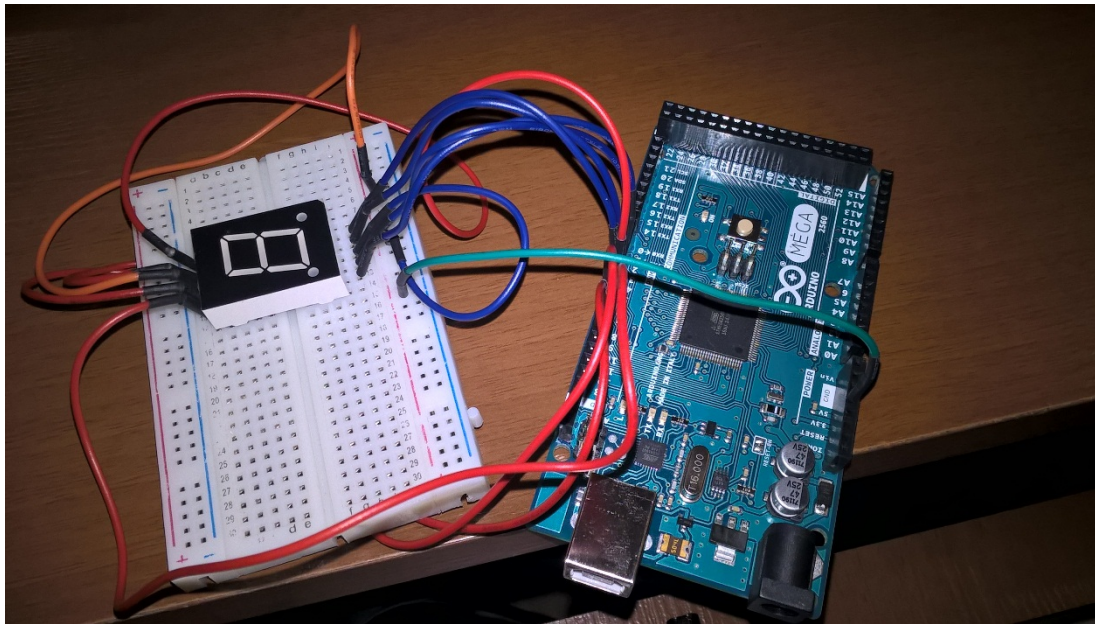
Eight-segment display. In fact it is composed of common LEDs. Usually there are two kinds, one is common-anode, the other is common-cathode.

We just need to light several led at the same time to make it show the special number.





Hardware Connection





Code Writing

```
void setup() {  
  for(int i=2;i<10;i++)  
  {  
    pinMode(i,OUTPUT);  
    digitalWrite(i,HIGH);  
  }  
}
```

```
void loop() {  
  digitalWrite(7,LOW);//A  
  delay(1000);  
  digitalWrite(6,LOW);//B  
  delay(1000);  
  digitalWrite(4,LOW);//C  
  delay(1000);  
  digitalWrite(3,LOW);//D  
  delay(1000);  
  digitalWrite(2,LOW);//E  
  delay(1000);  
  digitalWrite(8,LOW);//F  
  delay(1000);  
  digitalWrite(9,LOW);//G  
  delay(1000);  
  digitalWrite(5,LOW);//DP  
  delay(1000);  
}
```



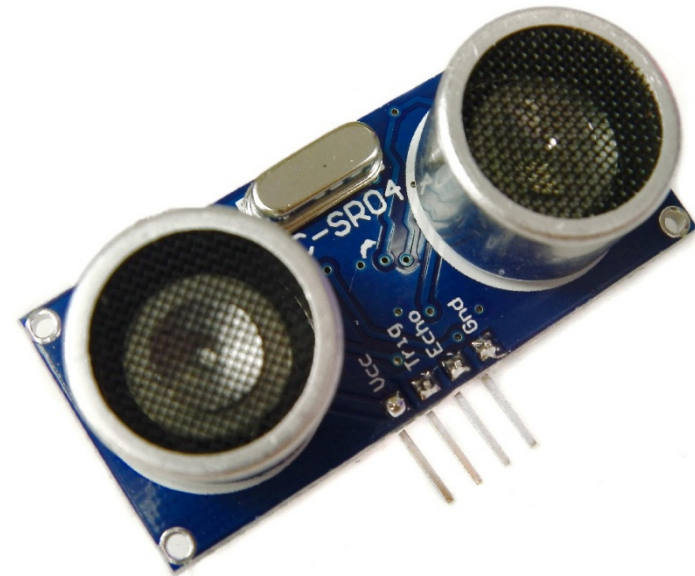
Components Needed

1. Arduino board
2. 1 x Ultrasonic Sensor
3. 1 x Breadboard
4. Wires



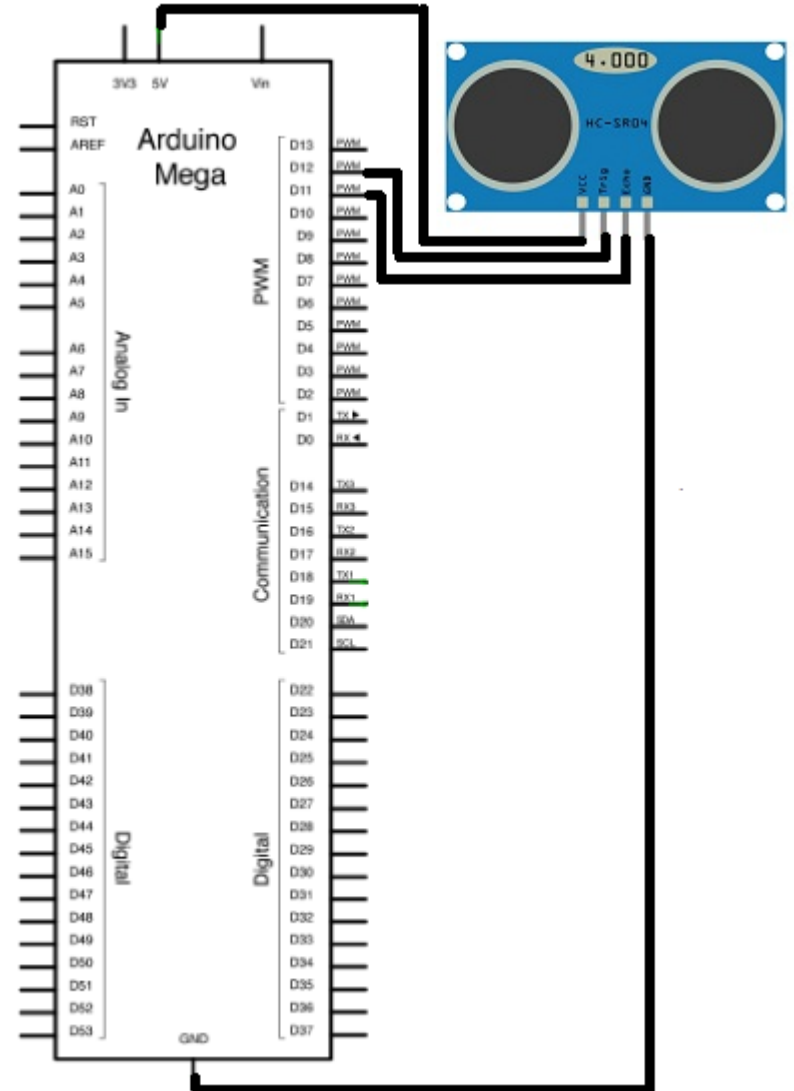
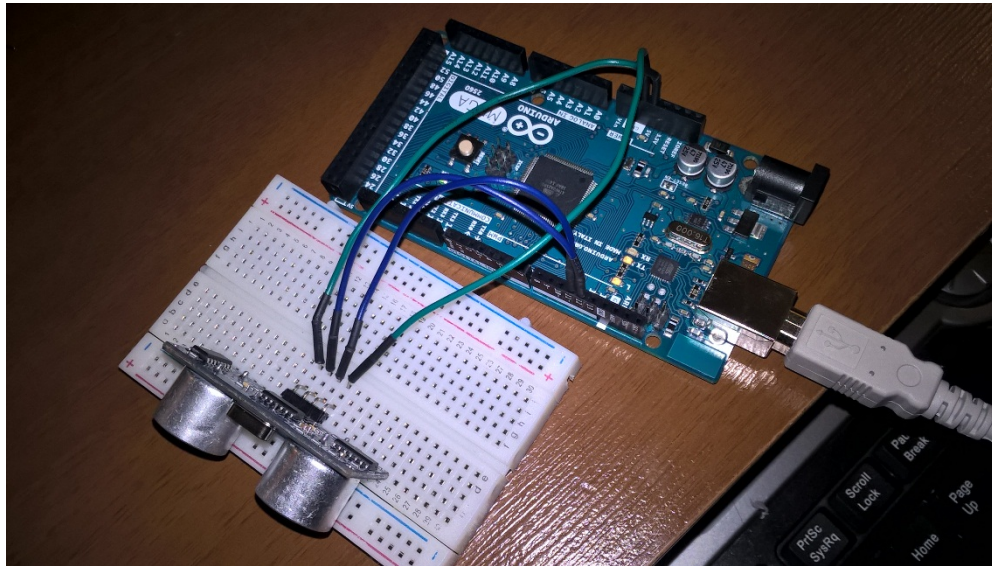
Knowledge Needed

Ultrasonic Distance Sensor is a popular and low cost solution for non-contact distance measurement function. It is able to measure distances from 2cm to 400cm with an accuracy of about 3mm. This module includes ultrasonic transmitter, ultrasonic receiver and its control circuit.





Hardware Connection





Code Writing

```
void setup() {  
  pinMode(12,OUTPUT); //triger  
  pinMode(11,INPUT); // echopin  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  digitalWrite(12,LOW);  
  delayMicroseconds(2000);  
  digitalWrite(12,HIGH);  
  delayMicroseconds(15);  
  digitalWrite(12,LOW);  
  delayMicroseconds(10);  
  Serial.print("Distance from the object is: ");  
  Serial.println(pulseIn(11,HIGH)/(2*29)); //cm  
  delay(500);  
}
```




Components Needed

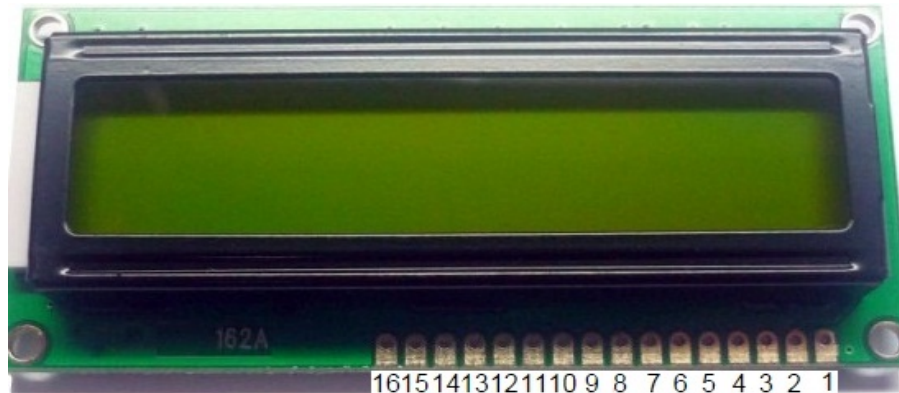
1. Arduino board
2. 1 x LCD (16x2)
3. 1 x Breadboard
4. Wires



Knowledge Needed

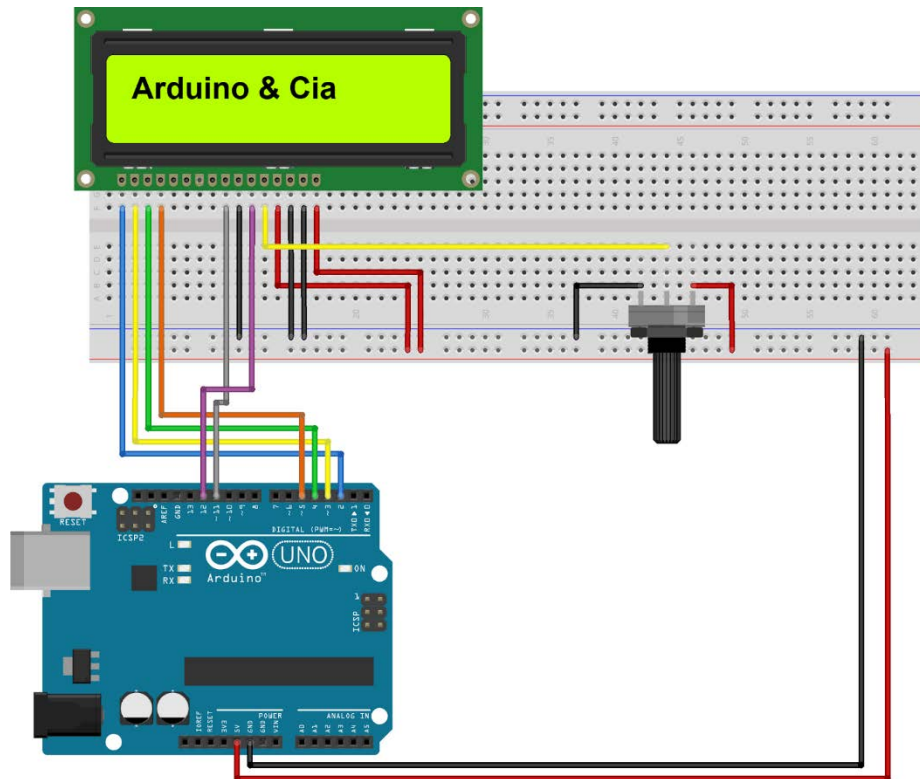
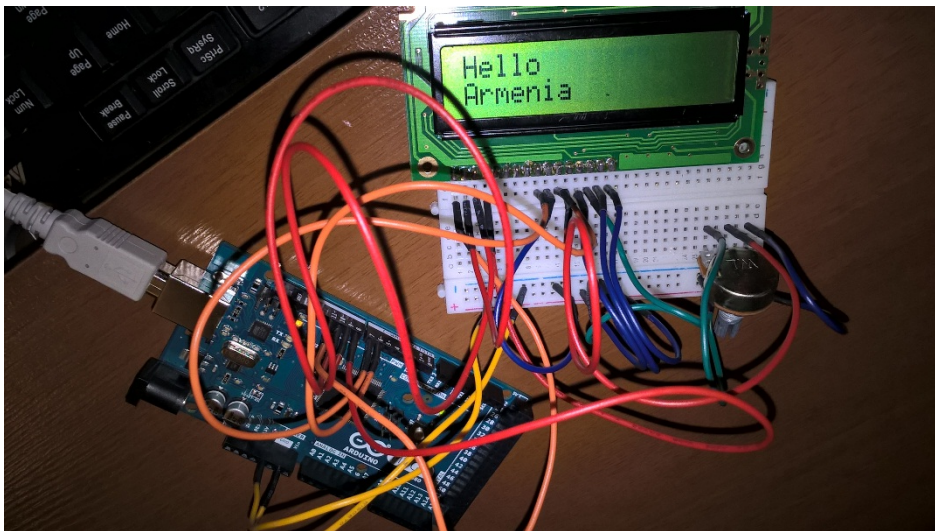
The LiquidCrystal library allows you to control LCD displays.

You will also learn how to use `lcd.begin()`, `lcd.print()` and `lcd.setCursor()` functions.





Hardware Connection





Code Writing

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(6,7,8,9,10,11);

void setup() {
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print ("Hello");
  lcd.setCursor(0,1);
  lcd.print ("Armenia");
}

void loop() {

}
```



Thank you for your attention!

