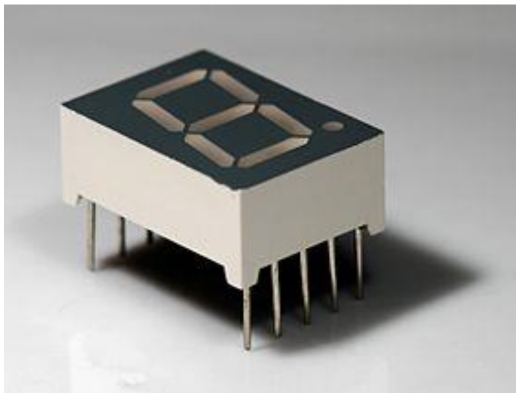
Digital Clock using Arduino and 7 Segment LED display

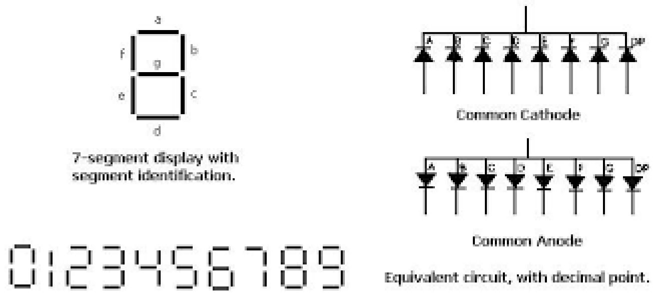
Introduction

7 segment display is a collection of 8 LED embedded in a single ceramic or plastic casing. It can be used for displaying numbers, digits or text. For this purpose, we have to lit each LED separately and make such a combination to display a meaningful text. There are LED driver ICs available in market for this purpose. One such IC is LM7447. It takes a BCD code for a number and give 7 outputs to drive this LED.

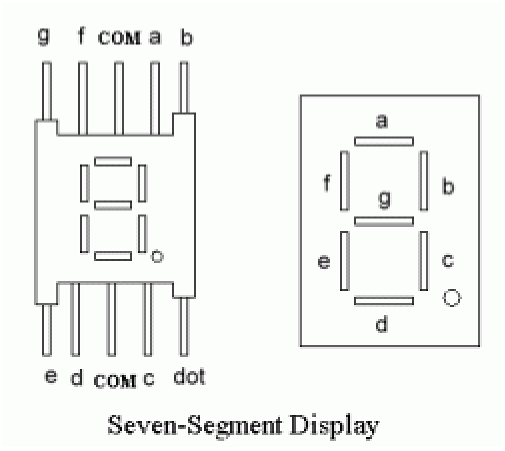


7 Segment LED internal Schematics

To drive this LED, we have to understand its internal configuration and how it can be used for our purposes. Following is the configuration for brighten an LED on it. 'a' through 'g' are used to display a number, while 'dot' pin is used to lit the point adjacent to digit. The central pin on both sides is called "Common Pin". It is used for common cathode or common anode configuration. What signal is given here depends on what type of LED segment it is? If it is common anode, a positive signal is given here. On the contrary, negative signal is provided if it is common cathode.


a
b
c
d
e
f
g
7-segment display with segment identification.
Common Cathode
Common Anode
Equivalent circuit, with decimal point.

Common Cathode and Common Anode Configuration

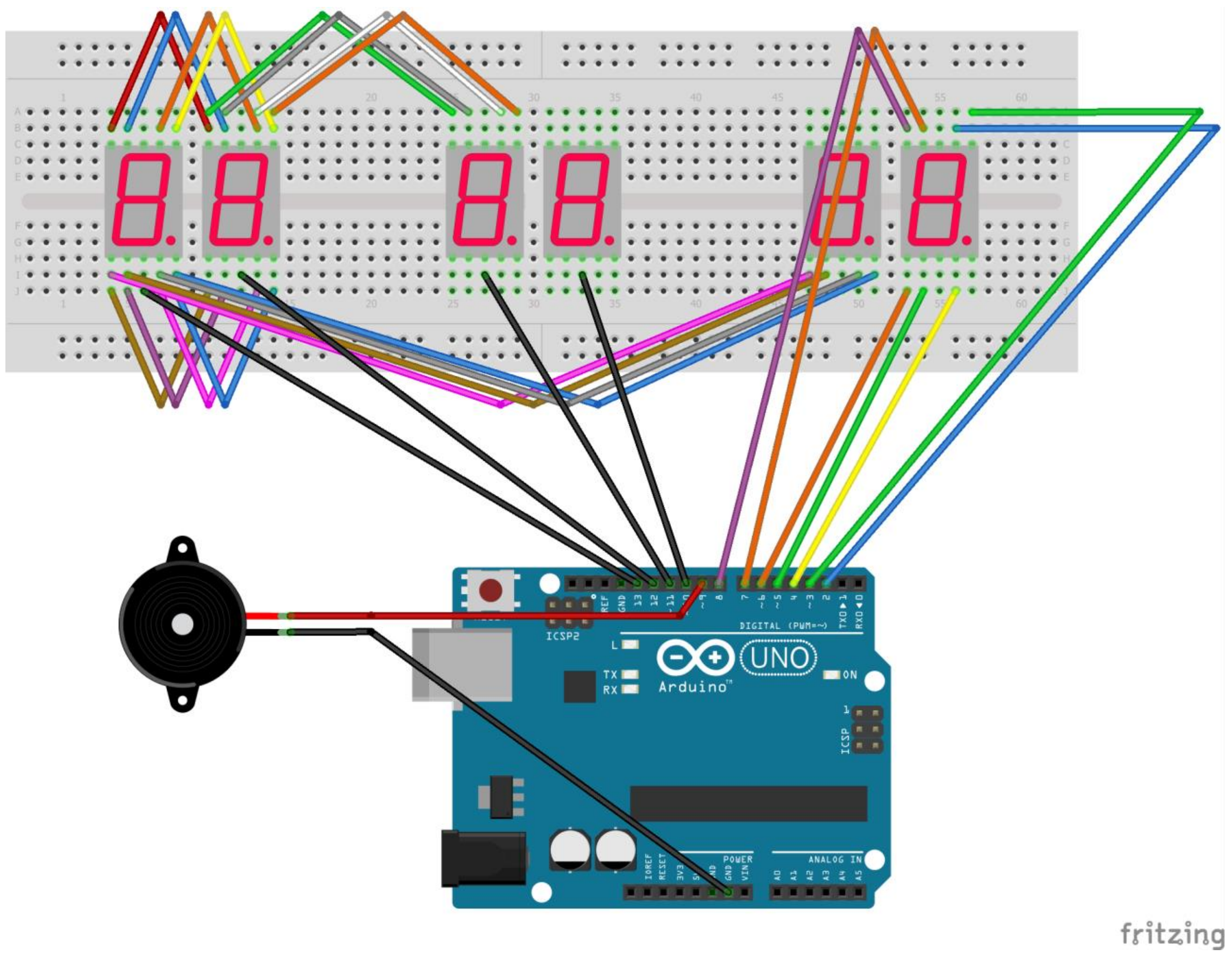


Components

Sr#	Components	Pieces
1	Arduino UNO	1
2	Male to Male Wires	Upto 40
3	Breadboard	1
4	7 Segment Display	4

Circuit Schematics

Following is the circuit given, it is not complete yet. To avoid complexity, only method for connection is shown in diagram. All the data pins are provided the same signal, only common pin is different as it is used as enable here. See code for details. Follow the given pin configuration to make connections, it is also same in given figure.



Pin configuration

Sr#	UNO pin#	Purpose
1	2	**A**, as given in schematics
2	3	**B**, as given in schematics
3	4	**C**, as given in schematics
4	5	**D**, as given in schematics
5	6	**E**, as given in schematics
6	7	**F**, as given in schematics
7	8	**G**, as given in schematics
8	9	Buzzer Pin
9	10	**En**able segment**# 1**
10	11	**En**able segment**# 2**
11	12	**En**able segment**# 3**
12	13	**En**able segment**# 4**

Task Description

Today, we will display data on it by use of Arduino. We will simply display a countdown timer this time only. It will start from 60:60 and decrement after one second. Next time, we will create algorithm to display time on it. Since Arduino have much computation power than an LED driver, we can take it to the next level by adding creative algorithm. All we have to do is to connect component in given schematics. After that you have uploaded the companion code, which is just counting down the timer. What changes you can make for this task? What will happen if we change value of variables in the loop? Beside this, what features we can add in it for improving, or what do you want to make out of it? A digital clock maybe, or an alarm clock. What changes it will require? Let's see the code for accomplishing your ideas:

```
const int a=2;
const int b=3;
const int c=4;
const int d=5;
const int e=6;
const int f=7;
const int g=8;

const int buzzer = 9;

const int m1 = 10;

const int m2 = 11;

const int s1 = 12;

const int s2 = 13;


int mm=60;

int ss=60;


unsigned long timer=0;


void setup()

{

  Serial.begin(9600);


  pinMode(a,OUTPUT);
  pinMode(b,OUTPUT);
  pinMode(c,OUTPUT);
  pinMode(d,OUTPUT);
  pinMode(e,OUTPUT);
  pinMode(f,OUTPUT);
  pinMode(g,OUTPUT);

  pinMode(buzzer,OUTPUT);

  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
}

void loop()
{
  for (int i=60;i>=0;i--)
  {
    for (int j=60;j>=0;j--)
    {
      displayTime(i,j);
    }
    Alarm();
  }
}

void displayTime(int mm, int ss)
{
  timer = millis();
  while((millis()- timer) <=1000)
  {
      displayNumber(mm/10,1);
      displayNumber(mm%10,2);
      displayNumber(ss/10,3);
      displayNumber(ss%10,4);
  }
}
void displayNumber(int number, int serial)
{
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,LOW);

  digitalWrite(m1,HIGH);
  digitalWrite(m2,HIGH);
  digitalWrite(s1,HIGH);
  digitalWrite(s2,HIGH);

  if (number==0)
  {

    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    //digitalWrite(g,HIGH);
  }

  if (number==1)
  {
//    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
//    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
//    digitalWrite(f,HIGH);
//    digitalWrite(g,HIGH);
  }

  if (number==2)
  {
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
//    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
//    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (number==3)
  {
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
//    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (number==4)
  {
//    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
//    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (number==5)
  {
    digitalWrite(a,HIGH);
//    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }
  if (number==6)
  {
    digitalWrite(a,HIGH);
//    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (number==7)
  {
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
//    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
//    digitalWrite(f,HIGH);
//    digitalWrite(g,HIGH);
  }

  if (number==8)
  {
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (number==9)
  {
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
//    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if (serial==1)
  {
    digitalWrite(m1,LOW);
  }
  if (serial==2)
  {
    digitalWrite(m2,LOW);
  }
  if (serial==3)
  {
    digitalWrite(s1,LOW);
  }
  if (serial==4)
  {
    digitalWrite(s2,LOW);
  }
```

```
}

void Alarm()
{
  for (int i=0;i<1000;i++)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=1000;i>0;i--)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=0;i<100;i++)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=100;i>0;i--)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=0;i<5000;i++)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=5000;i>0;i--)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=0;i<1500;i++)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=1500;i>0;i--)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=0;i<15000;i++)
    tone(buzzer,i+1000,(i*i)+1000);
  for (int i=15000;i>0;i--)
    tone(buzzer,i+1000,(i*i)+1000);
}
```