

Experiment No. 04

Seven Segment display interfacing with Microcontroller.

Introduction

The main aim of this lab is to interface seven segment display with microcontroller (Arduino) and observe their functionalities. The display counts from 0-9 and resets itself to zero.

Objectives

- Understand the functionality of Seven Segment Display
- Understand the code of Arduino and its working.

Outcomes

By the end of this experiment, student will have basic understanding of how to interface the seven segment display with microcontroller and will also understand the working principle of seven segment display.

Theory Overview

Seven segment displays are the output display devices that provide a way to display information in the form of image or text. For displaying the images or text in a proper manner, some types of displays can show only alphanumeric characters and digits. A seven segment display got its name from the very fact that it got seven illuminating segments. Each of these segments has a LED (Light Emitting Diode), hence the lighting. The LEDs are so fabricated that lighting of each LED is contained to its own segment as shown in figure 4.1

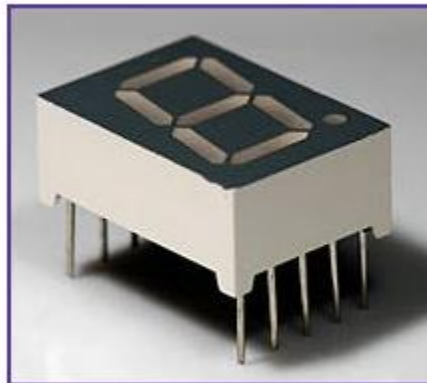
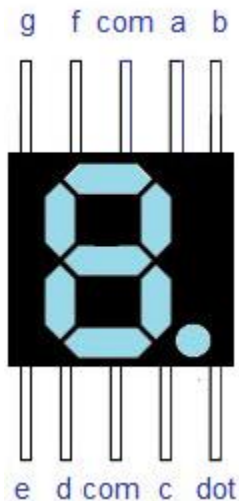


Figure 4.1: Seven Segment Display

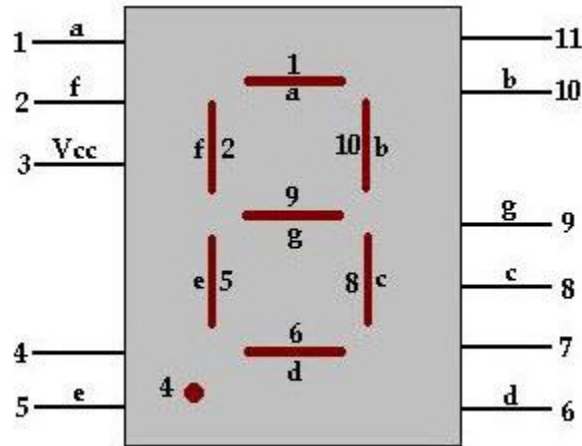


Figure 4.2: Pin diagram of Seven Segment Display

Types of 7-Segment Displays

There are two types of seven segment displays available in the market. According to the type of application, these displays can be used as shown in figure 4.3. The two configurations of seven segment displays are discussed below.

- Common Anode Display
- Common Cathode Display

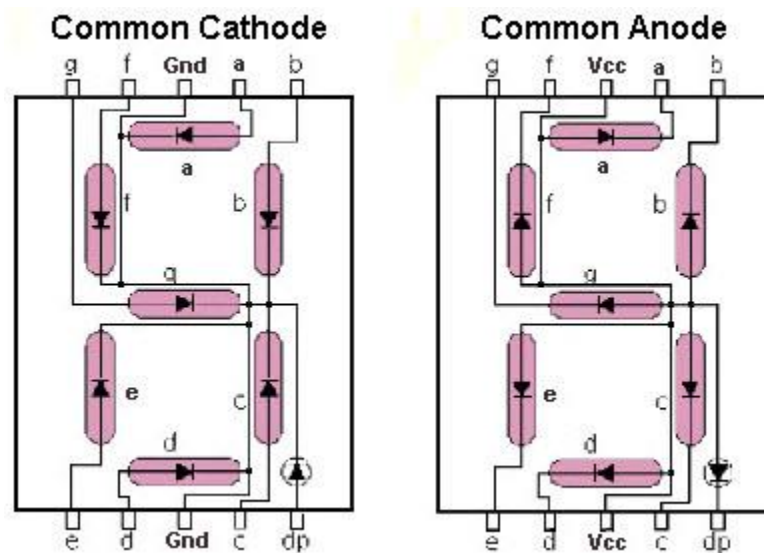


Figure 4.3: Types of Seven Segment Display

1) Common Anode Seven Segment Display

In common anode type, all the anodes of 8 LED's are connected to the common terminal and cathodes are left free as shown in figure 4.4. Thus, in order to glow the LED, these cathodes have to be connected to the logic '0' and anode to the logic '1'.

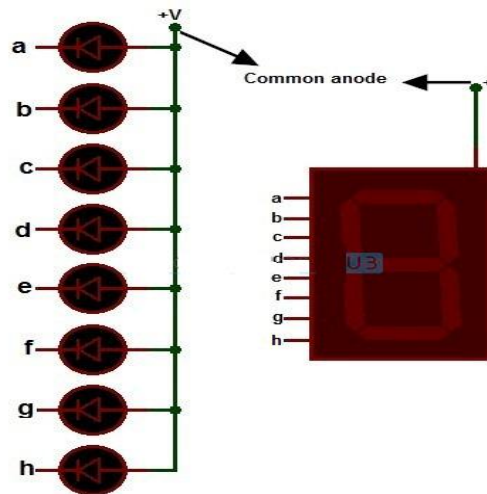


Figure 4.4: Common Anode Seven Segment Display

Below truth table gives the information required for driving the common anode seven segments.

Table 4.1: Truth Table of common anode seven segment display

Segments Inputs							7 Segment Display Output
a	b	c	d	e	f	g	
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2
0	0	0	0	1	1	0	3
1	0	0	1	1	0	0	4
0	1	0	0	1	0	0	5
0	1	0	0	0	0	0	6
0	0	0	1	1	1	1	7
0	0	0	0	0	0	0	8
0	0	0	0	1	1	0	9

In order to display zero on this segment one should enable logic high on a, b, c, d, e and f segments and logic low on segment 'g'. Thus, the above table provides data on seven segments for displaying numerals from 0-9.

2. Common Cathode Seven Segment Display

As the name indicates cathode is the common pin for this type of seven segments and remaining 8 pins are left free as shown in figure 4.5. Here, logic low is applied to the common pin and logic high to the remaining pins.

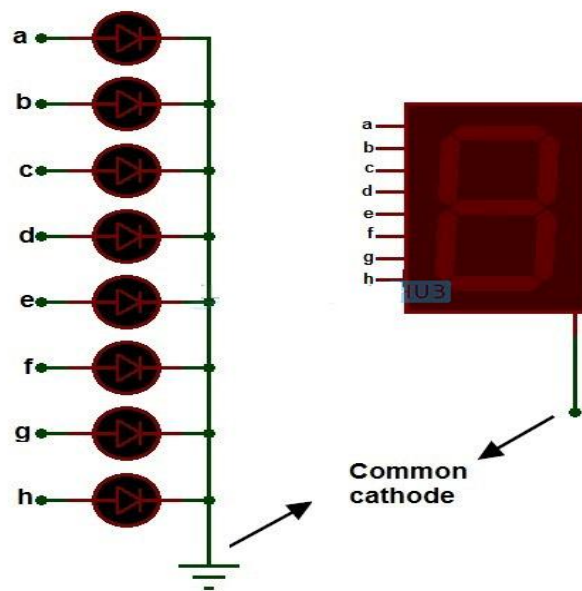


Figure 4.5: Common Cathode Seven Segment Display

Truth Table: The truth table of seven segment display is shown below.

Table 4.2: Truth Table of common cathode seven segment display

Segments Inputs							7 Segment Display Output
a	b	c	d	e	f	g	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	0	1	9

Above truth table shows the data to be applied to the seven segments to display the digits. In order to display digit '0' on seven segment, segments a, b, c, d, e and f are applied with logic high and segment g is applied with logic low.

To test the 7-segment use your Digital multi-meter and set it in the ohmmeter, connect the red probe to pin 3 or 8 (com pins) of the 7-segment and run the black probe on pins 1,2,4,5,6,7,9,10 (a,b,c,d,e,f,g,dp). When the meter is set to ohmmeter it acts as a voltage source as the positive terminal is the positive of the source and the negative terminal is the negative of the source.

LAB Exercise 1

Make a project to display a number 0 to 9 on seven segment display by using Arduino microcontroller also create a circuit diagram and develop a code.

Required Hardware

- Arduino board
- Adjustable DC Power Supply
- HDSP5503 seven segment display (common cathode)
- Resistors 220Ω
- Digital Multimeter
- Breadboard
- Probes
- Wires

Circuit Diagram

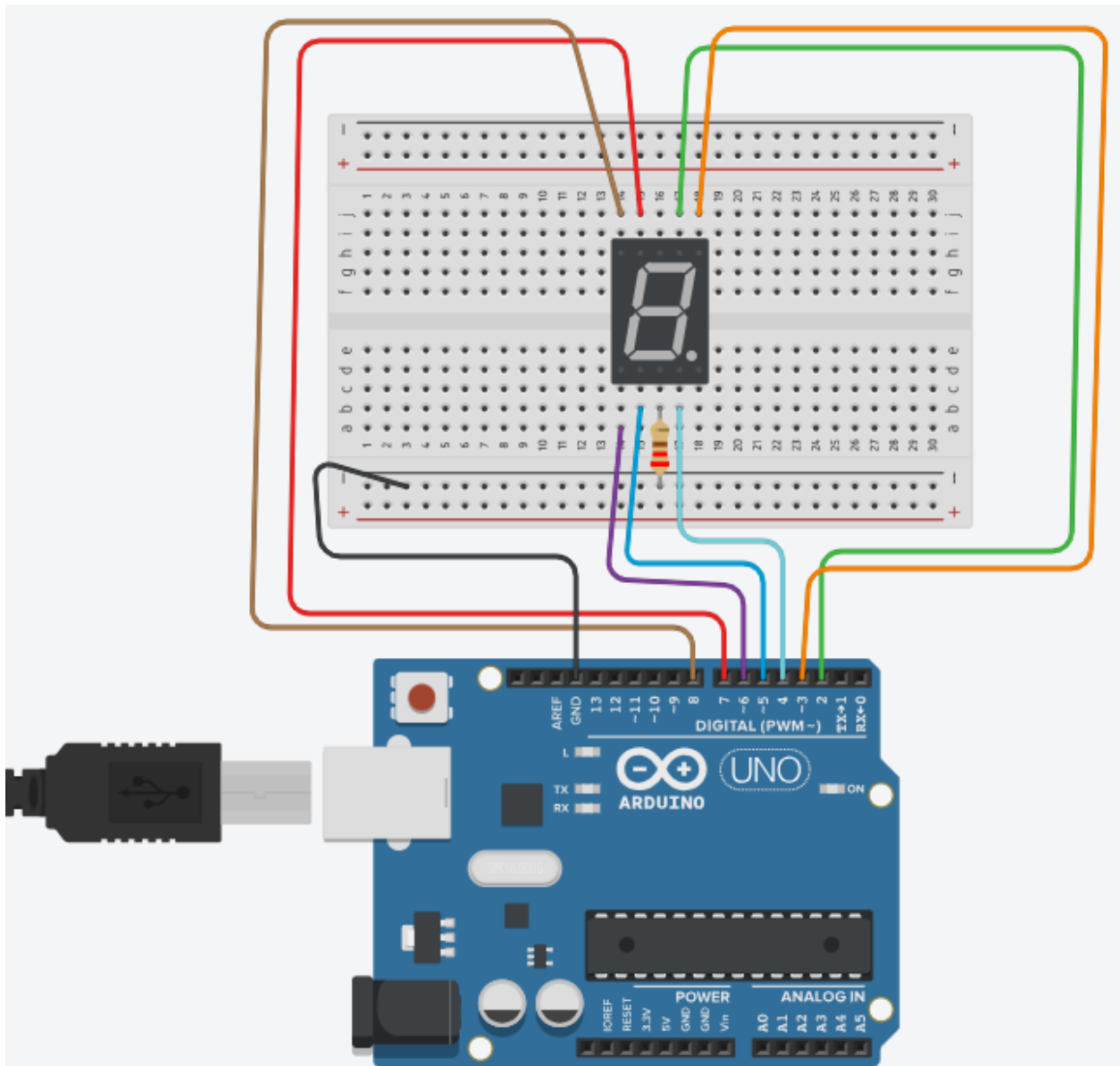


Figure 4.6 Circuit diagram of Seven Segment Display with Arduino

Procedure

Each segment of display can be checked by using multimeter in diode mode. Each segment should not be power with a voltage greater than 4v, if did the display will be damaged permanently. For avoiding this a common resistor can be provider at common terminal, as shown in circuit diagram. Now, if we want to display a “0” in this display

as shown in figure 4.7.1

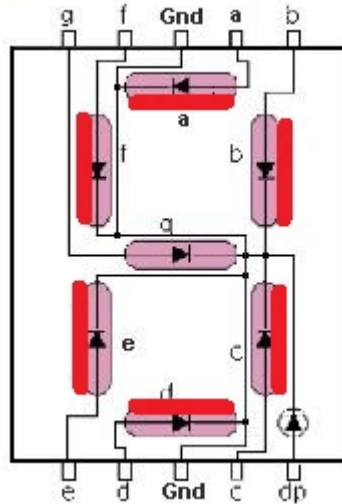


Figure 4.7.1 display of '0' on seven segment display

We need to turn the LEDs of segments “A, B, C, D, E F”, so we need to power PIN0, PIN1, PIN2, PIN3, PIN4 and PIN5. So every time we need a “0”, we need to power all the pins mentioned. Now, if we want “1” on display as shown in figure 4.7.2

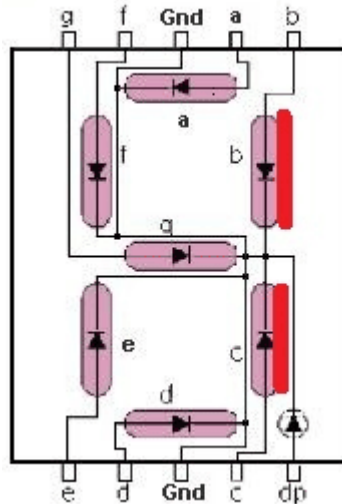


Figure 4.7.2 Display of '1' on seven segment display

We need to power segments “B, C”, for segment B, C to turn ON we need to power PIN1, PIN2. With both the pins high we get “1” on display. So as seen above we are going to power pins corresponding to the digit that to be shown on display.

There are sequence of steps to make this project are listed below

Step1: Prepare the circuit diagram according to figure 4.6

Step 2: Make a program to interface seven segment display with Arduino

Step 3: Connect the Arduino board to the computer using USB cable. Launch Arduino IDE and ensure correct USB port number and board name for establishing the connection.

Step 4: Once the program is uploaded and running on Arduino, open serial monitor

Step 5: You can verify the output of seven segment digits shows 0 to 9 and repeated the sequence again and again.

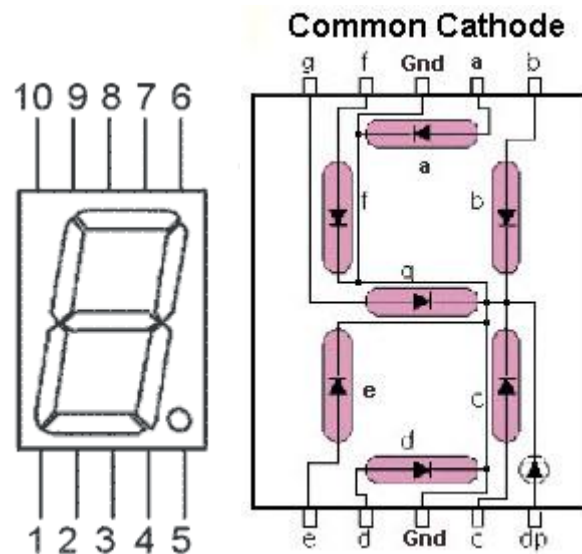


Figure 4.8: connection diagram

Arduino Pin	Seven Segment Pin
2	a 7
3	b 6
4	c 4
5	d 2
6	e 1
7	f 9
8	g 10

Code

```
#define segA 2//connecting segment A to PIN2
#define segB 3// connecting segment B to PIN3
#define segC 4// connecting segment C to PIN4
#define segD 5// connecting segment D to PIN5
#define segE 6// connecting segment E to PIN6
#define segF 7// connecting segment F to PIN7
#define segG 8// connecting segment G to PIN8
```

```

        int COUNT=0;//count integer for 0-9 increment
void setup()
{

    for (int i=2;i<9;i++)

    {

        pinMode(i, OUTPUT);// taking all pins from 2-8 as output

    }

}

void loop()

{

switch (COUNT)

    {

        case 0://when count value is zero show"0" on display
digitalWrite(segA, HIGH);
digitalWrite(segB,   HIGH);
digitalWrite(segC,   HIGH);
digitalWrite(segD,   HIGH);
digitalWrite(segE,   HIGH);
digitalWrite(segF,   HIGH);
digitalWrite(segG,   LOW);
break;

        case 1:// when count value is 1 show"1" on display
digitalWrite(segA, LOW);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);

```

```
digitalWrite(segF, LOW);  
digitalWrite(segG, LOW);  
break;
```

```
case 2:// when count value is 2 show"2" on disp
```

```
digitalWrite(segA, HIGH);  
digitalWrite(segB, HIGH);  
digitalWrite(segC, LOW);  
digitalWrite(segD, HIGH);  
digitalWrite(segE, HIGH);  
digitalWrite(segF, LOW);  
digitalWrite(segG, HIGH);  
break;
```

```
case 3:// when count value is 3 show"3" on disp
```

```
digitalWrite(segA, HIGH);  
digitalWrite(segB, HIGH);  
digitalWrite(segC, HIGH);  
digitalWrite(segD, HIGH);  
digitalWrite(segE, LOW);  
digitalWrite(segF, LOW);  
digitalWrite(segG, HIGH);  
break;
```

```
case 4:// when count value is 4 show"4" on disp
```

```
digitalWrite(segA, LOW);  
digitalWrite(segB, HIGH);  
  
digitalWrite(segC, HIGH);  
digitalWrite(segD, LOW);  
digitalWrite(segE, LOW);  
digitalWrite(segF, HIGH);  
digitalWrite(segG, HIGH);  
break;
```

```
case 5:// when count value is 5 show"5" on disp
```

```
digitalWrite(segA, HIGH);  
digitalWrite(segB, LOW);  
digitalWrite(segC, HIGH);  
digitalWrite(segD, HIGH);  
digitalWrite(segE, LOW);  
digitalWrite(segF, HIGH);  
digitalWrite(segG, HIGH);  
break;
```

```
case 6:// when count value is 6 show"6" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
```

```
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;
```

```
case 7:// when count value is 7 show"7" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, LOW);
break;
```

```
case 8:// when count value is 8 show"8" on display
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);
break;
```

```
case 9:// when count value is 9 show"9" on
disp digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;
```

```
break;
```

```
}
```

```
if (COUNT<10)

{

    COUNT++;

    delay(1000);///increment count integer for every second

}

if (COUNT==10)

{

    COUNT=0;/// if count integer value is equal to 10, reset it to zero.
    delay(1000);
}

}
```

LAB Exercise 2

Make a project to display the numbers from 00 to 99 on two Seven-Segment. Create a circuit diagram and develop code in Arduino IDE.

Required Hardware

- Arduino Uno - 1
- Programmer Cable - 1
- Seven-Segment Display - 2
- Bread Board - 1
- Male-Male Jumper Wires

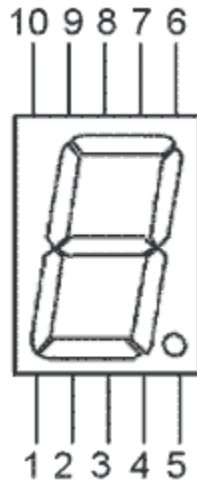


Figure 4.8: Pin diagram of common anode seven segment

Attach the pins 3 and 8 on both the SSD's to 3.3V.(Use resistors)

Since this is for Common Anode SSD, the LED glows when it is LOW/0.

If you are using a Common Cathode SSD, attach pins 3 and 8 to ground and then the LEDs glow when attached to HIGH/1.

Do **not** let 5 V pass through any of your LEDs, You must use resistors.

Table 4.5 connection table of Arduino uno with seven segment display

Arduino Pin	SSD 1		Arduino Pin	SSD 2
0	7		7	7
1	6		8	6
2	4		9	4
3	2		10	2
4	1		11	1
5	9		12	9
6	10		13	10

Project Circuit Diagram

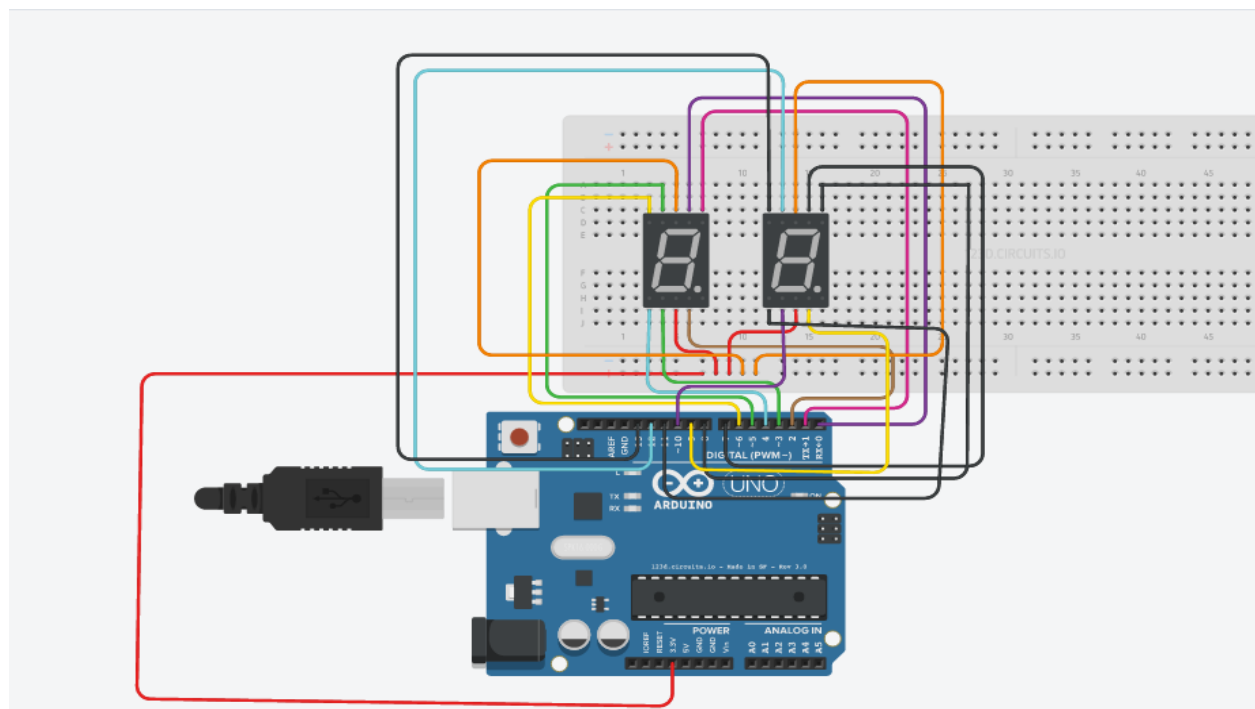


Figure 4.9: Circuit diagram of counter display 00 to 99

Code

//SSD is Seven-Segment Display

void setup()

{

 for (int

 i = 0; i <= 13; i++)

 pinMode(i, OUTPUT); *//Set all pins from 0 to 13 as OUTPUT*

}

//The line below is the array containing all the binary numbers for the digits on a SSD from 0 to 9

```
const int number[11] = {0b1000000, 0b1111001, 0b0100100,
0b0110000,
  0b0011001, 0b0010010, 0b0000010, 0b1111000, 0b0000000,
0b0010000};
```

void loop()

{

 for (int tens = 0; tens < 10; tens++)

 {

 display_tens(tens);

 }

}

void display_tens(const int tens)

{

 int pin1, a, ones;

//pin1 is just used to deal with pins of the 1st SSD which displays the tens digit

 for (pin1 = 0, a = 0; pin1 < 7; pin1++, a++)

 {

 digitalWrite(pin1, bitRead(number[tens], a));

 }

 for (ones = 0; ones < 10; ones++)

 {

 display_ones(ones);

 delay(300);

//I have given a delay of 300 milliseconds. You can put your own Time!!


```

    }
}

void display_ones(const int x)
{
    int pin2, b;

    //pin2 is just used to deal with pins of the 2nd SSD which displays the ones digit

    for (pin2 = 7, b = 0; pin2 <= 13; pin2++, b++)
    {
        digitalWrite(pin2, bitRead(number[x], b));

    }
}

```

bitRead()

Examples

```

// sets x to 33 (00100001 in binary)
byte x = 33;
// Sets a = 1, since bitRead returns the value of the
5th bit of variable x
byte a = bitRead(x, 5);

```

Description The **bitRead** command returns the bit value of the bit index requested

Syntax var = **bitRead**(byteValue, bitIndex)

Parameters byteValue byte: the byte from which read the bit

bitIndex int: the bit to read

Returns the bit value 0 or 1

Precautions

Before you start to mess with a computer there are basic guidelines to be aware of, such as turn the power off and watch out for static electricity. The three kind of activity that requires touching the Arduino

- Setting up wire, component layout or breadboard.
- Debugging a running setup.
- Moving, mounting up the board somewhere.

Review Questions

Question 1

- Solve the LAB Exercise 1
- Develop the code
- Sketch the circuit diagram
- Attached the hardware running picture in color

Question 2

- Solve the LAB Exercise 2
- Develop the code
- Sketch the circuit diagram
- Attached the hardware running picture in color

LAB Assessment

Student Name		LAB Rubrics	CLO3 , P5, PLO3
		Total Marks	10
Registration No		Obtained Marks	
		Teacher Name	Syed M Hamedoon
Date		Signature	