Lab Manual for Embedded System Design

Lab No. 2

Arduino Interfacing with Relay & Seven Segment Display

Objectives

In this lab students are introduced to the interfacing of Arduino with Relay & seven segment display

LAB # 2

Arduino Interfacing with Relay & Seven Segment Display

Introduction

RELAY:

Relay is an electrically operated switch as shown in Fig. 2.1. 12V relay is used to isolate electrical load using Microcontrollers. Relay have two configurations. i.e. NO (Normally Open) and NC (Normally Close). Relay have coil which is energized by 12V that results a Switching Action based on NO-NC Configuration. If Relay is NO configuration then after the coil is Energized, Switching Action makes it in NC State that connects the Load. The internal circuitry of Relay is as shown in Fig. 2.2.

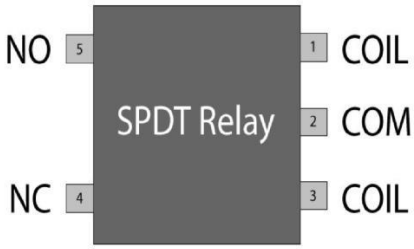

Fig. 2.1: Relay.

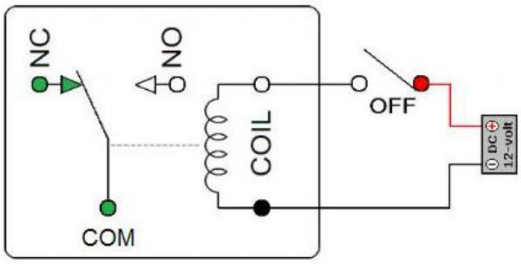

Fig. 2.2: Relay Internal Circuitry.

SEVEN SEGMENT DISPLAY:

A seven segment LED display is a special arrangement of 7 LED elements to form a rectangular shape using two vertical segments on each side with one horizontal segment on the top, middle, and bottom as shown in Fig. 2.3.

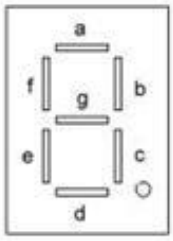

Fig 2.3: Seven segment display

By individually turning the segments on or off, numbers from 0 to 9 and some letters can be displayed. Seven segment displays sometime also have an eighth segment to display the decimal point. Therefore, a seven-segment display will require seven outputs to display a number, and one more output if the decimal point is to be displayed too. The segments are marked with non-capital letters: *a, b, c, d, e, f, g* and *dot,* where *dot* is the decimal point. The 8 LEDs inside the display can be arranged with a common cathode or common anode configuration. With a common cathode display, the cathodes of the entire segment LEDs are tied together and this common point must be connected to the ground. A required LED segment is then turned on by applying logic 1 to its anode. In common anode displays, all the anodes are tied together and the common anode is connected to the supply voltage Vcc. Individual segments are turned on by applying logic to their cathodes. The pin configuration is shown as Fig. 2.4.

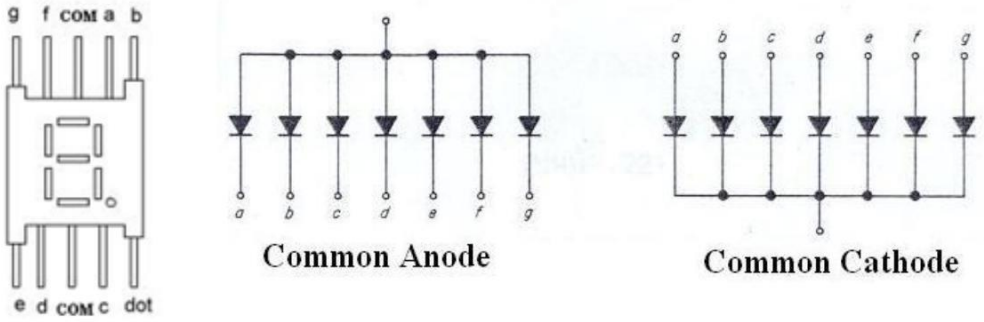

Fig. 2.4: Seven Segment Display (Types & Pin Configuration).

Time Boxing

Activity Name	Activity Time	Total Time
Login Systems + Setting up Proteus and Arduino Environment	3 mints + 5 mints	8 mints
Walk through Theory & Tasks	60 mints	60 mints
Implement Tasks	80 mints	80 mints
Evaluation Time	30 mints	30 mints
	Total Duration	178 mints

Objectives

In this lab students are introduced to open source single board microcontroller-Arduino, and its various design applications in designing embedded systems. Digital hardware components like Relay and Seven Segment Display will be interfaced with it.

Lab Tasks/Practical Work

1. **Write a sketch to interface Arduino with the Relay. The Relay should be controlled by a SPDT Switch.**

When the "Switch" is ON, "Relay" will be in "ON" state and Lamp will glow, otherwise Lamp will not glow.

Connect and *Switch* with Digital Pin 13 and a Relay through Transistor to Digital Pin 3 of the Arduino.

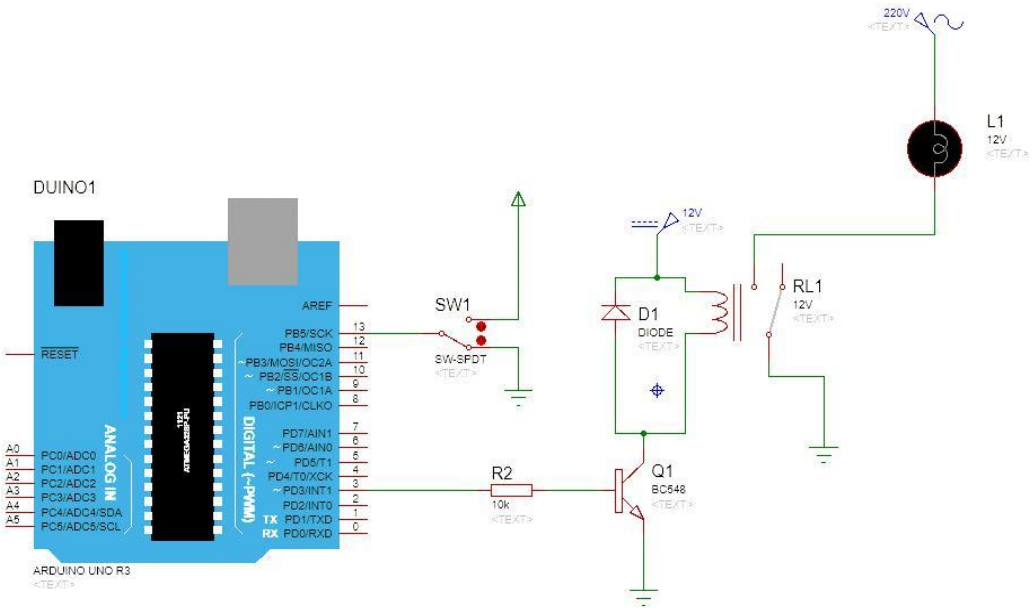

Fig. 2.5: Arduino – Relay Connection

Code

```
int buttonPin = 13;
int relayPin = 3;

// Variables will change. int buttonState = 0;

void setup( )
{
        pinMode(relayPin, OUTPUT); pinMode(buttonPin, INPUT);
}

void loop( )
{
        buttonState = digitalRead(buttonPin);

        // Check if the Push Button is Pressed. if (buttonState == HIGH)
        {
                digitalWrite(relayPin, HIGH);
         }

        else
        {
                digitalWrite(relayPin, LOW);
        }
}
```

2. **Write a sketch to interface Arduino with Seven Segment Display. It should work as a decade counter. The Start / Stop of counting should be controlled through a SPDT Switch.**

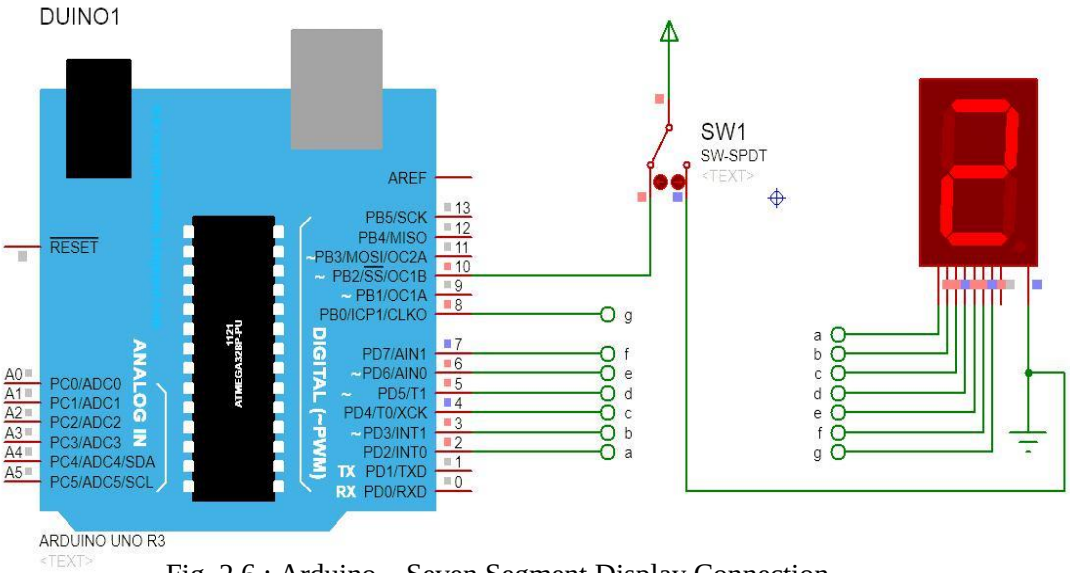

Fig. 2.6 : Arduino – Seven Segment Display Connection

Code:

```
#define segA 2          // Connecting Segment A to PIN 2.
#define segB 3          // Connecting Segment B to PIN 3.
#define segC 4          // Connecting Segment C to PIN 4.
#define segD 5          // Connecting Segment D to PIN 5.
#define segE 6          // Connecting Segment E to PIN 6.
#define segF 7          // Connecting Segment F to PIN 7.
#define segG 8          // Connecting Segment G to PIN 8.
#define button          // Connecting SPDT Switch to PIN 10.
10


int COUNT=0;            // Count Integer for 0-9 Increment.
int ButtonState;        // Variable Checking the State of the Button.


void setup()
{
        for (int i=2;i<9;i++)
        {
          pinMode(i,OUTPUT);          // Taking all Pins from 2-8 as OUTPUT.
```

```
        }
        pinMode (10, INPUT);
}
void loop()
{
   ButtonState=digitalRead(button);
   if(ButtonState==HIGH)
   {
       switch (COUNT)
       {
         case 0:
               digitalWrite(segA, HIGH);
               digitalWrite(segB, HIGH);
               digitalWrite(segC, HIGH);
               digitalWrite(segD, HIGH);
               digitalWrite(segE, HIGH);
               digitalWrite(segF, HIGH);
               digitalWrite(segG, LOW);

         case 0:
               digitalWrite(segA, HIGH);
               digitalWrite(segB, HIGH);
               digitalWrite(segC, HIGH);
               digitalWrite(segD, HIGH);
               digitalWrite(segE, HIGH);
               digitalWrite(segF, HIGH);
               digitalWrite(segG, LOW);
         break;

         case 1:
               digitalWrite(segA, LOW);
               digitalWrite(segB, HIGH);
               digitalWrite(segC, HIGH);
               digitalWrite(segD, LOW);
               digitalWrite(segE, LOW);
               digitalWrite(segF, LOW);
               digitalWrite(segG, LOW);
         break;

          case 2:
               digitalWrite(segA, HIGH);
               digitalWrite(segB, HIGH);
               digitalWrite(segC, LOW);
               digitalWrite(segD,HIGH);
               digitalWrite(segE, HIGH);
               digitalWrite(segF, LOW);
               digitalWrite(segG, HIGH);
         break;
```

```
case 3:
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
        digitalWrite(segF, LOW);
        digitalWrite(segG, HIGH);
break;

case 4:
        digitalWrite(segA, LOW);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, LOW);
        digitalWrite(segE, LOW);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
break;

case 5:
        digitalWrite(segA, HIGH);
        digitalWrite(segB, LOW);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
break;

case 6:
        digitalWrite(segA, HIGH);
        digitalWrite(segB, LOW);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, HIGH);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
break;

case 7:

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

```
    case 8:
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, HIGH);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
    break;

    case 9:
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
    break;

}

if (COUNT<10)
{
        COUNT++;
        delay(1000);
        }

if (COUNT==10)
{
        COUNT=0;
        delay(1000);

        }
         }
```