NIELIT, Gorakhpur

Course Name: A-level (1st Sem.) **Subject:IoT**

Topic: 7-segment LED Display cont. **Date: 08.04.2020**

PRINTING NUMBERS TO THE DISPLAY

```
 #include "SevSeg.h"
SevSeg sevseg;

void setup(){
    byte numDigits = 1;
    byte digitPins[] = {};
    byte segmentPins[] = {6, 5, 2, 3, 4, 7, 8, 9};
    bool resistorsOnSegments = true;

    byte hardwareConfig = COMMON_CATHODE;
    sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments);
    sevseg.setBrightness(90);
}

void loop(){
        sevseg.setNumber(4);
        sevseg.refreshDisplay();
}
```

hardwareConfig = COMMON_CATHODE; This sets the type of display.Suppose I'm using a common cathode, but if you're using a common anode then use COMMON_ANODE instead.

byte numDigits = 1; This sets the number of digits on your display. I'm using a single digit display, so I set it to 1. If you're using a 4 digit display, set this to 4.

byte digitPins[] = {}; Creates an array that defines the ground pins when using a 4 digit or multi-digit display. Leave it empty if you have a single digit display. For example, if you have a 4 digit display and want to use Arduino pins 10, 11, 12, and 13 as the digit ground pins, you would use this: byte digitPins[] = {10, 11, 12, 13};. See the 4 digit display example below for more info.

byte segmentPins[] = {6, 5, 2, 3, 4, 7, 8, 9}; This declares an array that defines which Arduino pins are connected to each segment of the display. The order is alphabetical (A, B, C, D, E, F, G, DP where DP is the decimal point). So in this case, Arduino pin 6 connects to segment A, pin 5 connects to segment B, pin 2 connects to segment C, and so on.

resistorsOnSegments = true; This needs to be set to true if your current limiting resistors are in series with the segment pins. If the resistors are in series with the digit pins, set this to false. Set this to true when using multi-digit displays.

sevseg.setBrightness(90); This function sets the brightness of the display. It can be adjusted from 0 to 100.

sevseg.setNumber(); This function prints the number to the display. For example, sevseg.setNumber(4); will print the number “4” to the display. You can also print numbers with decimal points. For example, to print the number “4.999”, you would use sevseg.setNumber(4999, 3);. The second parameter (the 3) defines where the decimal point is located. In this case it’s 3 digits from the right most digit. On a single digit display, setting the second parameter to “0” turns on the decimal point, while setting it to “1” turns it off.

sevseg.refreshDisplay(); This function is required at the end of the loop section to continue displaying the number.

COUNT UP TIMER

```
#include "SevSeg.h"
SevSeg sevseg;

void setup(){
   byte numDigits = 1;
   byte digitPins[] = {};
   byte segmentPins[] = {6, 5, 2, 3, 4, 7, 8, 9};
   bool resistorsOnSegments = true;

   byte hardwareConfig = COMMON_CATHODE;
   sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments);
   sevseg.setBrightness(90);
}

void loop(){
   for(int i = 0; i < 10; i++){
      sevseg.setNumber(i, i%2);
      delay(1000);
      sevseg.refreshDisplay();
      }
}
```