

**Computer Systems Engineering Technology**

**CST 120 – Embedded C Programming**

|  |  |
| --- | --- |
| Lab 03 – Seven Segment Display | Name\_\_Chris Thomas\_\_\_\_\_\_ |
|  | Due Date: |
| Instructor: G Drouant Turn in Monday (4/18) by 11:59pm | |
| Possible Points: 100 | |
|  | |

# Remember to use Lecture Notes from Monday’s Lab as a reference for this Lab.

# Parts List

You will need:

7– 1K Resistors

1 -Seven Segment Display

1-Pushbutton Switch

1 – Active Buzzer (unwashed)

1 – Diode (1N4001)

1– Arduino UNO

1-Protoboard

Wires

Voltmeter

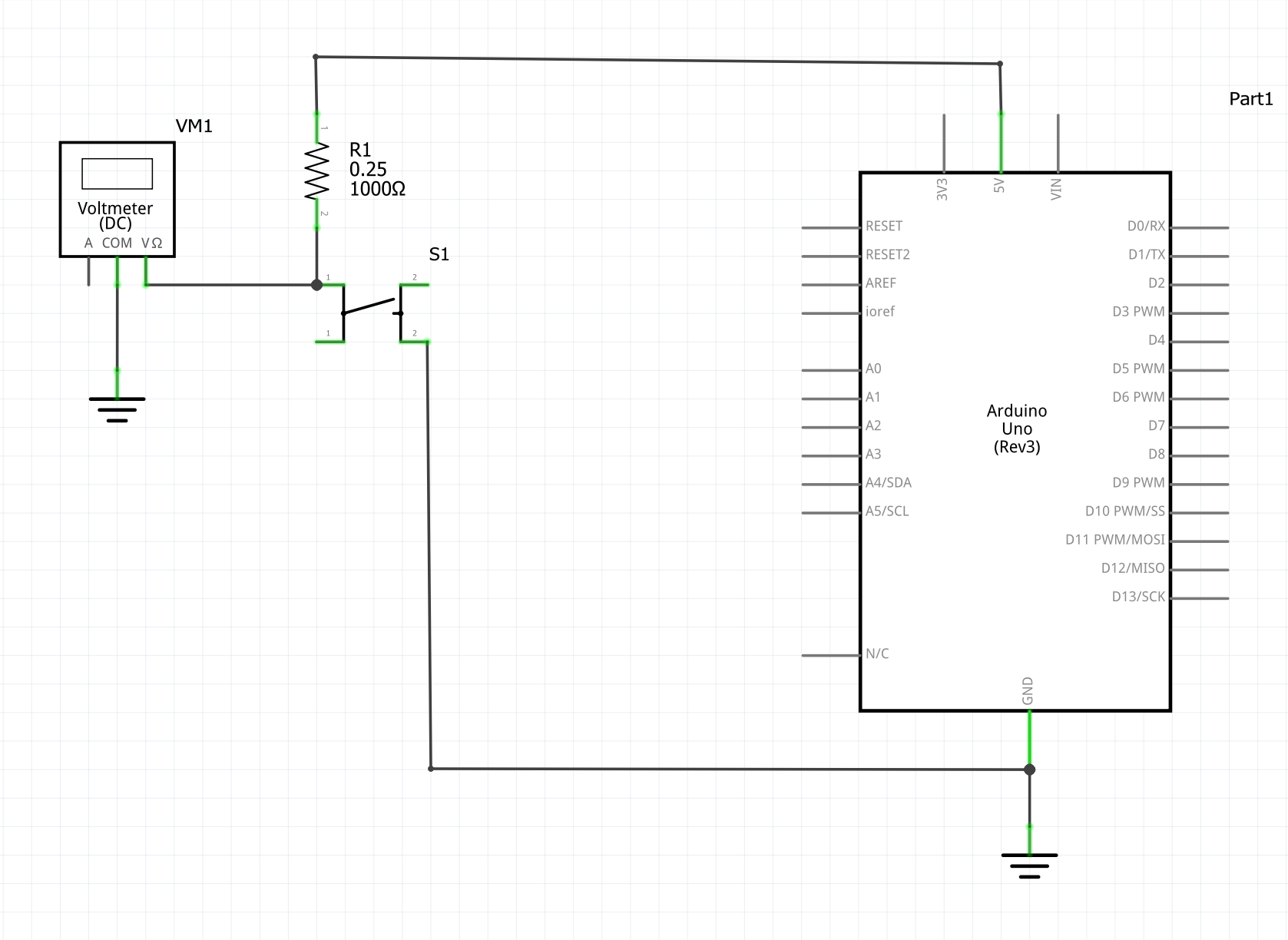
**Part 1**

Using the UNO, a pushbutton switch and a 1000 Ohm resistor construct the circuit in the following schematic.

In this circuit we will be using the UNO as a power supply. Be sure NOT TO CONNECT the UNO’s 5V pin and it’s ground pin. If you do, you will damage the UNO.

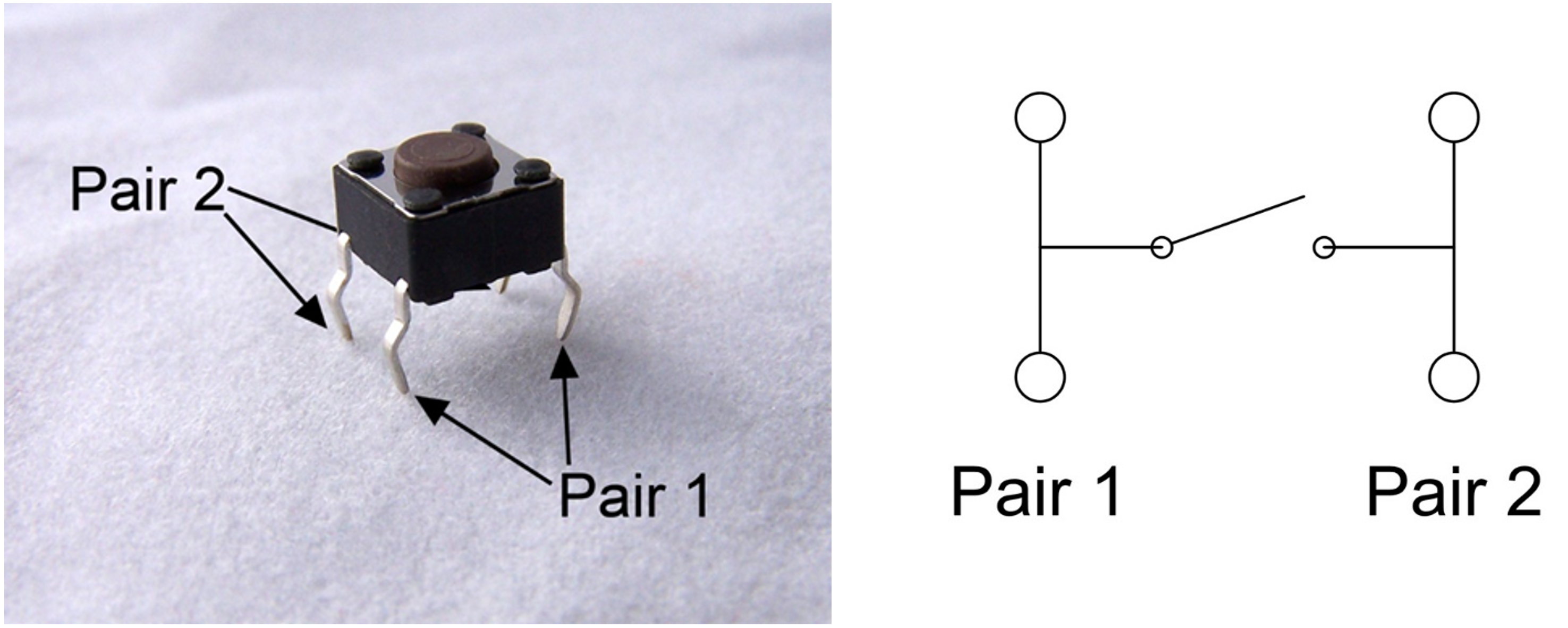
After building and checking the circuit, plug the UNO into your computer’s USB Port for power. Use the Digital Voltmeter (DVM) to measure the voltage across the switch while the pushbutton is not pressed. What is the voltage across the switch (Red probe from the voltmeter touching the connection of the resistor and switch and the Black probe of the voltmeter on Ground)?\_\_4.95V

Next, measure the voltage across the switch while the pushbutton switch is being pressed. What is the voltage across the switch?\_\_\_\_\_0V



Part 1 Signoff\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17.32\_

# 



# Part 2

The objective of Lab 3 is to build a 10 second countdown display using the UNO and a seven segment display that will count from 9 to 0 after a pushbutton switch has been pressed. Each digit will be displayed for one second. When the count reaches 0, the active buzzer will buzz for one second and then stop buzzing. The count will stay at a count of zero – frozen in place.

Please wire the board according to the preceding schematic.

Wire the switch and the active buzzer and 1N4001 diode before you wire the seven segment display.

When wiring the Seven Segment Display to the Arduino UNO proceed slowly and cautiously, one display segment at a time. Time spent carefully building this circuit will save a lot of time in troubleshooting the circuit.

The table below shows the relationship between the Display segment name and the Display pin to which each segment is connected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | Segment Name | 7 Seg Display Pin | Arduino Digital Pin | Arduino PORT Pin |  |
|  | A | 7 | 8 | PB0 |  |
|  | B | 6 | 9 | PB1 |  |
|  | C | 4 | 10 | PB2 |  |
|  | D | 2 | 11 | PB3 |  |
|  | E | 1 | 12 | PB4 |  |
|  | F | 9 | 13 | PB5 |  |
|  | G | 10 | 7 | PD7 |  |
|  |  |  |  |  |  |

The code below was demonstrated in Monday’s lecture. Some of the code has been removed so that you can figure out how to make the code work. The code snippets for displaying 0 and 2 on the Seven Segment Display has been left intact to give you an idea of how the code works. The comments are there to help you “fill in the blanks.”

You are permitted to use the code below, but you are not required to use the code. You may come up with a completely different program. The program must perform the countdown timer function and it must be very well documented.

/\*

\* Seven\_SegmentDisplay\_2.c

\*

\* Created: 4/10/2022 3:32:06 PM

\* Author : drouantg

\*/

// Defines

#define *F\_CPU* 16000000UL

//The following definitions relate the Display Segment

// Name to the hex code used to "light up" that segment

#define A 0x01

#define B 0x02

#define C 0x04

#define D 0x08

#define E 0x10

#define F 0x20

#define G 0x80

// Include Files

#include <avr/io.h>

#include <util/delay.h>

int i;

int main(void)

{

DDRB = \*\*\*\*;//sets pins 0 thru 5 of PORTB as OUTPUTS

DDRD = DDRD | \*\*\*\*;//PD7(output)->segment G; PD6(output)->buzzer; PD5(input)->switch

DDRD = (DDRD & \*\*\*\*);//sets PD5 as an input leaves other bits unchanged

PORTD=PORTD | \*\*\*\*;//turns on pullup resistor connected to pd5

*\_delay\_ms*(1);

while((PIND & 0x20)){}//wait for button to be pressed

//for loop steps through cases 0 through 9

//each case produces a number 0 through 9 on the display

//the numbers are displayed in reverse order

//9,8,7,6,5,4,3,2,1,0

for (i=9;i>-1;i--)

{

switch (i)

{

case 0: //number zero

PORTB=0x00; //turn off all segments connected to PORTB

PORTD=PORTD & 0x7F; //turn off segment G connected to PORTD

PORTB = A|B|C|D|E|F; //turn on display segments controlled by PORTB

*\_delay\_ms*(1000);

break;

case 1: //number 1

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB=\*\*\*\*; //turn on display segments controlled by PORTB

*\_delay\_ms*(1000);

break;

case 2: //number 2

PORTB=0x00; //turn off all segments connected to PORTB

PORTD=PORTD & 0x7F; //turn off segment G connected to PORTD

PORTB = A|B|E|D; //turn on display segments controlled by PORTB

PORTD = PORTD | G; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 3: //number 3

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 4: //number 4

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 5: //number 5

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 6: //number 6

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 7: //number 7

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

*\_delay\_ms*(1000);

break;

case 8: //number 8

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

case 9: //number 9

PORTB=\*\*\*\*; //turn off all segments connected to PORTB

PORTD=PORTD & \*\*\*\*; //turn off segment G connected to PORTD

PORTB = \*\*\*\*; //turn on display segments controlled by PORTB

PORTD = \*\*\*\*; //turn on display segment controlled by PORTD

*\_delay\_ms*(1000);

break;

}

}

PORTD =(PORTD | 0x40); //turn on active buzzer

*\_delay\_ms*(1000); //wait for one second

PORTD = (PORTD & 0xBF); //turn off active buzzer

while(1)

{

}

}

PART 2 Signoff\_\_\_\_\_\_\_\_\_17.32\_\_\_\_\_\_\_\_\_\_\_\_\_

**What to send in to Canvas:**

1. This Word document with answers.
2. Your main.c code.