Directions: Submit a report using the link below. Name file: **P1\_lastname**.

**Part 1: Create a 1ms delay function**• Use 2 nested for loops to create a 1ms delay that can create a delay between 1ms and 1s, as accurately as possible.  
• Report on the timing error for 1ms, 1s, 60s.

// Project 1, Questions 1

#define *F\_CPU* 16000000UL // 16 MHz

#include <avr/io.h>

#include <util/delay.h>

#include <stdint.h>

void delay1ms(int ms\_count){

for(int i = 0; i<ms\_count; i++){

for(int j = 0; j<1599; j++){

DDRB = 0;

}

}

}

**Answers:**

1ms = 1.001.25 µs



10ms = 10,002.38 µs



60ms = 60,008.63 µs



1 second = 100,013.63 µs



**Part 2: Interfacing a common anode 7-segment Display (7SD)**• Review the class notes describing simple parallel interfacing. Pay attention to the section on 7-segment Display.  
• Use the internet (www.alldatasheet.com) to find the datasheet for the chip. Use the datasheets to answer/do the following: What is the typical current to light-up a segment on the chip?  
• Draw the chip face-up (show the 8 pattern and the 2 dots). On the diagram identify the location of all pins. Label the pins.  
• Based on the typical voltage/current required by the chip what resistance is suitable when interfacing the chip with the IO ports?  
• The 7SD is a common anode device. Common anode means that the all the positive side of the LEDs that make up the device are connected together. Therefore, to switch a segment on the common anode pin must be connected to HI and the segment pin must be connected to LO.  
• By connecting the anode to 5V, the IO port will have to sink (ground) the current for the segment to turn ON. Must make sure that the current into the port is not greater than IIL. Answer the question: if IIL = 2.5mA and common anode is connected to 5V, what is the required resistance? Compare that value with that derived above.  
• To turn a segment OFF, the port must switch the pin ON. The main advantage of sinking over sourcing is that the MCU conserves output current (and thus can drive more devices).  
• Based on the common anode 7SD and assuming that segment a is connected to pin 0 of the IO port, derive a table to display 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, c, e, f.

**Answers:**

1. Typical light up current = 20 mA for forward-bias voltage
2. Diagram:

A screenshot of a computer

Description automatically generated

1. Typical voltage/current = ~2.2VF / 20 mA = 110Ω. The range on the datasheet varies from about 1.7V­­min and 2.8Vmax. If we use the min values, then 1.7V / 20 mA = 85Ω.
2. For an IIL = 2.5 mA and digital IO, we can say: 5V / 2.5 mA = 2kΩ. This number is significantly higher (by a margin of ~1/20th) mainly due to the far lower current. This level of current needs a lot of resistance to get to such a small value.

Common Anode Display Table

|  |  |  |
| --- | --- | --- |
| Digit | Binary | Hex |
| 0 | 0b1000000 | 0x40 |
| 1 | 0b1111001 | 0x79 |
| 2 | 0b0100100 | 0x24 |
| 3 | 0b0110000 | 0x30 |
| 4 | 0b0011001 | 0x19 |
| 5 | 0b0010010 | 0x12 |
| 6 | 0b0000010 | 0x02 |
| 7 | 0b1111000 | 0x78 |
| 8 | 0b0000000 | 0x00 |
| 9 | 0b0010000 | 0x10 |
| A | 0b0001000 | 0x08 |
| b | 0b0000011 | 0x03 |
| c | 0b0100111 | 0x27 |
| e | 0b0000100 | 0x04 |
| f | 0b0001110 | 0x0E |

**Part 3: 4xDip Switch Input, 7SD**  
• Review the class notes describing simple parallel interfacing. Pay attention to the section on DIP switch interfacing.  
• Look at the quad DIP switch example in the lecture. Note that the when the DIP switch is OFF the pin is HI and when it is ON the pin is LO.  
• Use your board’s manual to investigate port assignment and select ports that can be used for input and output.

• Write a program that uses one of the IO ports for input and another for output. The input port should read 4 switches DIP switches (1-4, where switch 1 is considered LSB and switch 4 is MSB). Convert the switches input to a hex number and display that number on the common anode 7SD.  
• Draw a pin to pin diagram showing ports used, 7SD, DIP, GND, VCC.  
• Test your design and produce images in your report to show that the efficacy of your design.  
• Take screenshots (DIP + 7SD), showing the results for each input.

**Answers:**

// Project 1: Question 3

#define *F\_CPU* 16000000L //16MHz

#include <avr/io.h>

#include <util/delay.h>

#include <stdint.h>

int main(void){

// 0 1 2 3 4 5 6 7 8 9 A B C D E F

*uint\_8t* SSD[] = {0x7E, 0x30, 0x6D, 0x79, 0x33, 0x5B, 0x5F, 0x70, 0x7F, 0x7B, 0x77, 0x1F, 0x4E, 0x3D, 0x4F, 0x47};

*uint8\_t* DIP4\_value;

DDRB &= 0xF0; // Port B = DIP4 Input

DDRA |= 0x7F; // Port A = 7SD Output

DDRC = 1; // 7SD Enable

PORTC = 1;

while (1){

DIP4\_value = (*uint8\_t*)(PINB & 0x0F);

PORTA = ~SSD[DIP4\_value];

}

}

A computer screen shot of a black screen

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A computer circuit board with wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer screen shot of a blue circuit board

Description automatically generated

A computer screen shot of a circuit board

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer screen shot of a blue circuit board

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

**Part 4: Use 4x7SD, 8xDIP Switch**  
• Show the table for digits 0-9 of a common cathode 7SD.  
• Draw a pin to pin diagram showing ports used, 7SD, DIP, GND, VCC.  
• Write a program that:  
1. Reads the 8 DIP switches as an 8-bit signed binary number.  
2. Display the signed number in decimal on 4x7SD. Use the leftmost 7SD on your board for the negative sign (‘–’). The remaining 7SD for the number.  
3. Example: if the 8 DIP switches are set ON as follows: 1111111 then, the 7SD display will show: –001.  
• Take screenshots (DIP +7SD) showing the results for positive values, negative values, 0.

**Answers:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal # | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 7SD # | 0x7E | 0x30 | 0x6D | 0x79 | 0x33 | 0x5B | 0x5F | 0x70 | 0x7F | 0x7B |

A computer screen shot of a computer

Description automatically generated

// Project 1: Question 4

#define *F\_CPU* 16000000L //16MHz

#include <avr/io.h>

#include <util/delay.h>

#include <stdint.h>

void delay1ms(int ms\_count){

for(int i = 0; i<ms\_count; i++){

for(int j = 0; j<1599; j++){

PORTF = 0;

}

}

}

int main(void)

{

*uint8\_t* SSD[] = {0x7E, 0x30, 0x6D, 0x79, 0x33, 0x5B, 0x5F, 0x70, 0x7F, 0x7B, 0x01};

*int8\_t* DIP8\_value;

*uint8\_t* SD\_zero, SD\_one, SD\_two, SD\_three;

DDRA = 0XFF; // 7SD Output Pins

DDRB = 0; // DIP Inputs

DDRC = 0X0F; // 7SD Enable pins

DDRD = 1;

PORTD = 1;

while(1)

{

DIP8\_value = PINB;

// Edge Casing for Negative and -128

if (DIP8\_value < 0)

{

SD\_three = SSD[10];

if(DIP8\_value != -128){

DIP8\_value = ~DIP8\_value + 1;

}

else{

DIP8\_value = 128;

}

}

else

{

SD\_three = 0;

}

// Forces display value to be unsigned for -128

*uint8\_t* display = DIP8\_value;

// Sign 7SD (-/+)

PORTA = (*uint8\_t*)SD\_three;

PORTC = ~0x1;

delay1ms(1);

// MSD 7SD

SD\_two = (*uint8\_t*)(display/100);

PORTA = SSD[SD\_two];

PORTC = ~0x02;

delay1ms(1);

// Middle Digit 7SD

SD\_one = (*uint8\_t*)(display/10)%10;

PORTA = SSD[SD\_one];

PORTC = ~0x04;

delay1ms(1);

// LSD 7SD

SD\_zero = (*uint8\_t*)display%10;

PORTA = SSD[SD\_zero];

PORTC = ~0x08;

delay1ms(1);

}

}

A computer circuit board with wires and wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated

A computer circuit board with many wires

Description automatically generated