

# Midterm 1

Student Name: Meral Abu-Jaser

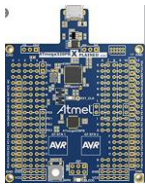
Student #: 5003137888

Student Email: abujaser@unlv.nevada.edu

Primary Github address: [https://github.com/MeralAbuJaser/Submission\\_da.git](https://github.com/MeralAbuJaser/Submission_da.git)

Directory: [https://github.com/MeralAbuJaser/Submission\\_da/tree/master/Midterm%201/Midterm1](https://github.com/MeralAbuJaser/Submission_da/tree/master/Midterm%201/Midterm1)

## 1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS



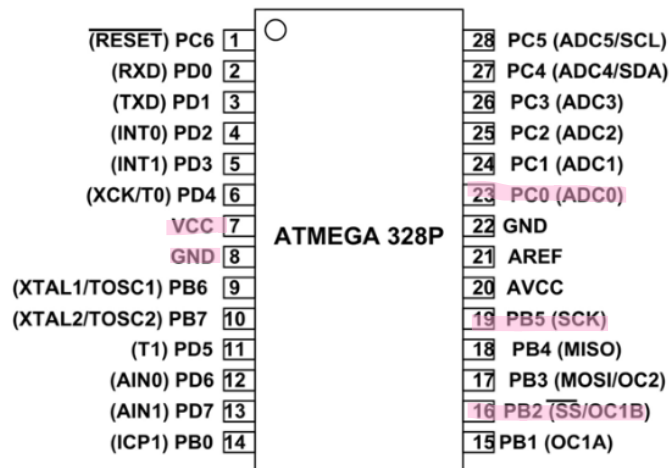
Atmega 328pb

### Atmel Studio 7.0

- debugger
- simulator
- assembler
- programmer
- terminal window

### Additional components

- LM35



## 2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
/*
 * Midterm1.c
 *
 * Created: 4/18/2020 1:12:20 am
 * Author: Meral
 */

#define F_CPU 16000000UL
#define BAUD 9600
#include <stdio.h>
#include <stdlib.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/setbaud.h>
#include <util/delay.h>

volatile uint8_t counter =0; //global variable
volatile float adc_fah;//declare float value for Fahrenheit temp
volatile float adc_cel;//declare float value for celcius temp
volatile float temp;

void USART_init(void){
    UBRR0H = UBRRH_VALUE;
    UBRR0L = UBRL_VALUE;
    UCSR0C = _BV(UCSZ01) | _BV(UCSZ00); //8-bit data
    UCSR0B = _BV(RXEN0) | _BV(TXEN0); //Enable RX and TX
}

//Send data to the serial port
void USART_tx_string( char *data ){
    while ((*data != '\0')){ //while the register is empty enter date
        while (!(UCSR0A & (1 <<UDRE0)));
        UDR0 = *data;
        data++; //increment data location forward
    }
}

ISR(TIMER0_OVF_vect){
    counter++; //increment counter
}

// Initialize ADC
void adc_init(void) {
    /**Setup and enable ADC**/
    ADMUX = (0<<REFS1) | //Reference selection bits
    (1<<REFS0) | //AVcc - external cap at AREF (5)V
    (0<<ADLAR) | //ADC right adjust result
    (1<<MUX2) | //Analog channel selection bits
    (0<<MUX1) | //ADC4 (PC4 PIN27)
    (0<<MUX0);

    ADCSRA = (1<<ADEN) | //ADC enable
    (0<<ADSC) | //ADC start conversion
    (0<<ADATE) | //ADC auto trigger enable
    (0<<ADIF) | //ADC interrupt flag
    (0<<ADIE) | //ADC interrupt enable
    (1<<ADPS2) | //ADC Prescaler select bits
    (1<<ADPS1) | //128 AS PRESCALAR SELECIION BIT
    (1<<ADPS0); //Select channel
}
```

```

void read_adc(void){
    unsigned char i = 10;
    adc_fah = 0;
    adc_cel = 0;
    temp = 0;
    while(i--){
        ADCSRA |= (1<<ADSC);
        while(ADCSRA & (1<<ADSC));
        temp += ADC;
    }
    adc_fah = (temp/6)-4; //convert ADC value to fahrenheit
    adc_cel = (temp/18); //convert fahrenheit to celcius
}

unsigned char USART_Receive( void ){
    while ( !(UCSR0A & (1<<RXC0)) ); //data is being received
    return UDR0; //return received data
}

unsigned char usart_sendbyte(void){
    unsigned char data = 0;
    if(UCSR0A & (1<<RXC0))
        data = 1;
    return data;
}

void display(){
    char celcius[20]; //character array for celcius degree
    char fahrenheit[20]; //character array for fahrenheit degree
    int delay; //to store the user input for the delay in seconds

    if(usart_sendbyte()){
        switch(USART_Receive()){
            case 'h': //if user enters h display the instructions on screen
                USART_tx_string("h -Help Screen \n");
                USART_tx_string("t -Display temperature in Centigrade \n");
                USART_tx_string("T -Display temperature in Fahrenheit \n");
                USART_tx_string("o -Turn on LED \n");
                USART_tx_string("O -Turn off LED \n");
                USART_tx_string("s -Send a string to the terminal \n");
                USART_tx_string("i -Send an integer to the terminal (you can choose from 1-5)\n");
                break;
            case 't': //if user enters t display the degree in Celsius
                read_adc(); //call the read adc function to read the temperature in Celsius
                USART_tx_string("Celcius degree: "); //display on screen
                snprintf(celcius, sizeof(celcius), "%f\r\n", adc_cel); //display on screen
                USART_tx_string(celcius); //prints the Celsius value
                USART_tx_string("\n");
                break;
            case 'T': //if user enters t display the degree in fahrenheit
                read_adc(); //call function to read the temperature in fahrenheit
                USART_tx_string("Fahrenheit degree: "); //display on screen
                snprintf(fahrenheit, sizeof(fahrenheit), "%f\r\n", adc_fah); //display
                USART_tx_string(fahrenheit); //prints the fahrenheit value
                USART_tx_string("\n");
                break;
            case 'o': //if user enter o turn on the LED
                PORTB &= ~(1<<5); //turn LED pin 13 on
                break;
            case 'O': //if user enter o turn off the LED
                PORTB |= (1<<5); //turn LED pin 13 off
                break;
            case 's': //send string to screen
                USART_tx_string("This is midterm 1 \n"); //display String
                break;
        }
    }
}

```

```

        //this part of the case's depened on the user inputs for the delays's in seconds
        case '1':
            DDRB |= (1<<PB2); //enable pb2
            PORTB &= ~(1<<PB2); //LED off
            _delay_ms(1000); //1s delay
            PORTB |= (1<<PB2); //LED on pin 12
            _delay_ms(1000); //1s delay
            break;
        case '2':
            DDRB |= (1<<PB2); //enable pb2
            delay = 2; //delay = the delay enters by the user
            while(delay--){ //2s loop to turn LED on and off
                PORTB &= ~(1<<PB2); //LED off
                _delay_ms(2000); //2s delay
                PORTB |= (1<<PB2); //LED on pin 12
                _delay_ms(2000); //2s delay
            }
            break;
        case '3':
            DDRB |= (1<<PB2); //enable pb2
            delay = 3; //delay = the delay enters by the user
            while(delay--){ //3s loop to turn LED on and off
                PORTB &= ~(1<<PB2); //LED off
                _delay_ms(3000); //3s delay
                PORTB |= (1<<PB2); //LED on pin 12
                _delay_ms(3000); //3s delay
            }
            break;
        case '4':
            DDRB |= (1<<PB2); //enable pb2
            delay = 4; //delay = the delay enters by the user
            while(delay--){ //4s loop to turn LED on and off
                PORTB &= ~(1<<PB2); //LED off
                _delay_ms(4000); //4s delay
                PORTB |= (1<<PB2); //LED on pin 12
                _delay_ms(4000); //4s delay
            }
            break;
        case '5':
            DDRB |= (1<<PB2); //enable pb2
            delay = 5; //delay = the delay enters by the user
            while(delay--){ //4s loop to turn LED on and off
                PORTB &= ~(1<<PB2); //LED off
                _delay_ms(5000); //5s delay
                PORTB |= (1<<PB2); //LED on pin 12
                _delay_ms(5000); //5s delay
            }
    }
}

int main(){
    USART_init(); //call function to initialize
    TCCR0A = 0x00; //normal mode timer
    TCCR0B = 0x05; //set pre-scaler = 1024
    TCNT0 = 0; //counter = 0
    TIMSK0 = (1<<TOIE0); //enable interrupt
    sei(); //enable global
    adc_init();//initialize adc

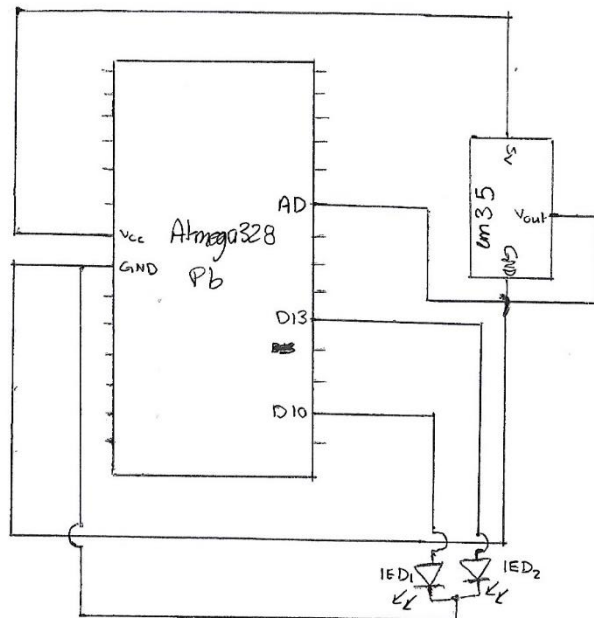
    DDRB = 0xFF; //Enable PB5
    PORTB = 0xFF; //Enable PB5 and set high
    while(1){
        display(); //to display to the terminal
    }
    return 0;
}

```

### 3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

N/A

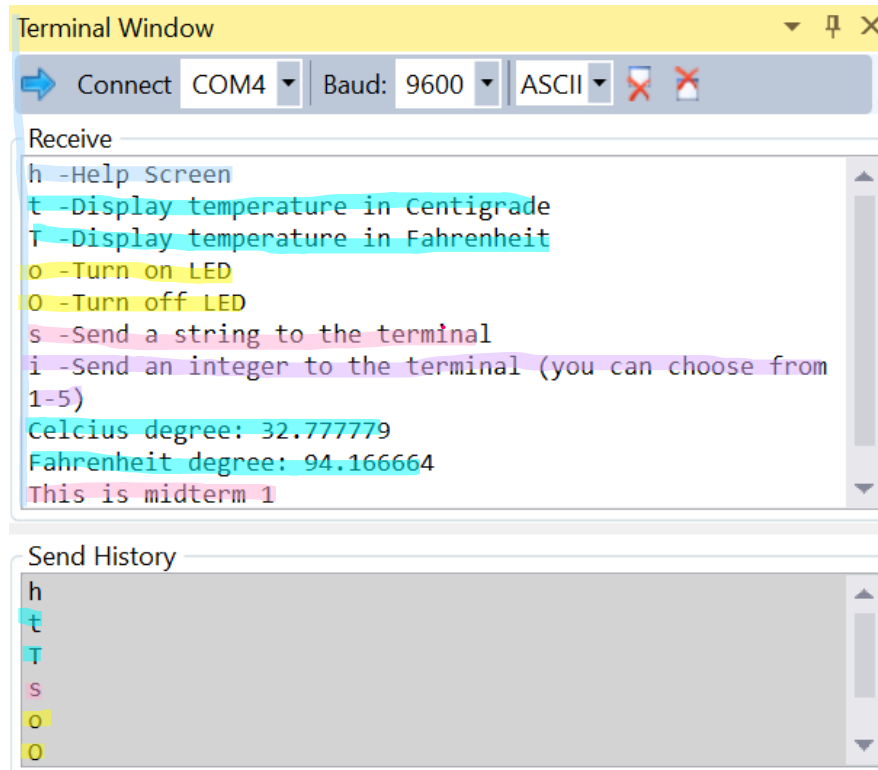
### 4. SCHEMATICS



### 5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

All commands printed to terminal

#### Task 1



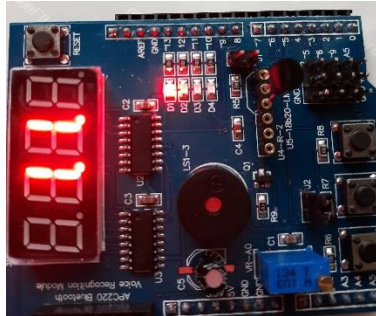
Task 2, task 3 task 4, task 5

## Simulation

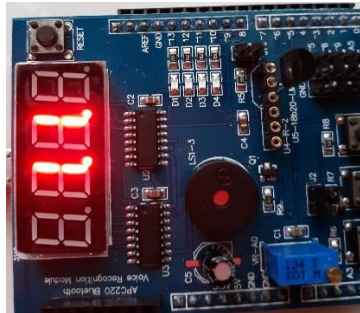
```
Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "Midterm1.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$ (PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "c:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "c:\Users\Meral\Documents\Midterm1\Midterm1.cproj"
Done building target "Build" in project "Midterm1.cproj".
Done building project "Midterm1.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

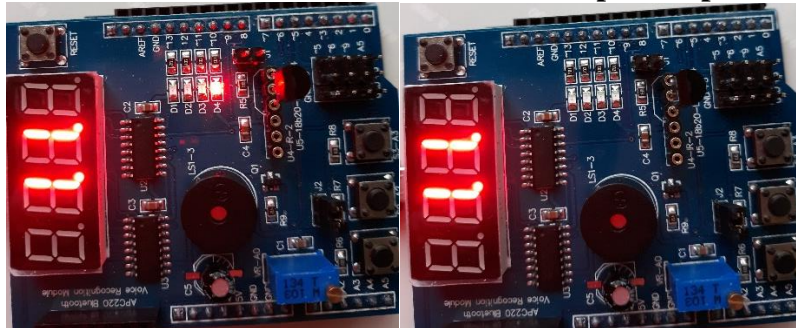
## When user enters o



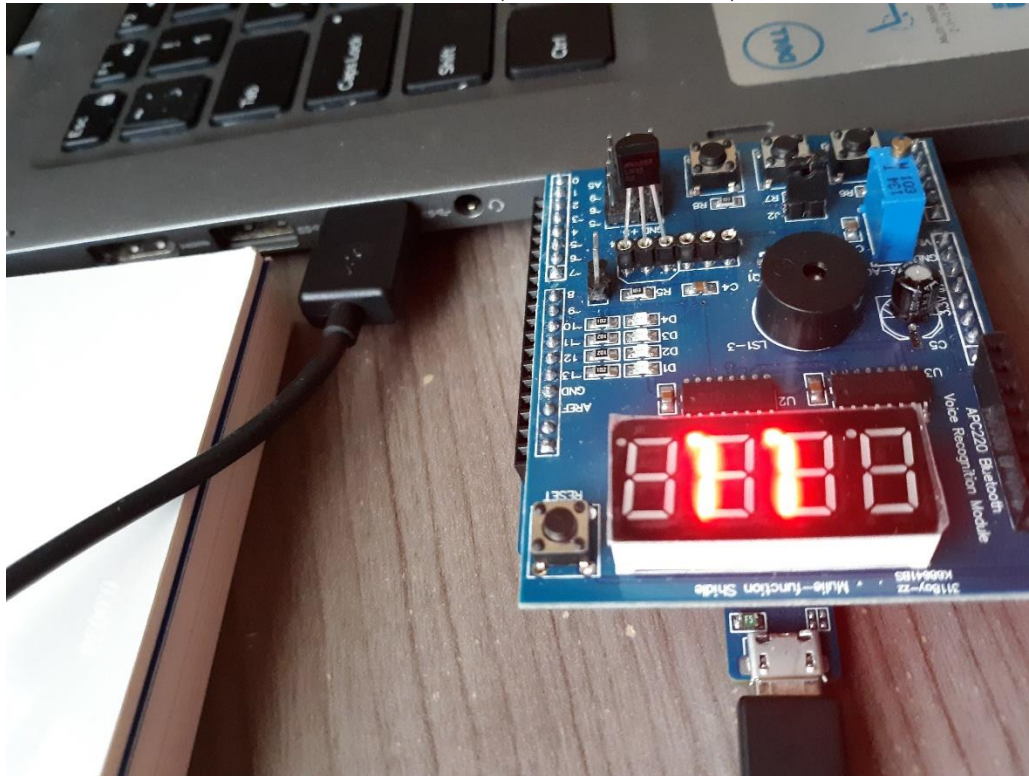
## When user enters O



## When user enters an int the LED where the speed depends on the delay



**6. SCREENSHOT OF EACH DEMO (BOARD SETUP)**



**7. VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=6ciR1NqunQg>

**8. GITHUB LINK OF THIS DA**

[https://github.com/MeralAbuJaser/Submission\\_da/tree/master/Midterm%201/Midterm1](https://github.com/MeralAbuJaser/Submission_da/tree/master/Midterm%201/Midterm1)

*"This assignment submission is my own, original work".*

Meral Abu-Jaser