CPE301 – SPRING 2020

MIDTERM 1

Student Name: Minsung Cho

Student #: 2001446442

Student Email: chom3@unlv.nevada.edu

Primary Github address: <https://github.com/cho-minsung>

Directory: https://github.com/cho-minsung/midterm1

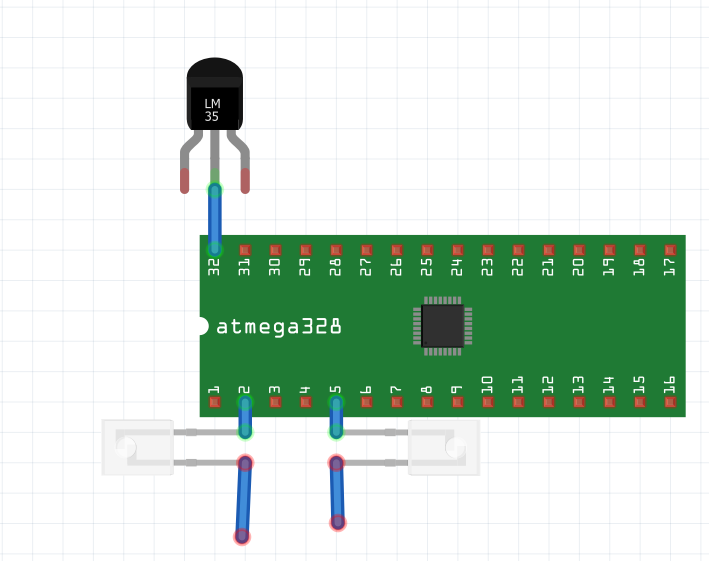
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used:

- LM35

- ATMEGA328PB

- Arduino shield



the LM35 isn’t connected to 32. It is just for the diagram’s sake.

Block diagram with pins used in the Atmega328P

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

All the previous codes were used as initial codes.

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

/\*

\* midterm1.c

\*

\* Created: 3/30/2020 7:11:20 PM

\* Author : Minsung Cho

\*/

#define *F\_CPU* 16000000

#define BAUD 9600

#include <util/setbaud.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/setbaud.h>

#include <stdio.h>

*uint8\_t* ascii\_2\_int(*uint8\_t* n)

{

if(n >= 48 && n <= 57) //outside of integers

{

send\_byte();

return n - 48;

}

return 1;

}

void send\_byte(*uint8\_t* byte)

{

if(UCSR0B & (1<<TXEN0))

{

while(!(UCSR0A & (1<<UDRE0)));

UDR0 = byte;

}

}

void send\_str(char string[])

{

int i = 0;

while (string[i] != '\0')

send\_byte(string[i++]);

}

char get\_byte()

{

while(!(UCSR0A & (1<<7)));

return UDR0;

}

*uint16\_t* analog\_conv()

{

ADCSRA |= (1<<6); //ADSC is enabled for conversion.

while (!(ADCSRA&(1<<4)));

ADCSRA |= (1<<4); //ADIF flag is cleared.

return ADC;

}

float dig\_2\_temp(*uint16\_t* temp, char mode)

{

float temp\_converted = temp\*110/1023;

switch (mode) {

case 't':

return temp\_converted; //Temperature in Celsius is returned.

case 'T':

default:

return temp\_converted\*1.8 + 32; //Temperature in Fahrenheit is returned.

}

}

int main(void)

{

//ADC is enabled here

DDRC &= ~(1<<4); //pin4 is set as input.

ADMUX = 0xC4; //1100 0100, internal 1.1V ref, right justified, ADC4.

ADCSRA = 0x00; //0000 0000, free running.

ADCSRB = 0x87; //1000 0111, enables ADC, prescaler of 128.

//UART is enabled here

DDRD |= 0x02; //port is set.

UCSR0C = 0x06; //Asynchronous, no parity.

//transmitter is enabled here

UCSR0B |= ~(1<<RXEN0);

//receiver is enabled here

DDRD &= ~1;

UCSR0B |= (1<<RXEN0);

//LED is enabled here

DDRB |= (1<<2);

PORTB &= ~(1<<2);

DDRB |= (1<<5);

PORTB &= ~(1<<5);

//timer0 and timer1 are initialized here

OCR1A = 15624; //An approximation of 1 second for 16Mhz 1024 prescaler.

TIMSK1 = 0x02; //compare match A interrupt is enabled.

TIFR1 |= 0x02; //interrupt flag is cleared.

sei();

TCCR1B = 0x0D; //timer1 is set as CTC mode with 1024 prescaler.

TCCR1A = 0x00;

TCNT1 = 0;

//various arrays are set

const char menu[] = "h: Help screen \n"

"t: Displays temperature in C \n"

"T: Displays temperature in F \n"

"o: Turns ON LED on PB5 \n"

"O: Turns OFF LED on PB5 \n"

"s: Sends a string to the terminal \n"

"i: Sends an integer to the terminal \n \n";

volatile char temp\_string[5];

volatile char integer[1];

//menu will pop up

send\_str(menu);

while(1)

{

char instruction = get\_byte();

if(instruction == 'h')

send\_str(menu);

else if(instruction == 't' || 'T')

{

*uint16\_t* temperature = analog\_conv();

float temp\_converted = dig\_2\_temp(temperature, instruction);

*snprintf*(temp\_string, 5, "%f\n", temp\_converted);

send\_str(temp\_string);

send\_byte('\n');

}

else if( instruction == 'o')

PORTB &= ~(1<<5);

else if(instruction == 'O')

PORTB |= (1<<5);

else if(instruction == 's')

send\_str("I love Dr. Venki");

else if(instruction == 'i')

{

*snprintf*(integer, 1, "%d\n", 1);

send\_str(integer);

send\_byte('\n');

}

}

}

ISR(TIMER1\_COMPA\_vect)

{

TIFR1 |= 0x02;

int delay\_time = 0;

delay\_time = (delay\_time + 1) %1;

if (delay\_time == 0)

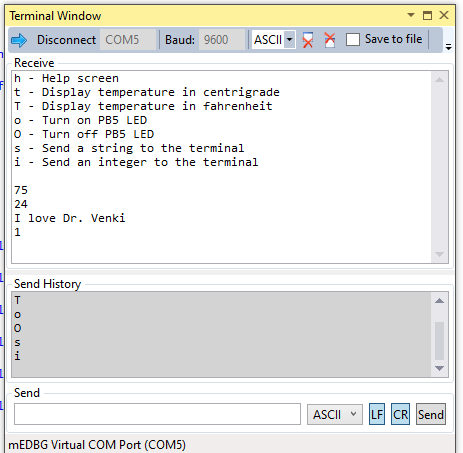
PORTB ^= (1<<2);

}

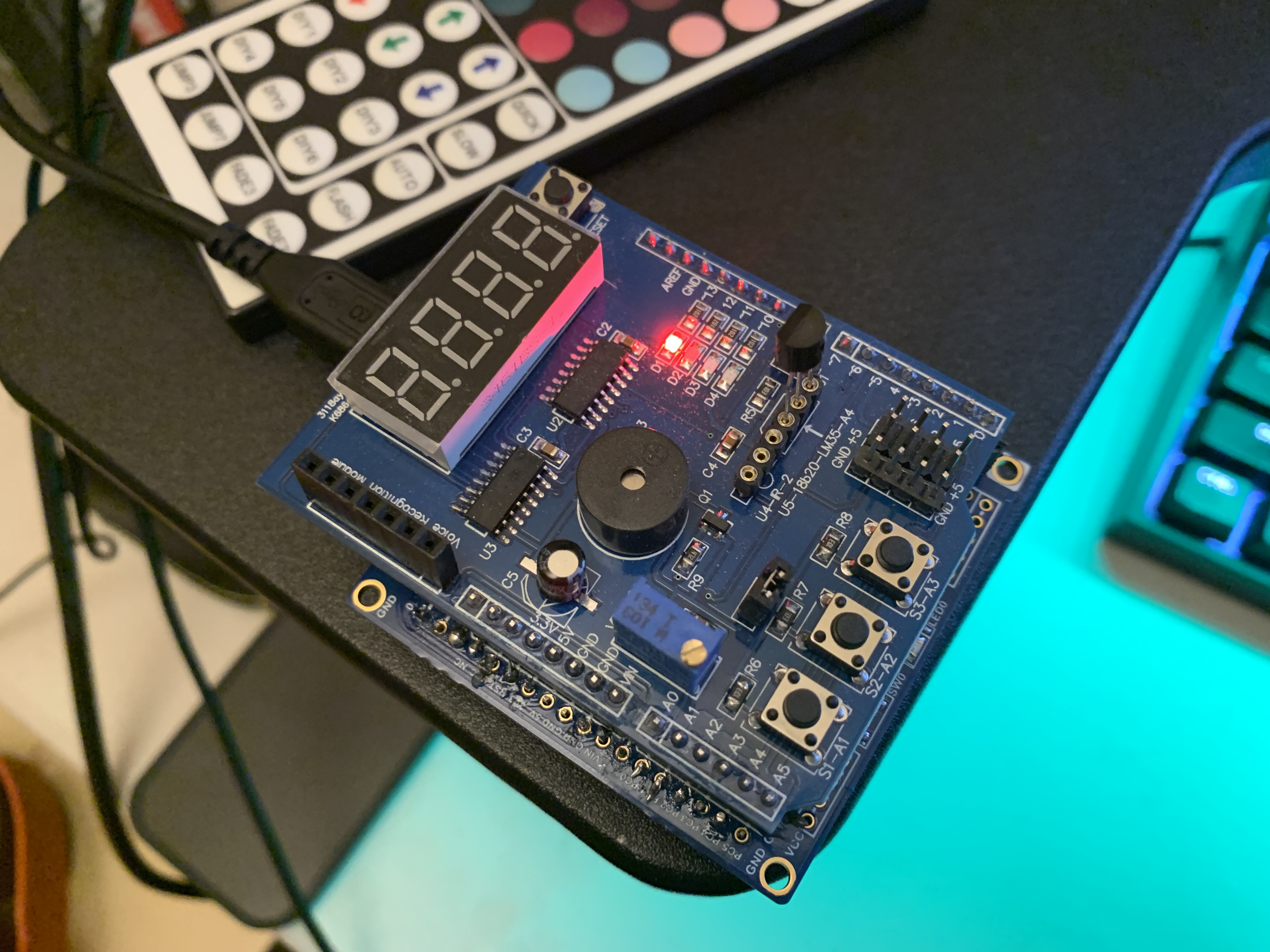
1. **SCHEMATICS**

see the schematic above.

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



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



1. **VIDEO LINKS OF EACH DEMO**

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

1. **GITHUB LINK OF THIS DA**

<https://github.com/cho-minsung/midterm1>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Minsung Cho