

Design Assignment 3B

Student Name: Francisco Mata Carlos

Student #: 1012593607

Student Email: matacarl@unlv.nevada.edu

Primary Github address: https://github.com/chicosisco/da_sub.git

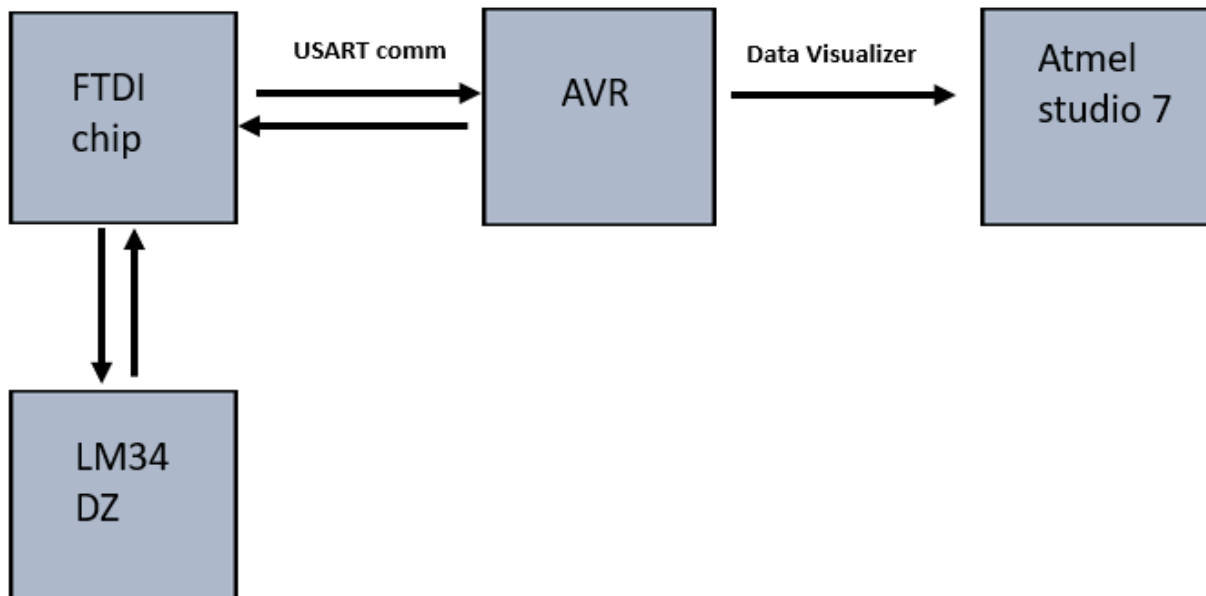
Directory: repository/cpe301/DesignAssignments/DA3B

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment are the next:

- Atmega328p Xplained Mini
- Atmel Studio 7
- FTDI chip
- LM34 DZ

Block diagram with pins used in the Atmega328P



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

Task 1

Write a C AVR program that will monitor the LM34/35 connected to an Analog pin (PC5) to display the temperature in F on the serial terminal every 1 sec. Use a timer with interrupt for the 1 sec delay. Use a FTDI chip for serial to USB conversion.

```
/*
 * DA3B.c
 *
 * Created: 3/29/2019 5:14:00 PM
 * Author : Francisco Mata carlos
 */

/*
 * The program reads the LM34 temperature of the MCU using ADC and sends it to the PC.
 * If you put your finger on the MCU, the number increases.
 */

#define F_CPU 16000000UL
#define BAUD_RATE 9600
#include <avr/interrupt.h>

#include <avr/io.h>
#include <util/delay.h>
int over_flow=0;

void usart_init ();
void usart_send (unsigned char ch);

int main (void)
{
    usart_init ();

    /** Setup and enable ADC **/
    ADMUX = (0<<REFS1)|    // Reference Selection Bits
    (1<<REFS0)|    // AVcc - external cap at AREF
    (0<<ADLAR)|    // ADC Left Adjust Result
    (1<<MUX2)|    // Analog Channel Selection Bits
    (0<<MUX1)|    // ADC5 (PC5 PIN27)
    (1<<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
    (0<<ADPS1)|
    (1<<ADPS0);

    //////////////////////////////////////
    //DDRB |= (1<<DDB4); // setting PB1 as output
}
```

```

TIMSK0 |= (1<<TOIE0);
TCNT0 = 0;           // setting initial value for counter
sei();               // enable global interrupts
TCCR0B |= (1<<CS02)|(1<<CS00); // setting prescaler to 1024

////////////////////////////////////

while (1)
{
    ADCSRA |= (1<<ADSC); //start conversion
    while((ADCSRA & (1<<ADIF)) == 0); //wait for conversion to finish

    ADCSRA |= (1<<ADIF);
    if (over_flow == 61) // when TCNT0 overflows 61 times, then the information
is updated
    {
        int a = ADCL;
        a = a | (ADCH << 8);
        a = (a/1024.0) * 5000/10;
        usart_send((a/100)+'0');
        a = a % 100;
        usart_send((a/10)+'0');
        a = a % 10;
        usart_send((a)+'0');
        usart_send('\r');
        over_flow = 0;
    }

    // _delay_ms(1000);
}
return 0;
}

// timer_0 overflow interrupt
////////////////////////////////////
ISR(TIMER0_OVF_vect)
{
    while (!(TIFR0 & 0X01) == 0);
    TCNT0 = 0X00; //resetting counter to zero
    TIFR0 = 0X01; // reset the overflow flag
    over_flow++; //increasing overflow counter
}

////////////////////////////////////

void usart_init (void)
{
    UCSRB = (1<<TXEN0);
    UCSRC = (1<< UCSZ01)|(1<<UCSZ00);
    UBRR0L = F_CPU/16/BAUD_RATE-1;
}

void usart_send (unsigned char ch)
{

```

```

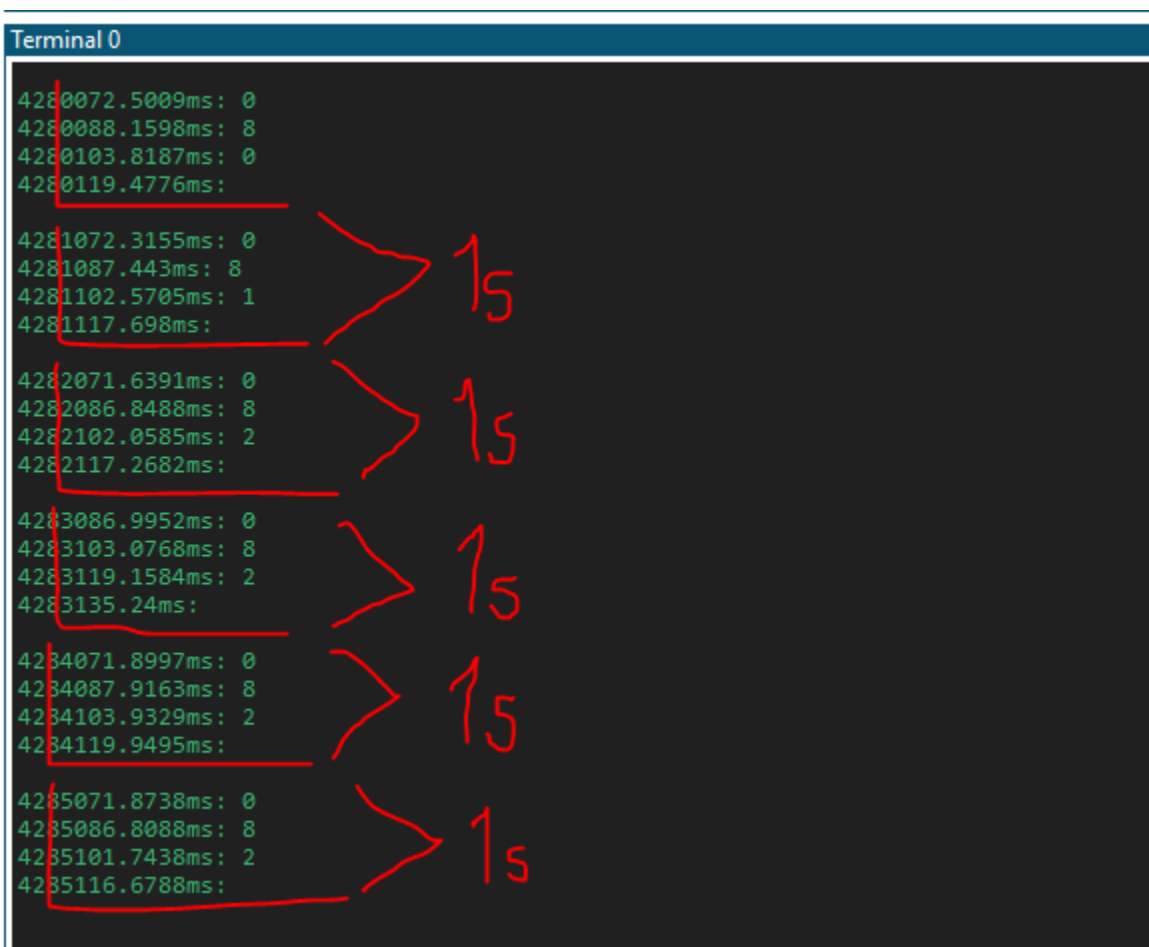
    while (! (UCSR0A & (1<<UDRE0))); //wait until UDR0 is empty
    UDR0 = ch;                       //transmit ch
}

void usart_print(char* str)
{
    int i = 0;
    while(str[i] != 0)
        usart_send(str[i]);
}

```

Task 2

Use the ATMELE Studio Data Visualizer or any Charting program to display the values in time.

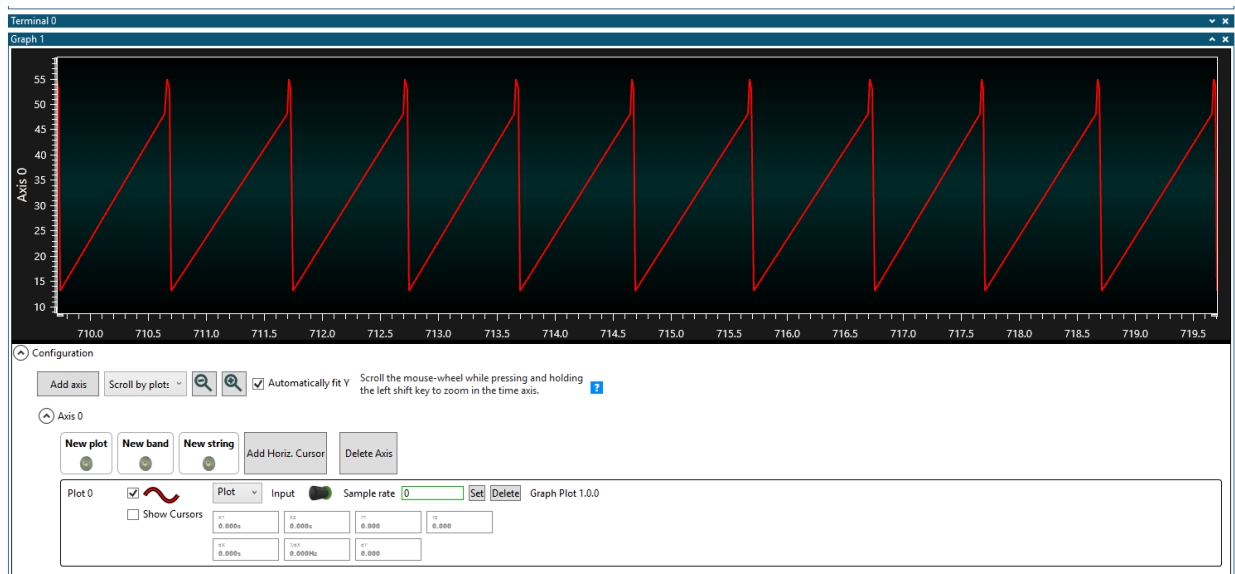
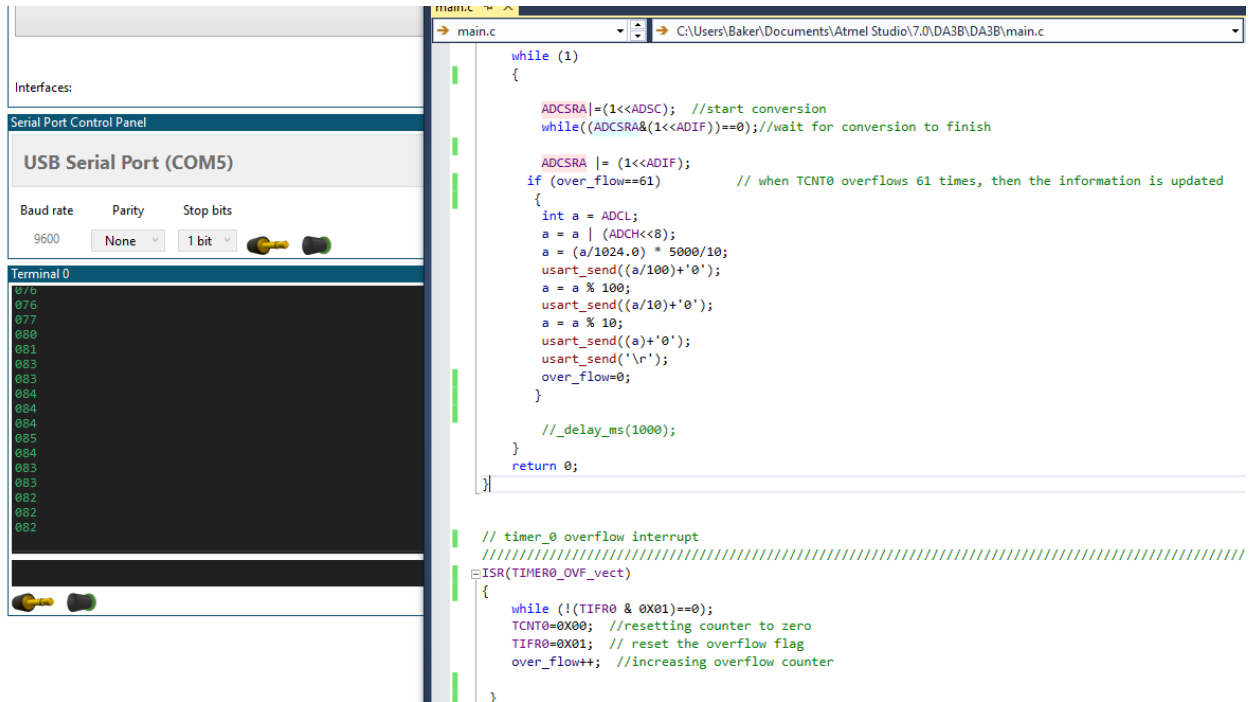


3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Same as above

4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1 C code



Task 2 C code

Serial Port Control Panel
USB Serial Port (COM5)
Baud rate: 9600
Parity: None
Stop bits: 1 bit
Terminal 0

```

3915508.8453ms: 0
3915523.9979ms: 8
3915539.1505ms: 3
3925554.3031ms: 3
3916541.217ms: 0
3926556.6883ms: 8
3926570.9596ms: 3
3926585.8309ms: 3
3927509.3761ms: 0
3927525.2117ms: 8
3927541.0473ms: 3
3927556.6829ms: 3
3918540.3631ms: 0
3918556.1803ms: 8
3918571.9975ms: 3
3928587.6147ms: 3
3929492.3808ms: 0
3929522.7839ms: 8
3929554.1953ms: 2
3929585.6067ms: 3

```

```

while (1)
{
    ADCSRA |= (1<<ADSC); //start conversion
    while((ADCSRA & (1<<ADIF)) == 0); //wait for conversion to complete

    ADCSRA |= (1<<ADIF);
    if (over_flow == 61) // when TCNT0 overflows 61
    {
        int a = ADCL;
        a = a | (ADCH<<8);
        a = (a/1024.0) * 5000/10;
        usart_send((a/100)+'0');
        a = a % 100;
        usart_send((a/10)+'0');
        a = a % 10;
        usart_send((a)+'0');
        usart_send('\r');
        over_flow = 0;
    }

    //delay_ms(1000);
}

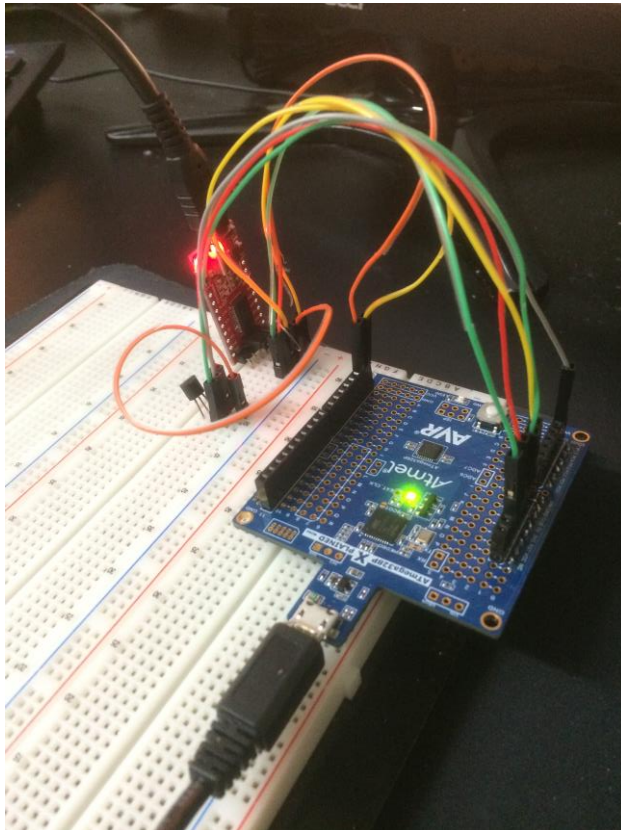
return 0;
}

// timer_0 overflow interrupt
// timer_0 overflow interrupt
ISR(TIMERO_OVF_vect)
{
    while (!(TIFR0 & 0X01) == 0);
    TCNT0 = 0X00; //resetting counter to zero
    TIFR0 = 0X01; // reset the overflow flag
    over_flow++; //increasing overflow counter
}

```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Photo below shows the set up



6. VIDEO LINKS OF EACH DEMO

DA3B

<https://youtu.be/4NKymgPBUC8>

7. GITHUB LINK OF THIS DA

https://github.com/chicosisco/da_sub.git

Student Academic Misconduct Policy

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

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

Francisco Mata Carlos