RALPH MAGO

CPE301 – SPRING 2016

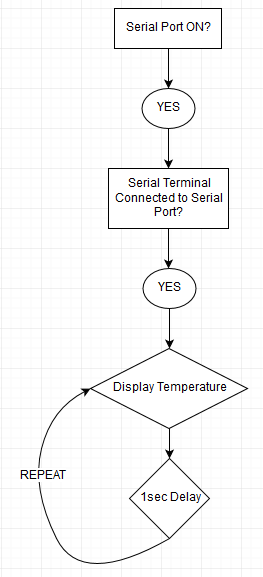
Design Assignment 4

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | FLOWCHART OF THE CODE | Y |  |
| 1. | COMPONENTS LIST | Y |  |
| 2. | INITIAL CODE OF TASK 1/A | Y |  |
| 3. | SCHEMATICS | Y |  |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT | Y |  |
| 5. | SCREENSHOT OF EACH DEMO | Y |  |
| 6. | VIDEO LINKS OF EACH DEMO | Y |  |
| 7. | GITHUB LINK OF THE DA | Y |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | FLOWCHART OF THE CODE |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 1. | COMPONENTS LIST |  |  |

|  |  |
| --- | --- |
| Atmel Xplained Mini (328P) | 1 |
| Sparkfun FTDI Basic Board | 1 |
| TI LM34 Temperature Sensor | 1 |
| PC running Atmel Studio Serial Terminal | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 1/A |  |  |

/\*

\* DA3T1.c

\*

\* Created: 3/24/2016 5:01:14 PM

\* Author: magor

\*/

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

#define BAUD 9600 //9600 Baud Rate (default)

#define MYUBRR (double)*F\_CPU*/16/BAUD - 1 //given baud rate calculation

#define TCNT1\_VALUE 65536 - ((double)*F\_CPU*/256) //calculation for Timer 1 Counter (1s Period)

#include <avr/io.h>

#include <avr/interrupt.h>

void init\_Timer1 (void)

{

TCNT1 = TCNT1\_VALUE; //overflow in 1 sec

TCCR1A = 0; //normal mode

TCCR1B = 4; //prescaler = 256

}

int main(void)

{

DDRC &= (0<<PORTC0); //PC0 = ADC0 for ADC input

UCSR0B = (1<<TXEN0); //enable TX

UCSR0C = ((1<<UCSZ01)|(1<<UCSZ00)); //set Asynchronous Mode

UBRR0L = MYUBRR; // Baud Rate 9600, 16MHZ

ADCSRA = 0x87;

//0b10000111, 7th bit set for ADEN. Bit 0-3 set to 111 for prescaler 128

ADMUX |= (1<<REFS0) | (1<<REFS1); //use internal VREF for LM34

sei(); //enable interrupt

init\_Timer1 (); //configure timer 1 to interrupt every second

TIMSK1 |= (1<<TOIE1); //enable Timer Overflow Interrupt

while(1);

{

// loop forever with interrupts

}

return 0;

}

void USART\_TX\_Out(char \*ch) //takes a string and sends it serially

{

while(\*ch != '\0') //send chars until NULL is found

{

while(!(UCSR0A & (1<<UDRE0))); //wait for UDRE0 to be 1

UDR0 = \*ch; //send char serially

ch++; //point to next char

}

}

ISR(TIMER1\_OVF\_vect) //timer1 overflow ISR

{

TCCR1B = 0; //stop timer 1

TIFR1 = 1; //clear overflow flag

int adc\_temp; //stores ADC temporarily

float adc\_temp\_Float; //float for calculations

int adc\_LeftSide; //left side of the value before decimal point

int adc\_RightSide; //right side of the value before decimal point

ADCSRA |= (1<<ADSC); //ADC start conversion

while((ADCSRA &(1<<ADIF)) == 0); //while conversion isn't finished

adc\_temp = ADC; //get ADC value (LM34 Vout)

adc\_temp\_Float = (float)adc\_temp \* (1.1 / 1024) / 0.01;

//AREF = 1.1V, 10 bit ADC = 1024, 10mV°F Scale Factor

adc\_LeftSide = (int)adc\_temp\_Float; //integer part before decimal point

adc\_temp\_Float = adc\_temp\_Float - adc\_LeftSide; //subtract left side

adc\_RightSide = (int)(adc\_temp\_Float \* 10); //decimal part before decimal point

char string\_temp[18] = "LM34 TEMPERATURE: ";

char TempInteger[27]; //stores temperature to be displayed

*sprintf*(TempInteger, "%d", adc\_LeftSide); //integer pAart to string

*strcat*(TempInteger, "."); //concatenate dot

char TempDecimal[6]; //stores decimal part

*sprintf*(TempDecimal, "%d°F\r\n", adc\_RightSide);

// %d displays integer value of decimal value

//output degrees Fahrenheit to the end of the

//transmission. \r\n combination of carriage return

//and line feed prevents scattered data in serial terminal

*strcat*(TempInteger, TempDecimal); //concatenate integer and decimal parts

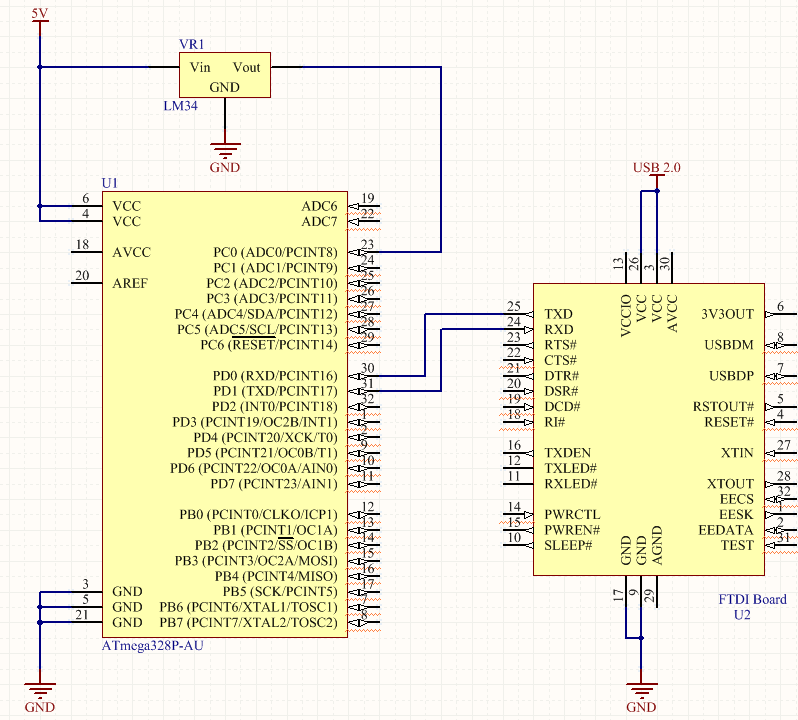
USART\_TX\_Out(string\_temp); //display finalized temperature value string on serial terminal

init\_Timer1 (); //configure timer 1 to interrupt every second

return;

}

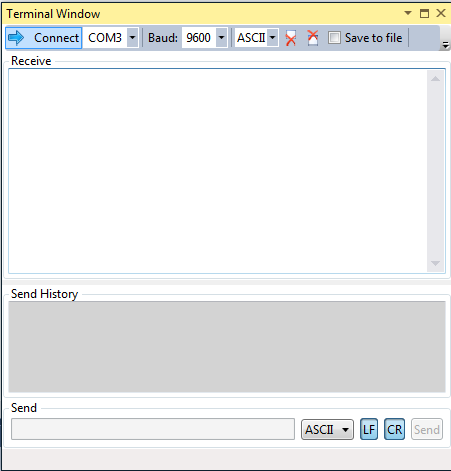
|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCHEMATIC |  |  |



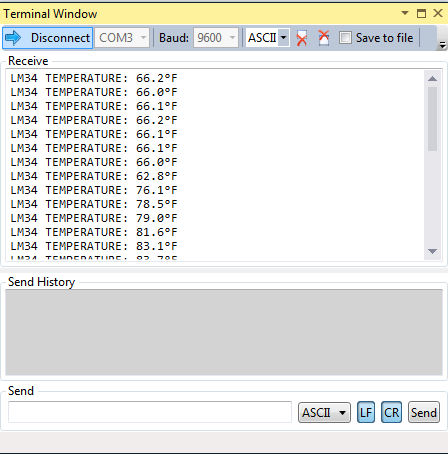
|  |  |  |  |
| --- | --- | --- | --- |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

Using Atmel Studio 7, I used the **Terminal** extension in order to access the serial terminal using a communication port. The FTDI basic was connected to my PC via USB 2.0.

**Initial Start**: Terminal Window is clear

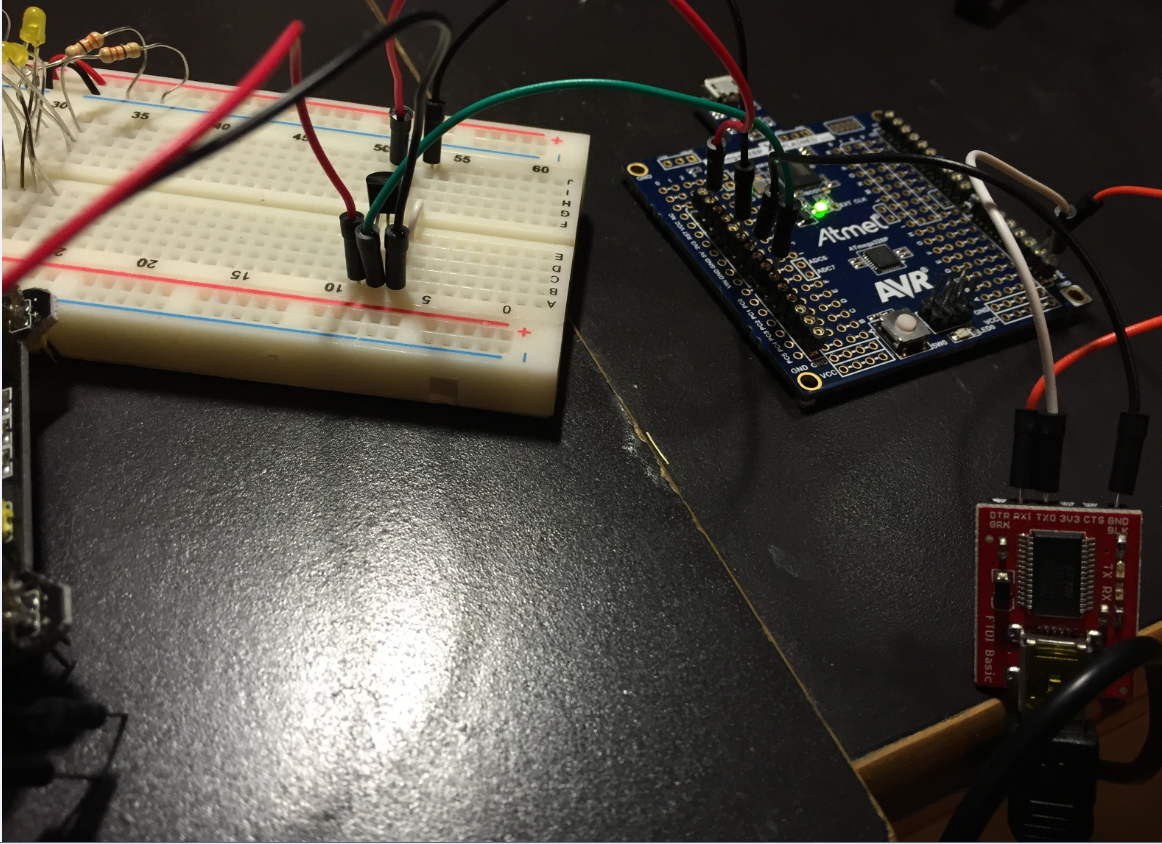


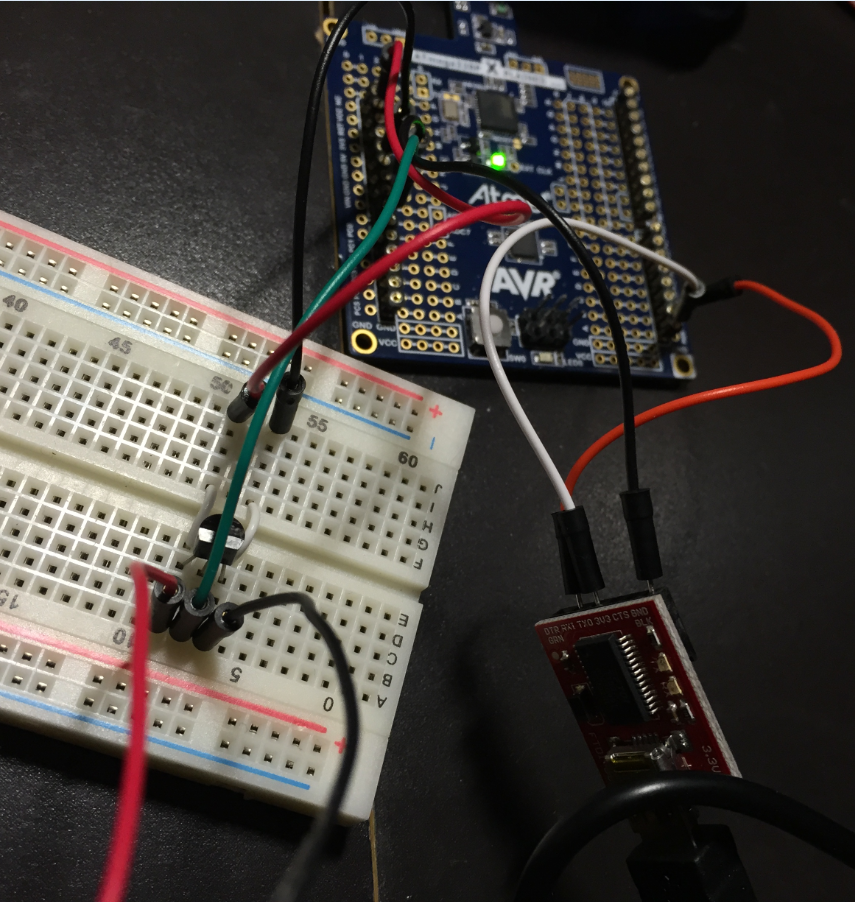
**Serial Port Connected**: FTDI transmitting LM34 Temperature converted with ADC.

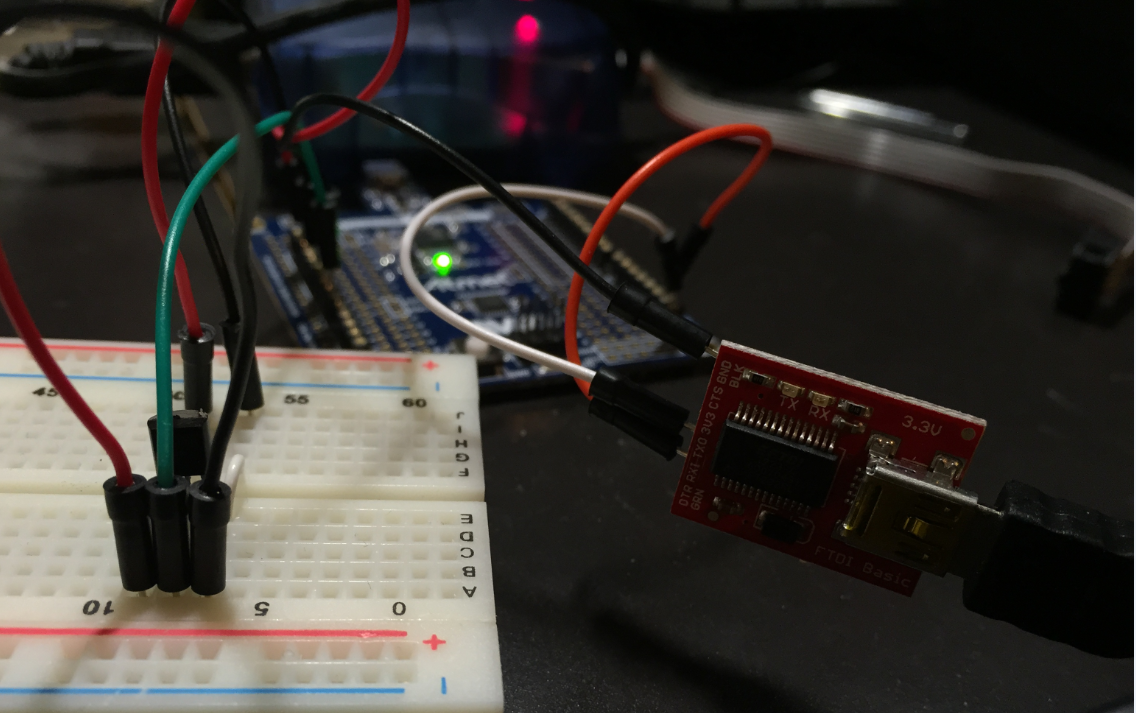


|  |  |  |  |
| --- | --- | --- | --- |
| 5. | SCREENSHOT OF EACH DEMO |  |  |

Breadboard, Atmel Xplained Mini 328P, and FTDI Basic connections:







|  |  |  |  |
| --- | --- | --- | --- |
| 6. | VIDEO LINK OF DEMO |  |  |
|  | https://www.youtube.com/watch?v=sbD2Pp3\_FiA |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | GITHUB LINK OF THE DA |  |  |
|  | https://github.com/magor1/embedded-design-VM/tree/master/DA3 |  |  |

**Student Academic Misconduct Policy**

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

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

RALPH MAGO