Michelle Mata

CPE301 – SPRING 2016

Design Assignment 3

**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. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 4. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 5. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

RS232-USB

Atmega 328P

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

; Write a C AVR program that will monitor the LMM24/25 connected to an Analog pin to display the temperature in F on the serial terminal every 1 sec.

#include <avr/io.h>

#include <stdint.h> // needed for uint8\_t

#include <avr/interrupt.h>

#define FOSC 16000000 // Clock Speed

#define BAUD 9600

#define MYUBRR FOSC/16/BAUD -1

volatile char ReceivedChar;

volatile uint8\_t ADCvalue; // Global variable, set to volatile if used with ISR

void TimDelay();

int main( void ){

/\*Set baud rate \*/

unsigned char str[15] =" Temperature: ";

unsigned char Length=15;

unsigned char i=0;

UBRR0H = (MYUBRR >> 8);

UBRR0L = MYUBRR;

UCSR0B |= (1 << RXEN0) | (1 << TXEN0); // Enable receiver and transmitter

UCSR0B |= (1 << RXCIE0); // Enable reciever interrupt

UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00)(1<<UMSEL00); // Set frame: 8data, 1 stp

DDRD =(1<<2); //PortD.2 output

DDRD =(1<<3); //PortD.3 output

ADMUX = 0; // use ADC0

ADMUX |= (1 << REFS0); // use AVcc as the reference

ADMUX |= (1 << ADLAR); // Right adjust for 8 bit resolution

ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // 128 prescale for 16Mhz

ADCSRA |= (1 << ADATE); // Set ADC Auto Trigger Enable

ADCSRB = 0; // 0 for free running mode

ADCSRA |= (1 << ADEN); // Enable the ADC

ADCSRA |= (1 << ADIE); // Enable Interrupts

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

sei(); // Thanks N, forgot this the first time =P

while(1)

{

; // Main loop

//receive data

while(!(UCSR0A & (1<<RXC0)));

PORTD=UDR0;

//send data

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

UDRE0=str[i++];

if(i>=Length)

i=0;

TimDelay();

}

}

ISR (USART\_RX\_vect){

ReceivedChar = UDRE0; // Read data from the RX buffer

UDRE0 = ReceivedChar; // Write the data to the TX buffer

}

ISR(ADC\_vect){

ADCvalue = ADCH; // only need to read the high value for 8 bit

}

void TimDelay(){

TCNTIH=0x7A; //Temp =0x7A

TCNTIL=0x11; //TCNTI=0x7A11

TCCR1A = 0x00; //WGM11:10 = 00

TCCR1B = 0x4; //WGM13:12=0, normal mode, CS=clk/256 prescaler

while((TIFR1&(0x1<<TOV1))==0); //wait for TOV1 to roll over

TCCR1B=0;

TIFR1=0x1<<TOV1; //Clear TOV1

}

——————————————//using FILE \*stream ———————————————————-

int USART0SendByte(char u8Data, FILE \*stream)

{

//wait while previous byte is completed

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

// Transmit data

UDRE0 = u8Data;

return 0;

}

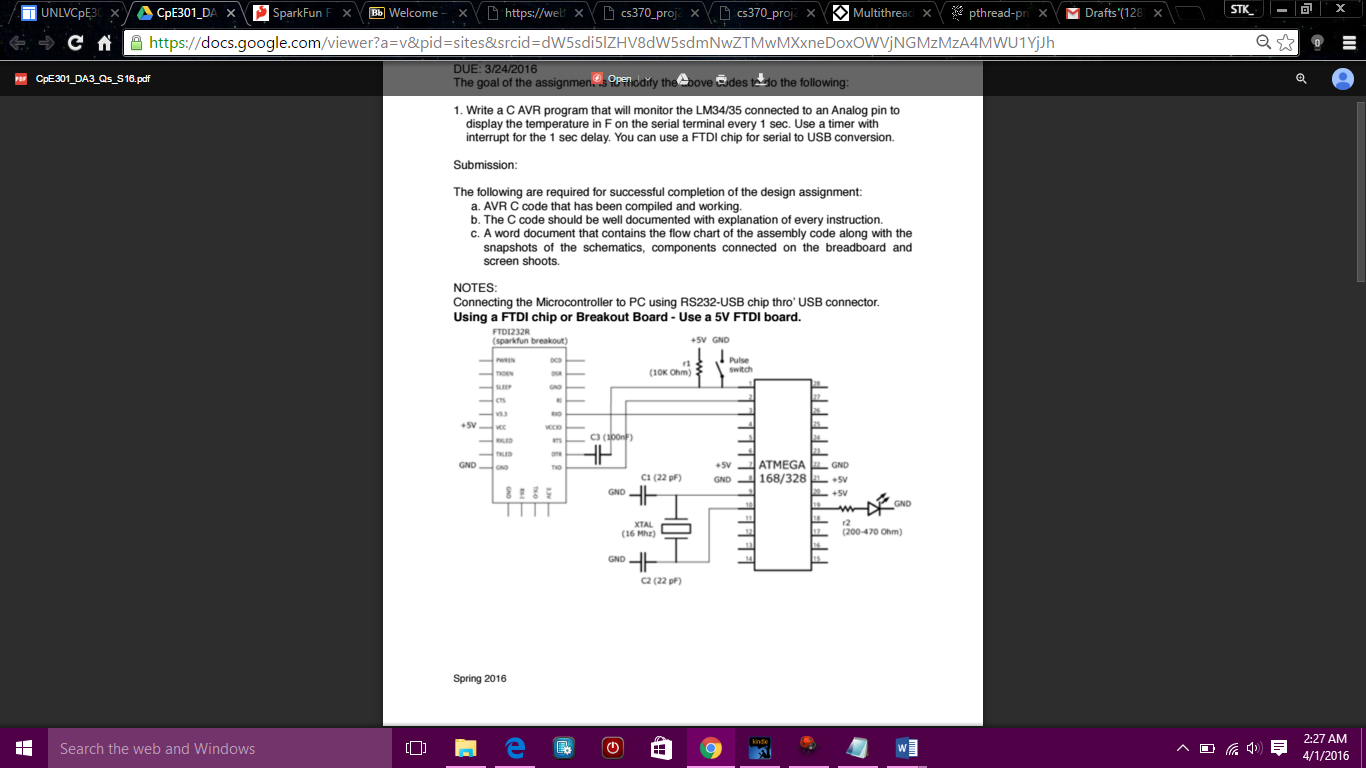
//set stream pointer

FILE usart0\_str = FDEV\_SETUP\_STREAM(USART0SendByte, NULL, \_FDEV\_SETUP\_WRITE);

printf("ADC ST\_ARRAY[%u] = %u\r\n", index, array[index]);

——————————————//end FILE \*stream ———————————————————-

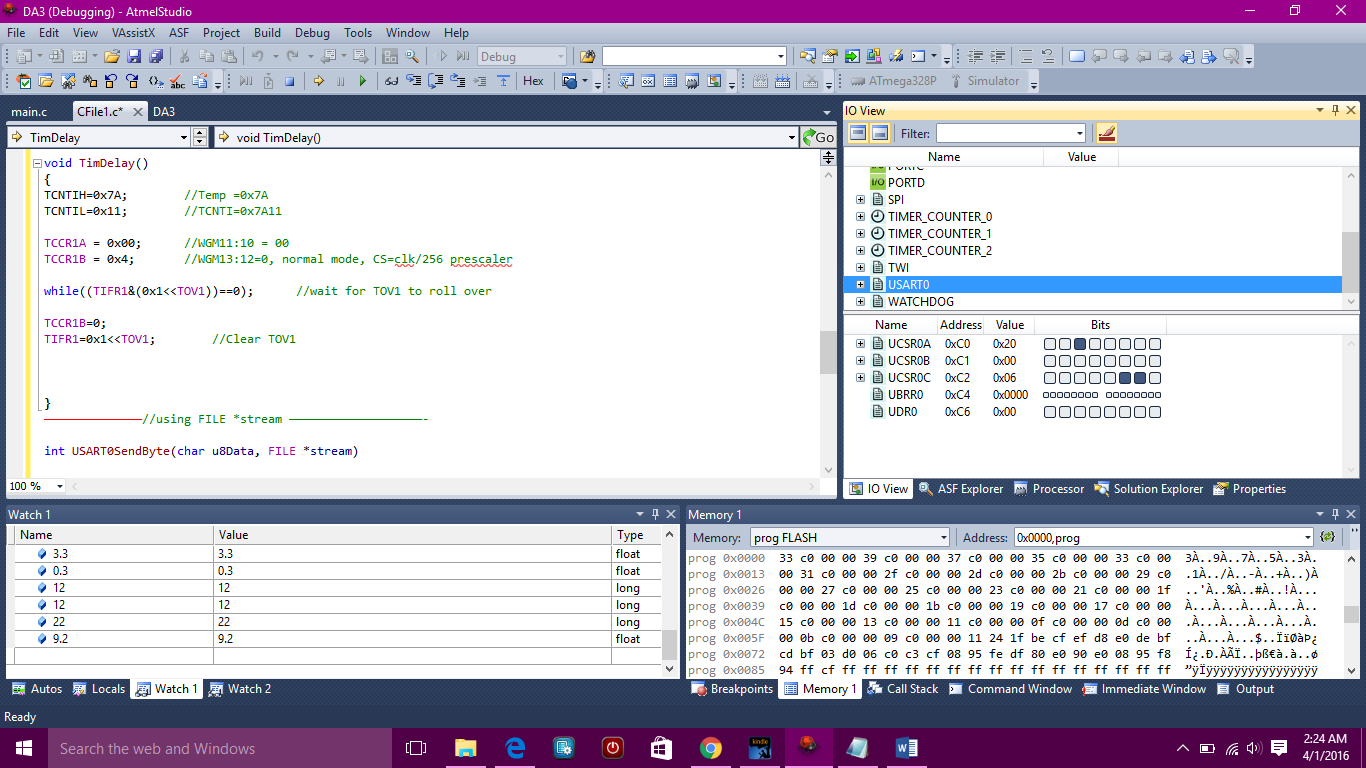
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

TASK 1/A:

Write a C AVR program that will monitor the LMM24/25 connected to an Analog pin to display the temperature in F on the serial terminal every 1 sec.



|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF EACH DEMO |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| http:// @youtube | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/matareye/MataReyesCPE301s16/tree/master | | | |

**Student Academic Misconduct Policy**

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

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

Michelle Mata