Michelle Mata

CPE301 – SPRING 2016

Design Assignment 6

**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 |  |  |

ATmega 328P

Nokia 5110 GLCD display module

LM34

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

; Write a C AVR program that will monitor the LM34/LM35 connected to an Analog pin to

display the temperature in C or F on the Nokia 5110 GLCD display Module.

//References: https://faculty.unlv.edu/eelabs/docs/labs/cpe301L/cpe301L\_09\_experiment\_9.pdf

// https://docs.google.com/viewer?a=v&pid=sites&srcid=dW5sdi5lZHV8dW5sdmNwZTMwMXxneDoxOWVjNGMzMzA4MWU1YjJh

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#define LCD\_DPRT PORTD //LCD DATA PORT

#define LCD\_DDDR DDRD //LCD DATA DDR

#define LCD\_DPIN PIND //LCD DATA PIN

#define LCD\_CPRT PORTB //LCD COMMANDS PORT

#define LCD\_CDDR DDRB //LCD COMMANDS DDR

#define LCD\_CPIN PINB //LCD COMMANDS PIN

#define LCD\_RS 0 //LCD RS

#define LCD\_RW 1 //LCD RW

#define LCD\_EN 2 //LCD EN

volatile uint8\_t ADCvalue;

volatile uint8\_t temp;

#define F\_CPU 8000000UL // XTAL = 8MHz

int main(void)

{

LCD\_DDDR=0xFF;

LCD\_CDDR=0xFF;

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();

temp=ADCvalue;

temp=temp\*(10/1024);

lcd\_init();

lcd\_print(temp);

while(1){

}

}

ISR(ADC\_vect){

ADCvalue = ADC;

}

void lcd\_init()

{

LCD\_DDDR = 0xFF;

LCD\_CDDR = 0xFF;

LCD\_CPRT &=~(1<<LCD\_EN); //LCD\_EN = 0

\_delay\_ms(2000); //wait for init

}

void lcdData(unsigned char data)

{

LCD\_DPRT = data; //send data to data port

LCD\_CPRT |= (1<<LCD\_RS); //RS = 1 for data

LCD\_CPRT &= ~(1<<LCD\_RW); //RW = 0 for write

LCD\_CPRT |= (1<<LCD\_EN); //EN = 1 for H-to-L pulse

\_delay\_ms(1); //wait to make enable wide

LCD\_CPRT &= ~(1<<LCD\_EN); //EN = 0 for H-to\_L pulse

\_delay\_ms(100); //wait to make enable wide

}

void lcd\_print(char\* str){

unsigned char i=0;

while(str[i]!=0){

lcdData(str[i]);

i++;

}

}

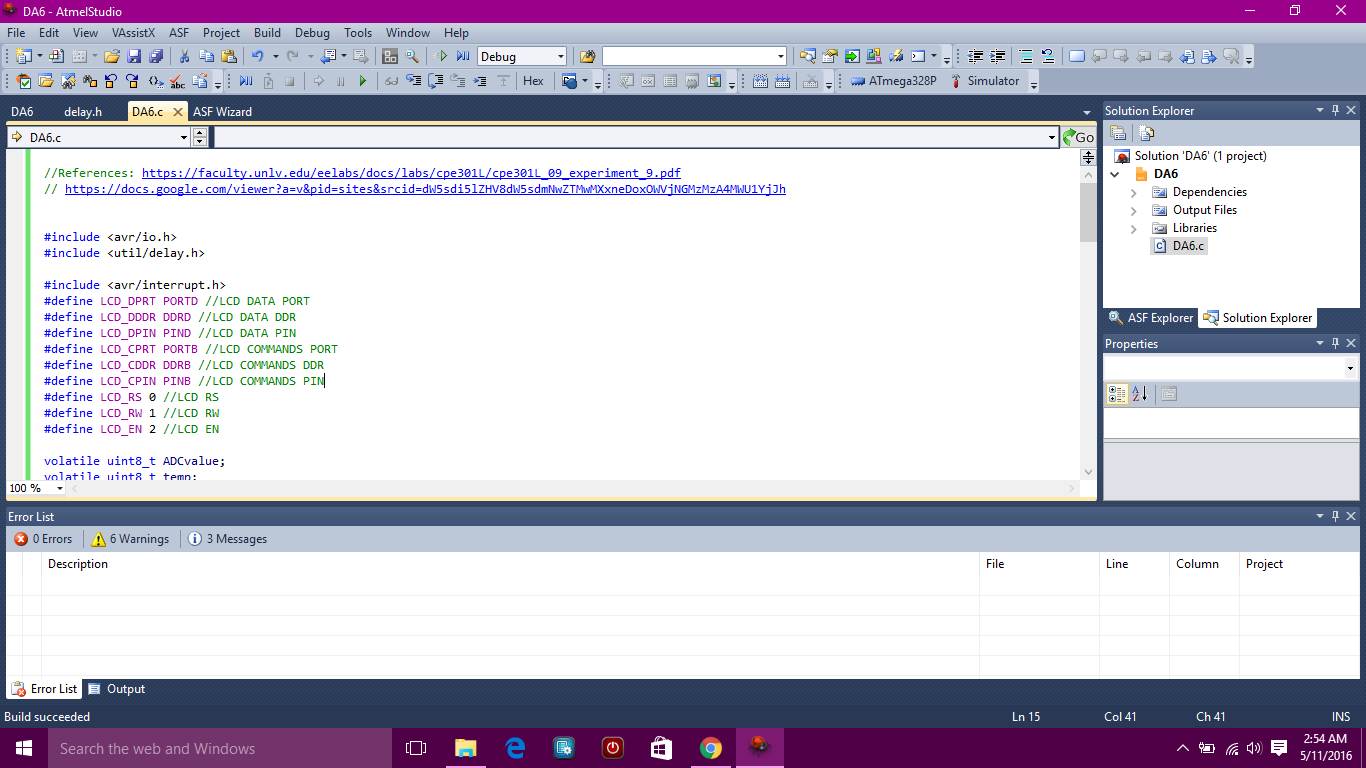
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |

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

TASK 1:

Write a C AVR program that will monitor the LM34/LM35 connected to an Analog pin to

display the temperature in C or F on the Nokia 5110 GLCD display Module.



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

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| http:// @youtube | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/matareye/MataReyesCPE301s16/blob/master/DA6 | | | |

**Student Academic Misconduct Policy**

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

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

Michelle Mata