Desi Battle

CPE301 – SPRING 2016

Design Assignment 6

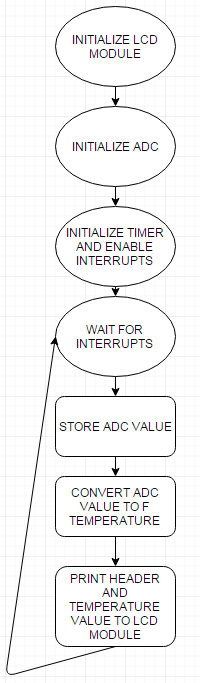
**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

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 FLOWCHART |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | SCHEMATICS |  |  |
| 4. | SCREENSHOT OF EACH DEMO |  |  |
| 5. | VIDEO LINKS OF EACH DEMO |  |  |
| 6. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND FlowChart |  |  |

* Atmega328P
* LM34 temperature sensor
* LCM S01601DTR LCD module



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

/\*

\* DA6.c

\*

\* Created: 4/28/2016 6:15:01 PM

\* Author: battled

\*/

#define F\_CPU 8000000UL //8 MHz xtal crystal

#include <avr/io.h> //

#include <util/delay.h> //delay header

#include <avr/interrupt.h>

#include <string.h> //for string concatenation

#include <stdio.h> //for sprintf

#define F\_CPU 8000000UL

#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

void lcdCommanda (unsigned char cmnd)

{

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

LCD\_CPRT &= ~(1<<LCD\_RS); //RS = 0 for command

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

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

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

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

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

}

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\_us(1); //wait to make enable wide

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

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

}

void lcd\_gotoxy(unsigned char x, unsigned char y)

{

unsigned char firstCharAdr[] = {0x80, 0xC0, 0x94, 0xD4};

lcdCommanda(firstCharAdr[y-1] + x -1);

\_delay\_us(100);

}

void lcd\_print(char \* str)

{

unsigned char i = 0;

while (str[i]!=0)

{

lcdData(str[i]);

i++;

}

}

void lcd\_init()

{

LCD\_DDDR = 0xFF;

LCD\_CDDR = 0xFF;

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

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

lcdCommanda(0x38); //inti. LCD 2 line, 5x7

lcdCommanda(0x0E); //display on, cursor on

lcdCommanda(0x01); //clear LCD

\_delay\_us(2000); //wait

lcdCommanda(0x06); //shift cursor right

}

int main(void)

{

DDRC &= ~(1<<PORTC0); //PC0 is analog input

lcd\_init();

ADCSRA = ((1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0));// ADC prescaler 128 ADEN

ADMUX = ((1<<REFS1)|(1 << REFS0)); // select internal 1.1 V Ref w/ ext cap at AREF pin and ADC0 (default)

sei(); //enable interrupts

//configure timer 1 to interrupt every second

TCNT1 = 65536 - ((double)F\_CPU/256); // set timer to overflow in 1 sec

TCCR1A = 0; //normal mode

TCCR1B = 4; //prescaler = 256

TIMSK1 |= (1<<TOIE1); //enable interrupt on overflow timer 1

while(1); //wait for interrupts

return 0;

}

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\_tempf; //float for calculations

int adc\_tempi; //integer part

int adc\_tempd; //decimal part

//read ADC

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

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

adc\_temp = ADC; //save ADC value

adc\_tempf = (float)adc\_temp \* (1.1 / 1024) / 0.01; //(ADC\*res/.01) (lm34 sf = 10mv/degF)

adc\_tempi = (int)adc\_tempf; //integer part

adc\_tempf = adc\_tempf - adc\_tempi; //remove integer part of float temp

adc\_tempd = (int)(adc\_tempf \* 100); //store 2 decimal bits as float

//current temp: xxx.xx°F\r\n == 24 chars

char TmpTemp[6] = "";

//sprintf here is used to "print" to TempInt string and then dot is added at end

sprintf(TmpTemp, "%d.%d F", adc\_tempi,adc\_tempd);

//output degrees Fahrenheit

lcd\_gotoxy(1,1);

lcd\_print("Temp is");

lcd\_gotoxy(1,2);

lcd\_print(TmpTemp);

//reset time 1 for interrupt

TCNT1 = 65536 - ((double)F\_CPU/256); //overflow in 1 sec

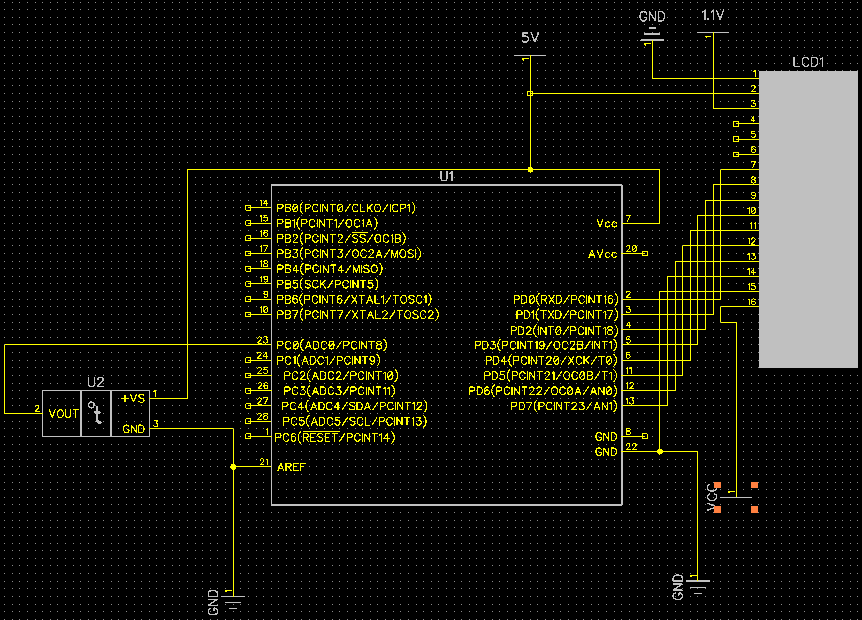
TCCR1A = 0; //normal mode

TCCR1B = 4; //prescaler = 256

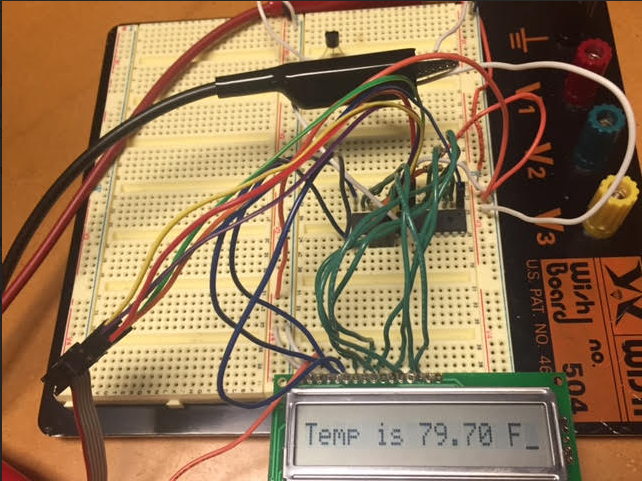
return;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | SCHEMATICS |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCREENSHOT OF EACH DEMO |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 4. | VIDEO LINKS OF EACH DEMO |  |  |
| https://www.youtube.com/watch?v=68ctYQPCqL4 | | | |
| 5. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/battled/DA0.git | | | |

**Student Academic Misconduct Policy**

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

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

Desi Battle