Derek Nhan

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.

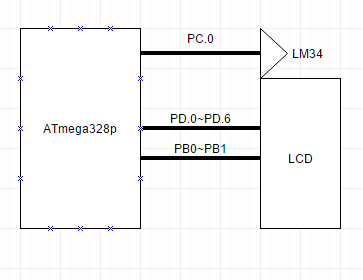
|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | SCHEMATICS |  |  |
| 3. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 4. | SCREENSHOT OF EACH DEMO |  |  |
| 5. | FLOWCHART |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
| 8. |  |  |  |
| 9. |  |  |  |
| 10. |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

LM34 Temperature Sensor

ATmega328p Xplained board

16x1 LCD Screen



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

#define *F\_CPU* 8000000UL

#include <avr/io.h> //standard AVR header

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

#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 unsigned int ADCvalue; //adc value buffer

void init\_328(void)

//subroutine to initialize all counter/adc used in the application

{

ADMUX|=(1<<REFS0); //use Vcc as ref (0.1uF cap attached)

ADCSRA|=(1<<ADSC)|(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE);

//start conversion; enable adc; enable interrupts; select 128 prescaler;

//auto trigger enable; ADC0 used

ADCSRB=0;

//adc in free running mode

}

ISR (ADC\_vect)

{

char \*temperature="00"; //temperature string array

char temp; //temp place holder

ADCvalue=ADC; //reads all 16-bits of ADCH:ADCL

ADCvalue=(ADCvalue)\*(500.0/1024.0); //convert the read ADCvalue to temperature

//500.0=>(Vref \* 100)=>(5V \* 100)

//divide by 1024, the max for the ADC values (0-1024)

temp=(ADCvalue/10); //divide by 10 to get the 10's place digit

temperature[0]=temp+48; //add ascii '0' to display on LCD

temp=(ADCvalue%10); //modulo by 10 to get 1's place digit

temperature[1]=temp+ 48; //add ascii '0' to get display on LCD

lcd\_gotoxy(5,2); //move to cell 5 on second half of lcd

lcd\_print(temperature); //print the converted temperature string array

*\_delay\_ms*(2000); //wait 2 seconds before retreiving data again

}

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\_ms*(150); //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 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*(150); //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\_init()

{

LCD\_DDDR = 0xFF;

LCD\_CDDR = 0xFF;

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

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

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

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

lcdCommanda(0x01); //clear LCD

*\_delay\_ms*(2000); //wait

lcdCommanda(0x06); //shift cursor right

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

//choose which position on the LED to display message

//our module is 16x1, this y=>x, but on the second half

{

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

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

*\_delay\_ms*(100);

}

void lcd\_print(char \* str)

//prints the string array, one cell at a time (ascii)

{

unsigned char i = 0;

while (str[i]!=0)

{

lcdData(str[i]);

i++;

}

}

int main(void)

{

char \*degree;

lcd\_init(); //initialize the lcd module

lcd\_gotoxy(1,1); //display first 8 characters of "temperature"

lcd\_print("Temperat");

lcd\_gotoxy(1,2); //display last 4 characters of "temperature:"

lcd\_print("ure:");

lcd\_gotoxy(7,2); //display degress symbol 2nd half, second to last cell

degree[0]=223; //set to ASCII degree symbol

lcd\_print(degree);

lcd\_gotoxy(8,2); //display F for farenheit

lcd\_print("F");

init\_328(); //initialize the adc

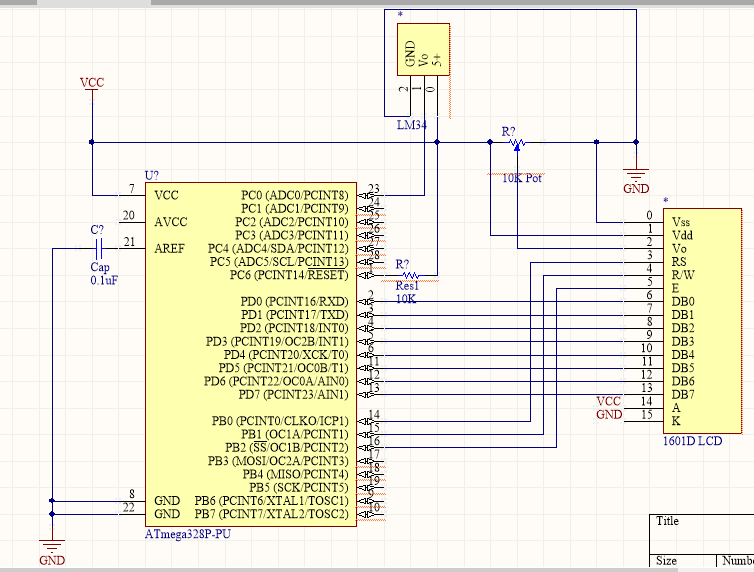
sei(); //enable interrupts

while(1);

return 0;

}

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



|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | SCREENSHOT OF EACH DEMO |  |  |

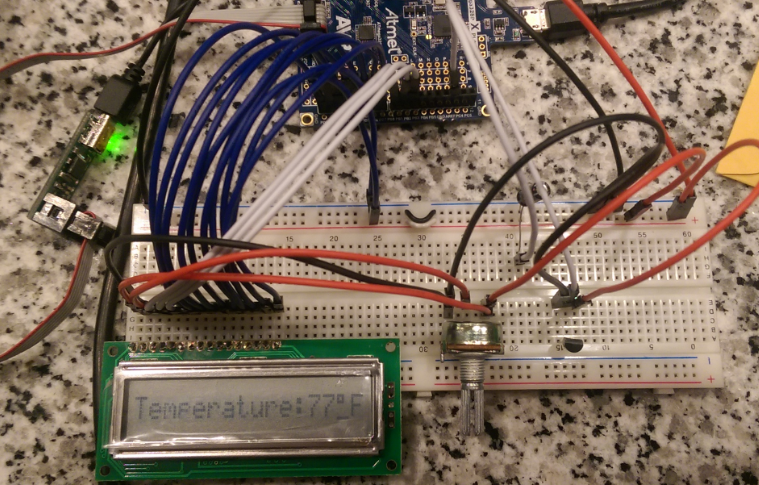
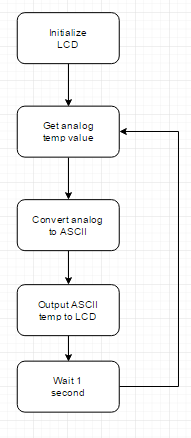


Figure : Temperature displayed on LCD via LM34 (see video for temp change)

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | FLOWCHART |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| https://youtu.be/rCu6Hb5xWvY | | | |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/nhand2/CPE301S16 | | | |

**Student Academic Misconduct Policy**

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

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

Derek Nhan