Martin Jaime

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 |  |  |

* Atmega328P
* Pololu USB AVR Programmer
* Breadboard
* Nokia 5110 GLCD
* Atmel Studio 7
* PC

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

#define *F\_CPU* 8000000UL // Set clock frequency

#include <stdio.h>

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include "nokia5110.h" // include LCD API

void ADC\_init(); // Function to initialize ADC

int main(void)

{

nokia\_lcd\_init(); // Initialize LCD

ADC\_init(); // Initialize ADC

sei(); // enable interrupts.

while(1); // Infinite loop waiting for interrupts.

}

void ADC\_init()

{

DDRC &= ~(0<<DDC0); // Clear bit 0 of DDRC

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

}

ISR(ADC\_vect)

{

*uint16\_t* ADC\_value = ADC; // read ADC conversion

*uint32\_t* temperature\_C, temperature\_F;

char \*header = "Temp(F) LM35:";

char buffer[10];

// Convert ADC value to temperature in Celcius

temperature\_C = (*int32\_t*)(ADC\_value \* 5 \* 100 / 1024.0);

// Convert Celcius to Farenheit

temperature\_F = temperature\_C \* 1.8 + 32;

if(temperature\_F < 120)

// There is noise at output that jumps to a very large value.

// This condition is to avoid that.

{

*sprintf*(buffer, "%u", temperature\_F); // write value to buffer as string

nokia\_lcd\_clear(); // clear LCD

nokia\_lcd\_set\_cursor(0, 0); // reset cursor

nokia\_lcd\_write\_string(header,1); // write message header

nokia\_lcd\_set\_cursor(0, 10); // move cursor down.

nokia\_lcd\_write\_string(buffer, 3); // Write buffer.

nokia\_lcd\_write\_string(" o", 1); // print letter 'o' as degrees symbol.

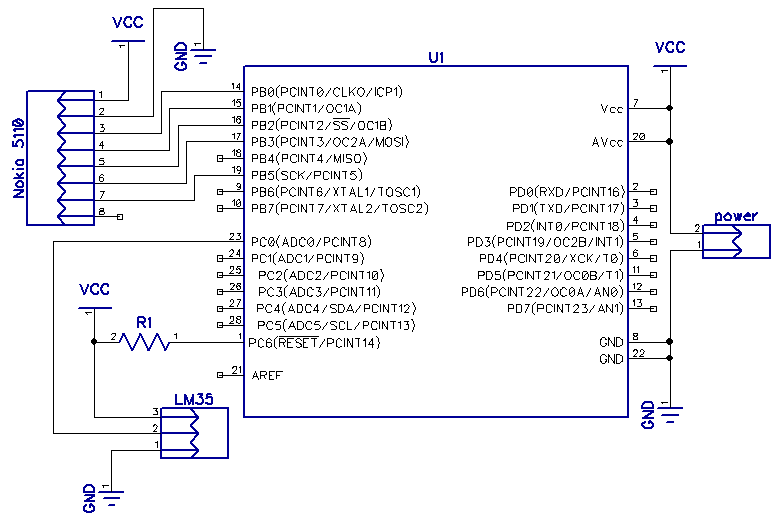
nokia\_lcd\_render(); // render the message to LCD.

}

*\_delay\_ms*(1000); // delay for 1 second.

}

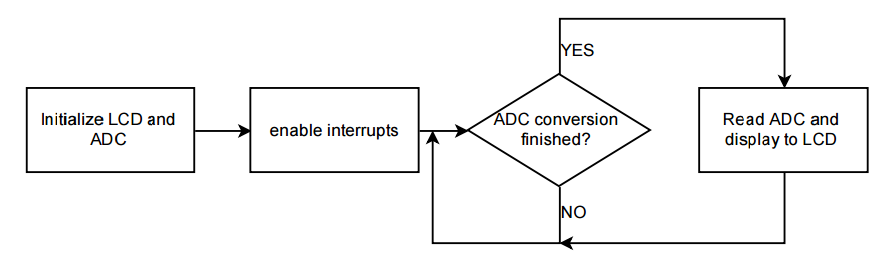
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |



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

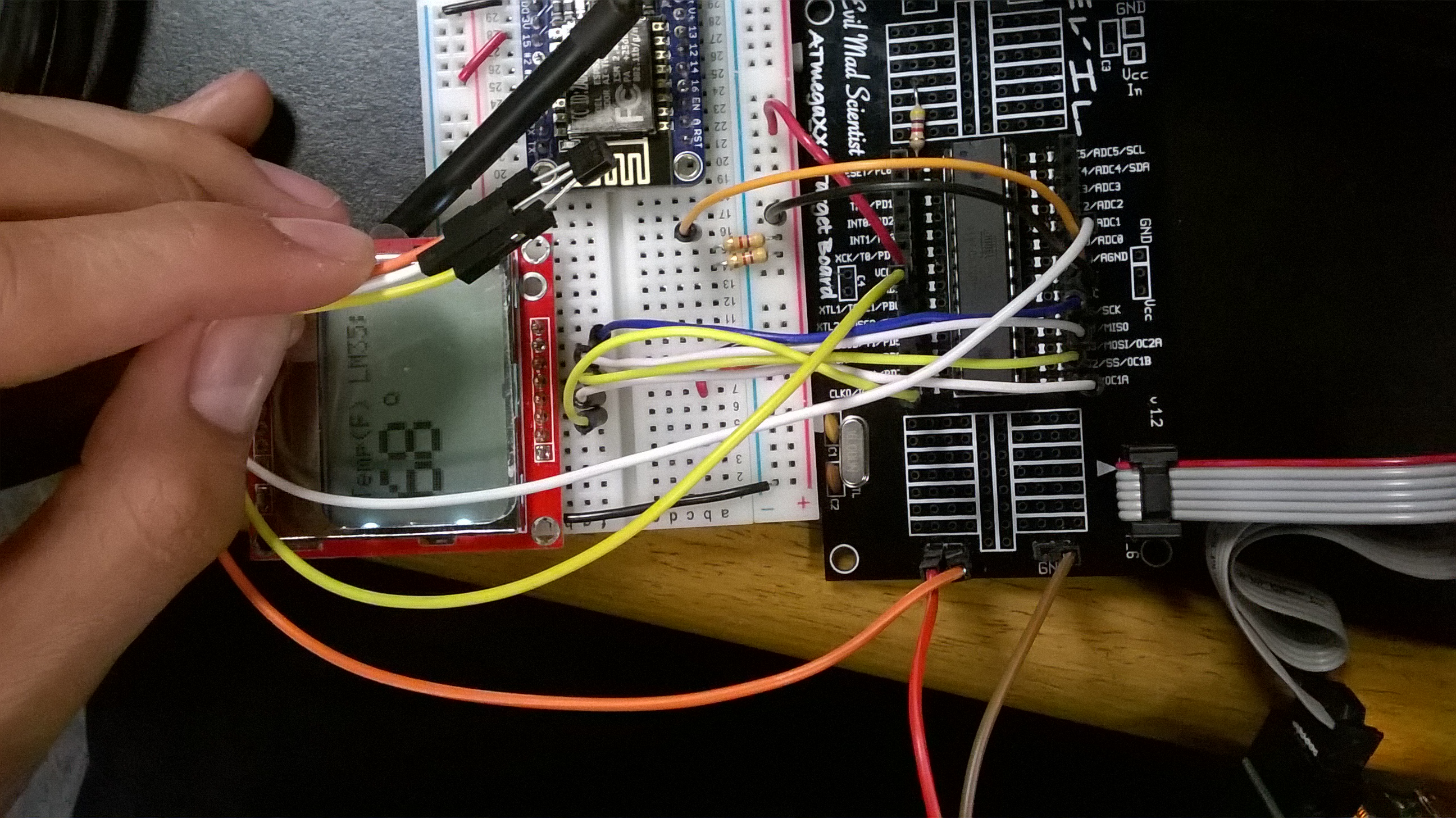
TASK 1:

Flow chart of code



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

TASK 1:



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://drive.google.com/folderview?id=0B4ItVBjMqlnyaVUtQjRTNkRWUjg&usp=sharing | | | |
| 10. | GITHUB LINK OF THE DA |  |  |
| https://github.com/martinjaime/CpE301\_Assignments2016S.git | | | |

**Student Academic Misconduct Policy**

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

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

Martin Jaime