Andrew Wells

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 |  |  |

Xplained Mini: Atmega328P Micro Controller

Potentiometer

Nokia 5110 GLCD Screen

LM-34 Temperature Sensor

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

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

#include "nokia5110.h"

void ADCstart(); // Function to initialize the ADC

volatile unsigned int ADCvalue; // Variable to save the temperature value

volatile unsigned int X; // variable to keep track of how many values have been read

float Ratio; // variable for calculating the temperature

char outs[8]; // output buffer

int main(void)

{

int Y = 0; // Change Y

Ratio = 500.0/1024.0; // Ratio to calculate the actual temperature

ADCstart(); // initialize the ADC port

*nokia\_lcd\_init*();// Initiaize LCD

while(1)

{

X = 4; // number of values taken before transmission

ADCvalue = 0; // reset temperature value

while(X) // stay here until 4 reading are taken

{

ADCSRA |= (1<<ADSC);// write one to ADSC bit to start conversion

while(ADCSRA & (1<<ADSC))// wait until conversion is complete

{

}

ADCvalue += (Ratio\*ADC);// save temperature

X--;

*\_delay\_ms*(250); //Delay to give time between readings

}

ADCvalue = ADCvalue/4;// average of values received

*snprintf*(outs, sizeof(outs),"%3d C", ADCvalue); // convert value to a string

*nokia\_lcd\_set\_cursor*(0, Y);

*nokia\_lcd\_write\_string*(outs,1); // Send Temperature to LCD

*nokia\_lcd\_render*();// Display Temperature on LCD

if (Y < 25)

Y+=8;

else

{

Y = 0;

*nokia\_lcd\_clear*(); // Clear screen

}

for (int z = 0; z< 1000;z++)

*\_delay\_ms*(4);

}

}

void ADCstart()

{

ADMUX = (0<<REFS1)| // Reference Selection Bits

(1<<REFS0)| //use AVcc as reference

(0<<ADLAR)| //ADC Left adjust Result

(0<<MUX2)| // Analog Channel Selection Bits

(1<<MUX1)| // ADC2 (PC2, PIN25)

(0<<MUX0);

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

(0<<ADSC)| //ADC Start Conversion

(0<<ADIF)| //ADC Auto Trigger Enable

(0<<ADIE)| //ADC Interrupt Flag

(0<<ADATE)| //ADC Interrupt Enable

(1<<ADPS2)| //ADC Prescaler Bits

(0<<ADPS1)|

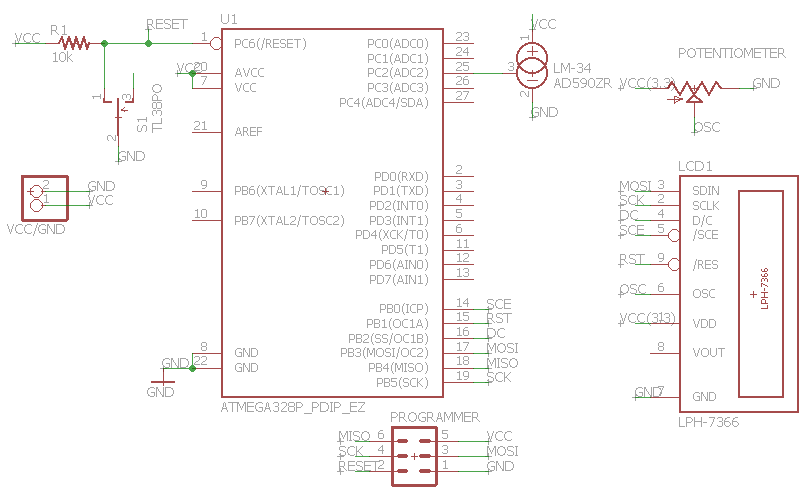
(1<<ADPS0);

return;

}

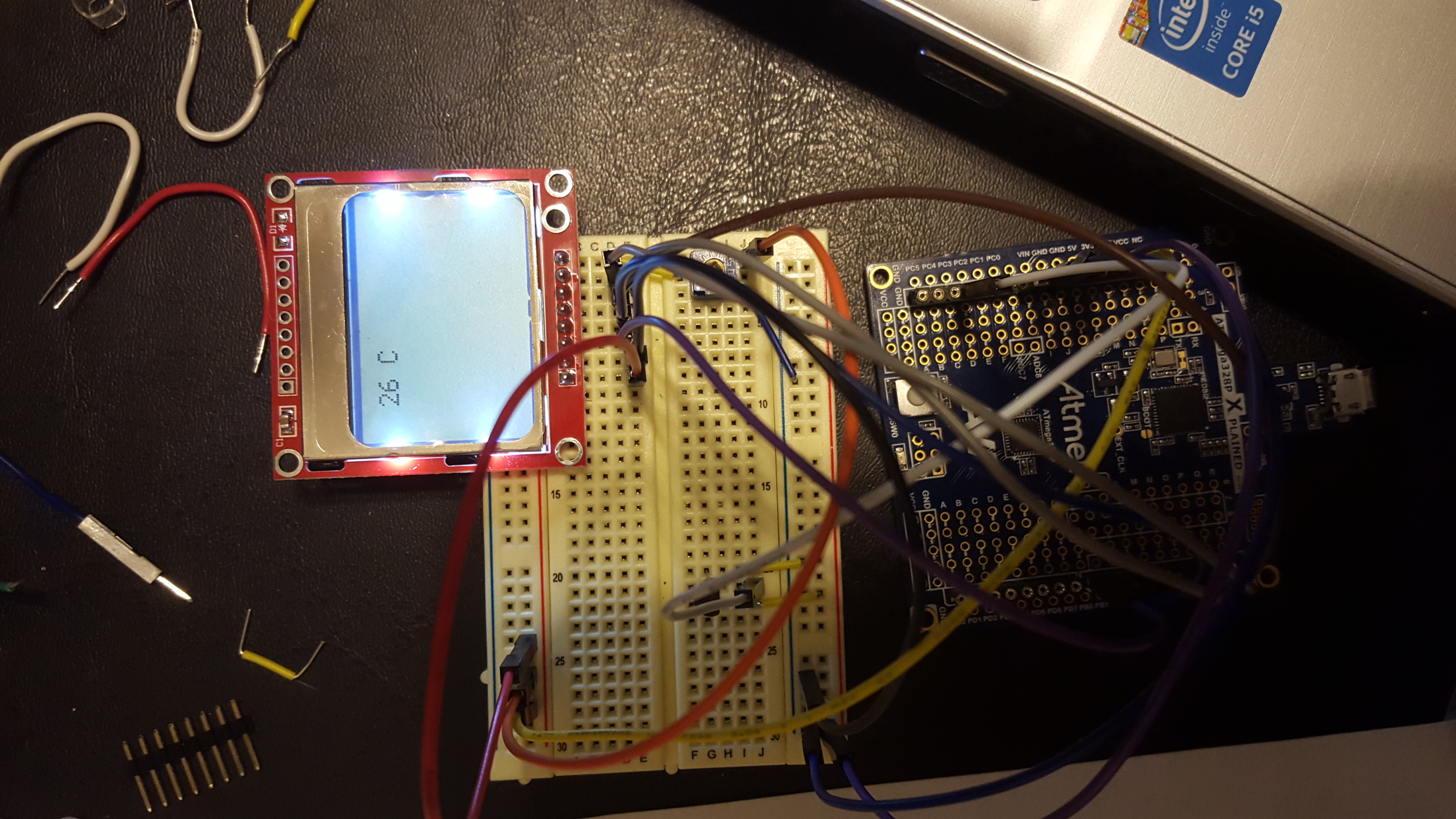
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |

TASK 1:



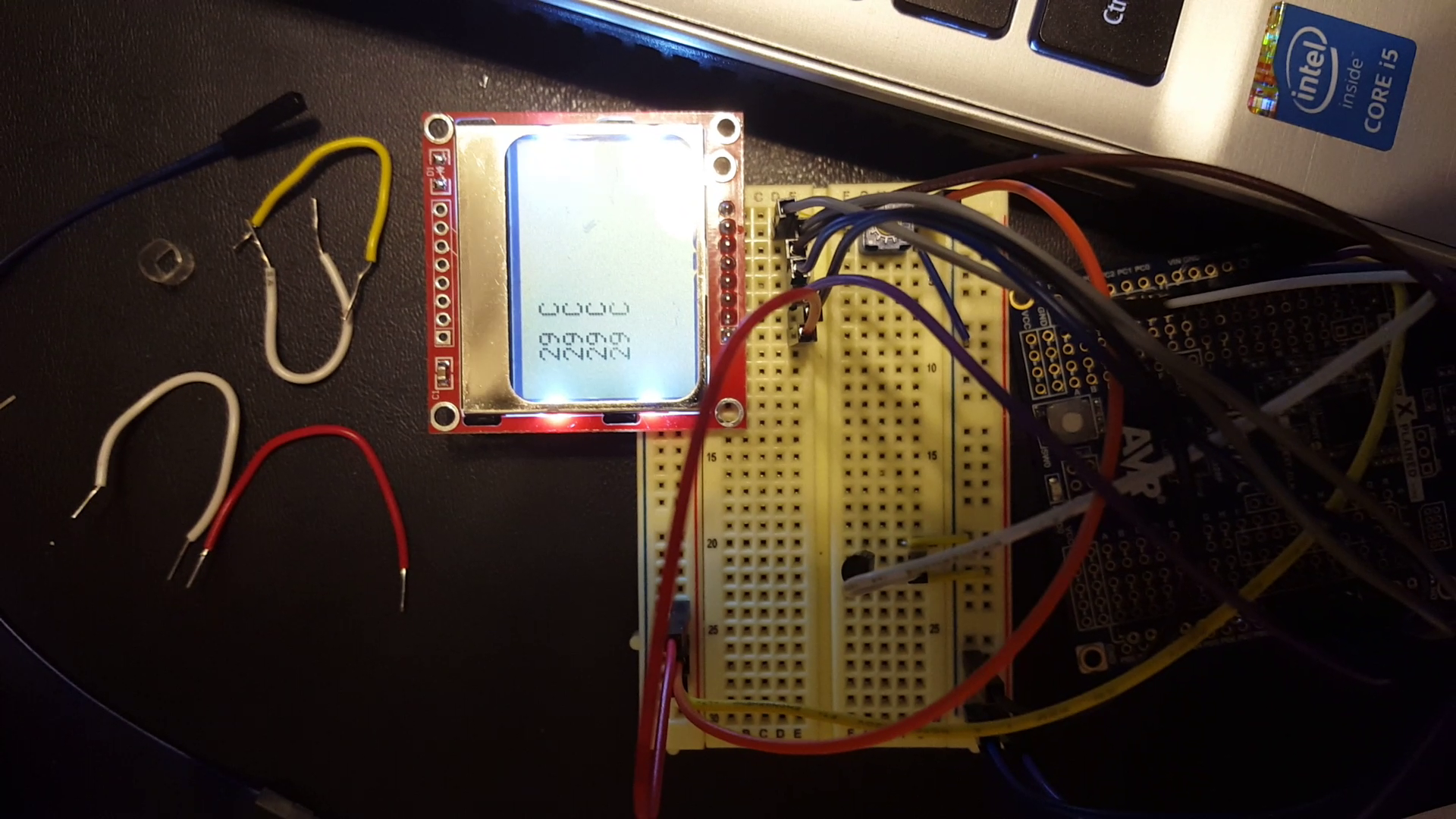
|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

TASK 1:



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

TASK 1:



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://youtu.be/OEYvZ1JDEaA | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/Wellsa15/wellsa\_unlv | | | |

**Student Academic Misconduct Policy**

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

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

Andrew Wells