Derek Nhan

CPE301 – SPRING 2016

Design Assignment 3

**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. | VIDEO LINKS OF EACH DEMO |  |  |
| 6. | GOOGLECODE LINK OF THE DA |  |  |
| 7. |  |  |  |
| 8. |  |  |  |
| 9. |  |  |  |
| 10. |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

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

// This code waits for a character and transmits the character back (with interrupts)

#include <avr/io.h>

#include <stdint.h> // needed for uint8\_t

#include <avr/interrupt.h>

#include <stdio.h>

#define FOSC 8000000 // Clock Speed

#define BAUD 9600

#define MYUBRR FOSC/16/BAUD -1

ISR(ADC\_vect)

{

while(!(UCSR0A&(1<<UDRE0)));

UDR0 = ADCH; // only need to read the high value for 8 bit

}

void wait ()

{

TCNT1=57724; //sets counter to 57724, which takes 1 s to overflow

TCCR1A=0x00; //normal more operation

TCCR1B=0x05; //prescaler of 1024

while((TIFR1&0x01)==0); //loops until TOV1 is set

TCCR1B=0x00; //stops the timer

TIFR1|=(1<<TOV1); //clear TOV1 flag

}

int main( void )

{

/\*Set baud rate \*/

UBRR0H = ((MYUBRR)>>8);

UBRR0L = MYUBRR;

UCSR0B |= (1 << TXEN0); // Enable transmitter

UCSR0B |= (1 << TXCIE0); // Enable transmitter interrupt

UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // Set frame: 8data, 1 stp

ADMUX |= 0x08; //Set to use internal temp sensor, Ref of 1.1V, left adjust

ADMUX |= (1<<ADLAR);

ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // 128 prescale for 16Mhz

ADCSRA |= (1 << ADATE); // Set ADC Auto Trigger Enable

ADCSRB = 0x06; // Start ADC when overflow flag is set

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

ADCSRA |= (1 << ADIE); // Enable Interrupts

sei();

while(1)

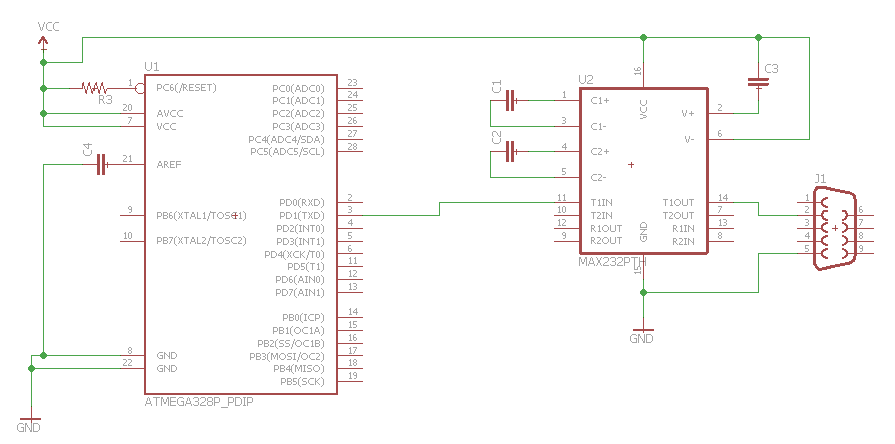
{

wait(); // Main loop

}

}

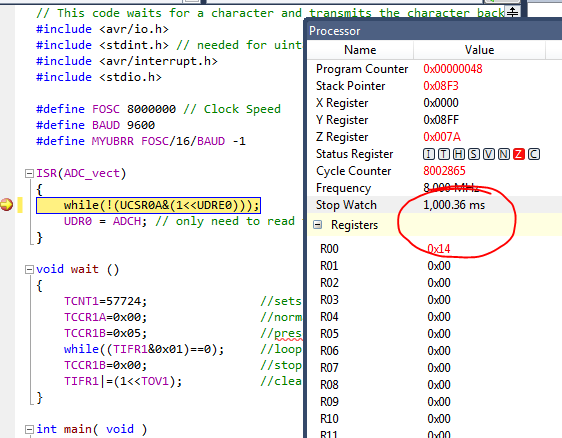
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |



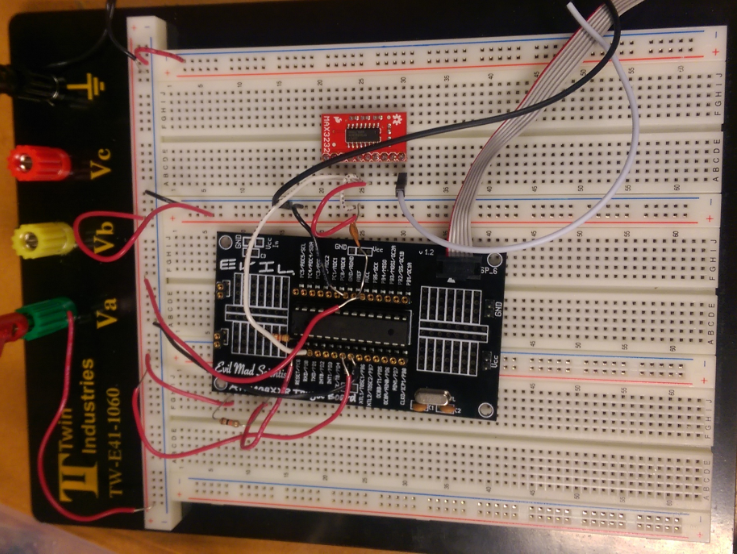
|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

TASK 1/A:

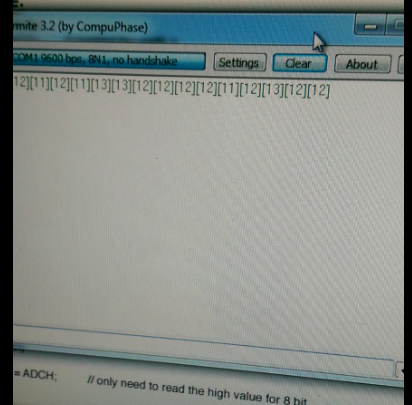
Due to the use of auto trigger at timer1 overflow, the ADC start interrupt should occur every 1 second. This starts the ADC8 to use the internal temperature sensor.



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



Screenshot of breadboard with the ATmega wired to the serial RS232 cable.



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://youtu.be/TYwWEmlX12A | | | |
| 10. | 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