Yannick Kengne Tatcha

CPE301 – SPRING 2018

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.	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		

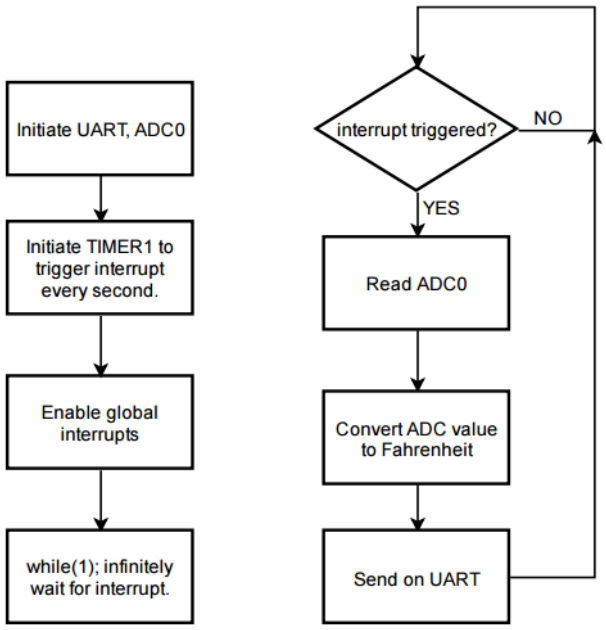
0.	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		

- Atmega328P
- LM34
- Pololu USB AVR programmer for serial communication
- AVR Studio 7

1.	INITIAL CODE OF TASK 1/A		

Task 1: Write a C AVR program that will monitor the LM34/35 connected to an Analog pin to display the temperature in F on the serial terminal every 1 sec. Use a timer with interrupt for the 1 sec delay. Use a FTDI chip for serial to USB conversion.

Flow chart:

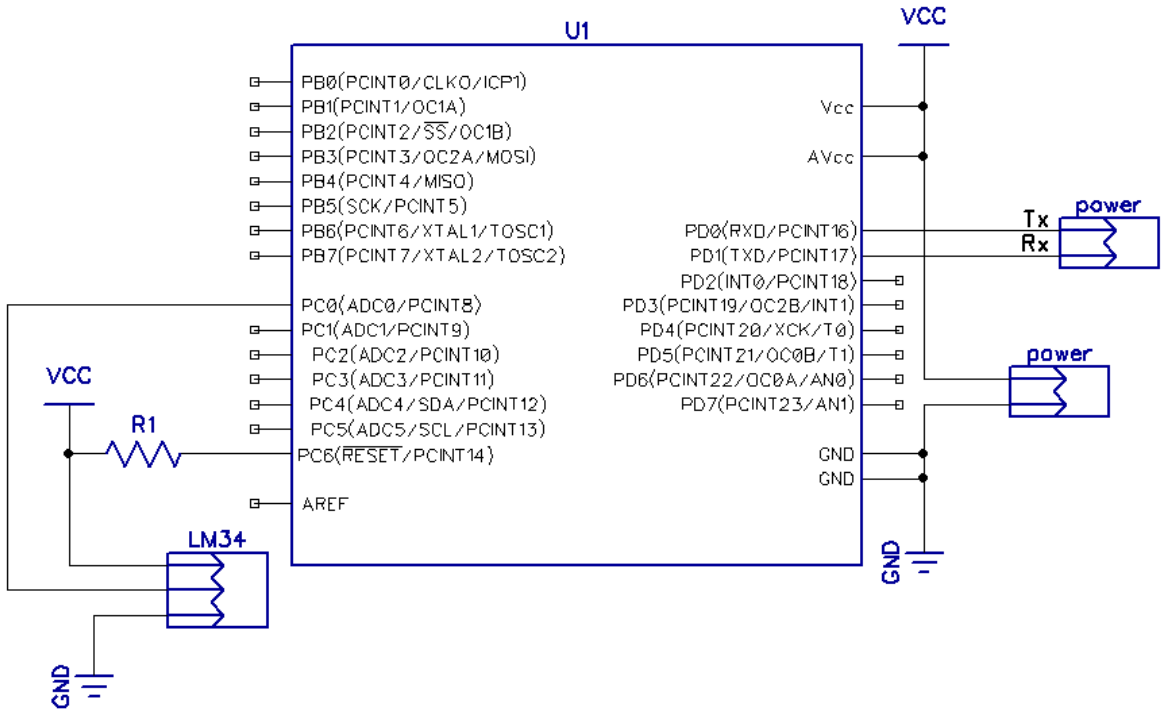



```
#define     F_CPU       8000000UL
#include    <stdio.h>
#include    <avr/io.h>
#include    <util/delay.h>
#include    <avr/interrupt.h>

#define BAUDRATE 9600    // set baudrate
#define ASYNCH_NORM_PRESCALER (F_CPU/16/BAUDRATE - 1) // prescaler value

int USART0_sendChar(char, FILE*);   // Send character on USART0
void USART0_init (void);            // Initialize USART0
void ADC0_init();                   // to initiate ADC
void TIMER1_init();                 // initiate timer1

// reset stream pointer
// http://www.gnu.org/savannah-checkouts/non-gnu/avr-libc/user-manual/group__avr__stdio.html
FILE USART0_stream = FDEV_SETUP_STREAM(USART0_sendChar, NULL, _FDEV_SETUP_WRITE);
```

```c
int main(void)
{

    stdout = &USART0_stream;   // change standard output to point to a USART stream

    USART0_init();      // Initiate USART0
    ADC0_init();        // Initiate ADC0
    TIMER1_init();      // Initiate TIMER1

    sei();              // enable global interrupts
    printf("\n");       // Go to the next line in case current line printed junk.
    while(1);           // infinite loop. Wait for interrupt.

    return 0;
}

ISR(TIMER1_COMPA_vect)          // interrupt routine for Output Compare Match A.
// ISR that triggers at TIMER1_COMPT_vect
{
    uint16_t ADC_value = ADC;    // read ADC conversion
    float temperature_F;
        c
    temperature_F = (ADC_value / 1024.0) * 5000;
    // 10 mV per degree Fahrenheit
    temperature_F = temperature_F / 10; // Convert to Fahrenheit
    printf("Temp(F) LM34: %3.2f\r", temperature_F);
}

void TIMER1_init()
// TIMER1_init initiates TIMER1 for 1 second delays to trigger an interrupt at 1 Hz
{
    // WGM = 0100
    TCNT1  = 0;
    OCR1A  = 2*F_CPU/2/256-1;
    TIMSK1 = (1<<OCIE1A);
    TCCR1A = 0x00;
    TCCR1B = (1<<WGM12) | (1<<CS12) | (0<<CS11) | (0<<CS10);
}

void ADC0_init()
// ADC0_init will initiate ADC acquisition at ADC0
{
    DDRC &= ~(0<<DDC0);         // Clear bit 0 of DDRC
    ADMUX = 0; // use ADC0
    ADMUX |= (1 << REFS0);      // use AVcc as the reference
    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 << ADSC);      // Start the ADC conversion
}

int USART0_sendChar(char data, FILE *stream)
/*
 * Procedure to send a single character over USART0. If character is linefeed, reset
 * line.
 * Assumes ASCII code.
 */
{
    if(data == '\n') // If linefeed, also print a return.
    {
        while(! (UCSR0A & (1<<UDRE0)) );// Wait for data register to be available.
        UDR0 = '\r';    // send return char to data register.
    }
    while(! (UCSR0A & (1<<UDRE0)) );    // Wait for data register to be available.
    UDR0 = data;        // send char to UART.
    return 0;
}
```

```
void USART0_init (void)
/*
 * Procedure to initialize USART0 asynchronous with enabled RX/TX, 8 bit data,
 * no parity, and 1 stop bit.
*/
{
    UCSR0B = (1<<TXEN0)  | (1<<RXEN0);  // enable transmit/receive
    UCSR0C = (1<<UCSZ01) | (1<<UCSZ00); // asynchronous, 8N1
    UBRR0L = ASYNCH_NORM_PRESCALER;     // To set 9600 baud rate with 8MHz clock
}
```

6.	SCHEMATICS		



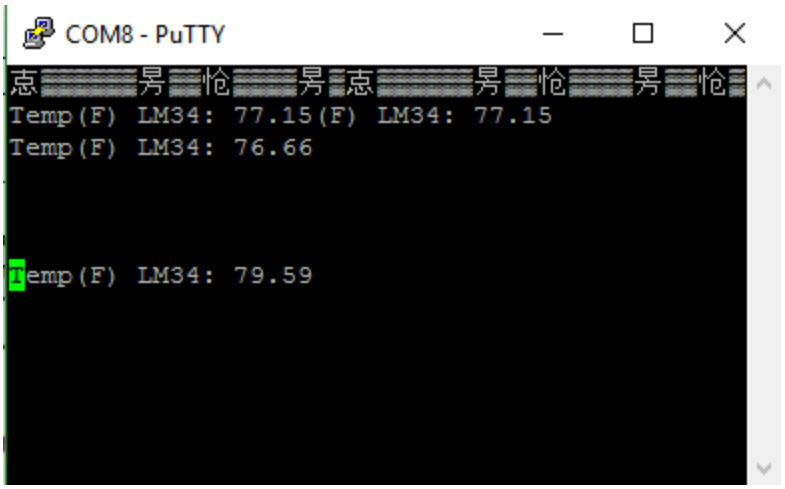
7.	SCREENSHOTS OF EACH TASK OUTPUT		

TASK 1:

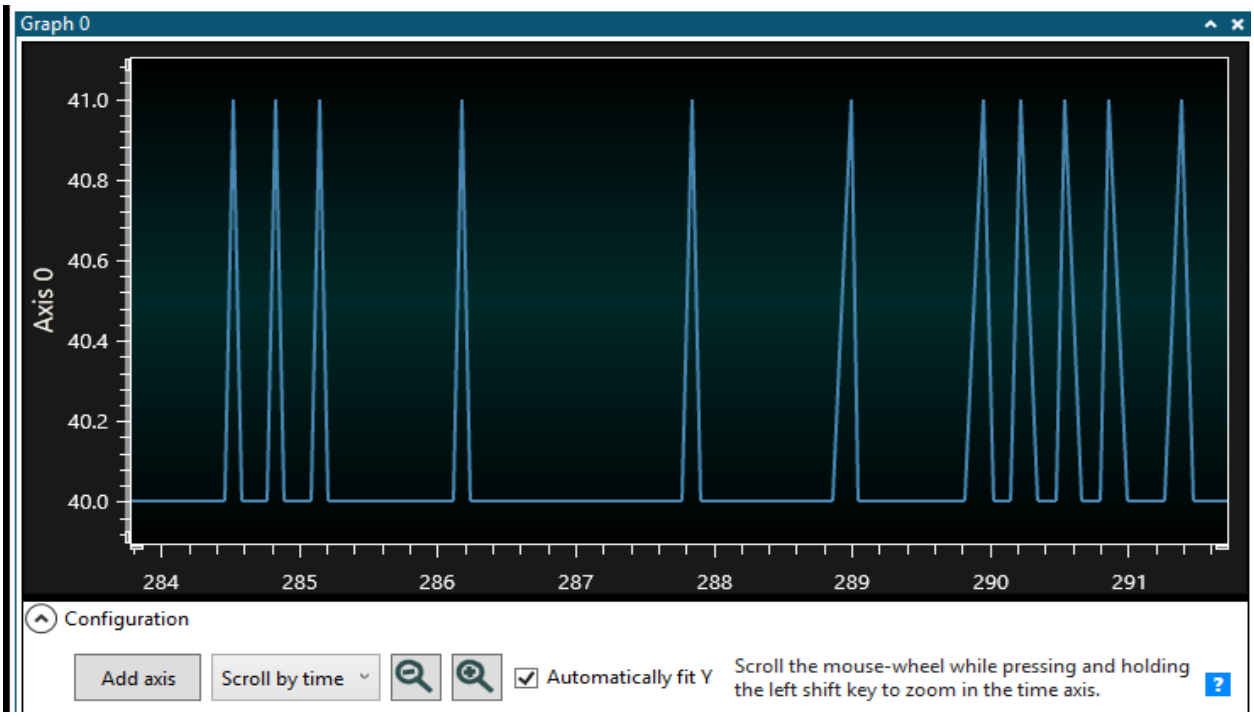
At a frequency of 1Hz, the current captured temperature was sent over UART and printed on the serial terminal. By using

```
printf("Temp(F) LM34: %3.2f\r", temperature_F);
```

The information could be printed on a single line since the return character '\r' made the next incoming text print over the old.


COM8 - PuTTY
Temp(F) LM34: 77.15(F) LM34: 77.15
Temp(F) LM34: 76.66
Temp(F) LM34: 79.59

Task2

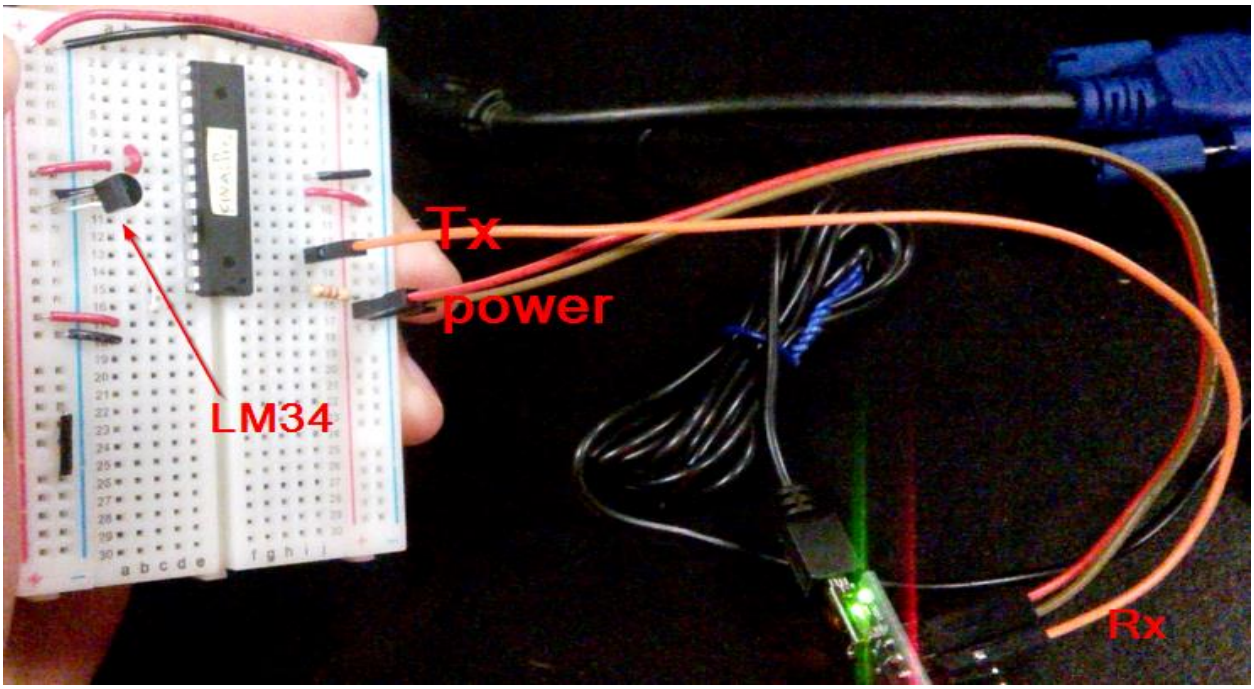

Graph 0
41.0
40.8
40.6
40.4
40.2
40.0
Axis 0
284
285
286
287
288
289
290
291
Configuration
Add axis
Scroll by time
Automatically fit Y
Scroll the mouse-wheel while pressing and holding the left shift key to zoom in the time axis.


(Temp is: 78.13
(Temp is: 78.13
) Temp is: 80.08
) Temp is: 80.08
(Temp is: 78.13
) Temp is: 80.08
) Temp is: 80.08
) Temp is: 80.08
) Temp is: 80.0
Clear
Add \r\n
Hexadecimal Values
Show Timestamp
Automatically Scroll to End
Graph 0
120
100
80
60
40
20
0
Axis 0
Cursor 0.0.3
20791.5
20792.0
20792.5
20793.0
20793.5
20794.0
20794.5
Configuration
Add axis
Scroll by time
Automatically fit Y
Scroll the mouse-wheel while pressing and holding the left shift key to zoom in the time axis.


Terminal 0
& Temp is: 74.22
% Temp is: 72.27
% Temp is: 72.27
% Temp is: 72.27
% Temp is: 72.27
& Temp is: 74.22
& Temp is: 74.22
& Temp is: 74.22
&
Clear
Add \r\n
Hexadecimal Values
Show Timestamp
Automatically Scroll to End
Graph 0
120
100
80
60
40
20
0
Axis 0
Cursor 0.0.3
20859.5
20860.0
20860.5
20861.0
20861.5
20862.0
20862.5
Configuration
Add axis
Scroll by time
Automatically fit Y
Scroll the mouse-wheel while pressing and holding the left shift key to zoom in the time axis.

8.	SCREENSHOT OF EACH DEMO		

Task1



9.	VIDEO LINKS OF EACH DEMO		
10.	GITHUB REPOSITORY LINK OF THE DA		
https://github.com/Vasty1995/CPE301			

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Yannick Kengne Tatcha