CPE301 – SPRING 2019

MIDTERM 1

Student Name: Cody Hudson

Student #: 8000603642

Student Email: hudsoc1@unlv.nevada.edu

Primary Github address: https://github.com/eed911/class\_proj.git

Directory: https://github.com/eed911/class\_proj/tree/master/Midterm

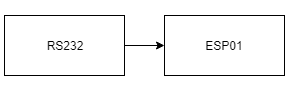
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used:

* Atmega 328p
* RS232
* LM35
* ESP01

Block diagram with pins used in the Atmega328P:

Programing ESP:



Communicating to ThingSpeak:



1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

/\*

\* SUMSTUFF.c

\*

\* Created: 4/6/2019 10:06:39 AM

\* Author : hudsoc1

\*/

// Variable Definitions

#define *F\_CPU* 16000000UL // MCU operating Frequency

#define BAUD 9600 // Baudrate setting

#define BAUD\_PRESCALLER *F\_CPU*/16/BAUD-1 // Baudrate prescaller

// Included Libraries

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <stdlib.h>

//Function Declaration

void read\_adc(void); //Read LM34 using ADC

void adc\_init(void); // Setup ADC

void USART\_init( unsigned int ubrr ); // Setup USART communication

void USART\_tx\_string(char \*data); //Print String from USART

volatile unsigned int adc\_temp; // temperature variable

volatile unsigned int tempF; // Fahrenheit temperature variable

volatile unsigned int tempC; // Celsius temperature variable used for converting to Fahrenheit

char outs[256]; // String array used for sending USART commands

volatile char received\_data; // String array used for receiving USART communication

int main(void) {

adc\_init(); // Setup Analog to Digital Converter on PC4

USART\_init(BAUD\_PRESCALLER); // Setup the USART (RS232 interface)

*\_delay\_ms*(500); // Delay to allow hardware to initialize

while(1)

{

// Setup AT commands for communicating to thingspeak

//AT connect check

unsigned char AT[] = "AT\r\n";

//Set device mode, 1 = Station mode

unsigned char AT\_CWMODE[] = "AT+CWMODE=1\r\n";

//Perform Wifi connection, provide SSID and Password // Used Clayton Higbee's Mobil Hotspot

unsigned char AT\_CWJAP[] = "AT+CWJAP=\"Higbee\_Pixel\",\"Higbee19\"\r\n";

//Set device for single IP Address Mode

unsigned char AT\_CIPMUX[] = "AT+CIPMUX=0\r\n";

// Start TCP connection to Thingspeak.com at port 80

unsigned char AT\_CIPSTART[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n";

// Give upcoming string length

unsigned char AT\_CIPSEND[] = "AT+CIPSEND=100\r\n";

// Send commands listed from above

*\_delay\_ms*(200);

USART\_tx\_string(AT);

*\_delay\_ms*(5000);

USART\_tx\_string(AT\_CWMODE);

*\_delay\_ms*(5000);

USART\_tx\_string(AT\_CWJAP);

*\_delay\_ms*(15000);

USART\_tx\_string(AT\_CIPMUX);

*\_delay\_ms*(10000);

USART\_tx\_string(AT\_CIPSTART);

*\_delay\_ms*(10000);

USART\_tx\_string(AT\_CIPSEND);

*\_delay\_ms*(5000);

PORTC^=(1<<5);

// Read next raw ADC value from LM35

read\_adc();

// Convert raw into Celsius and Fahrenheit

adc\_temp = (adc\_temp\*500)/1023;

tempC = adc\_temp;

tempF = (tempC\*1.8)+32;

// Print Data to Thingspeak using provided link, website channel key, and field location

*snprintf*(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api\_key=EQ3IBZ6YL429LH17&field1=%3d\r\n", tempF);

USART\_tx\_string(outs);//send data

*\_delay\_ms*(10000);

}

}

void adc\_init(void) //initialize ADC

{

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

(1<<REFS0)| // AVcc - external cap at AREF

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

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

(0<<MUX1)| // ADC4 (PC4)

(0<<MUX0); // 010

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

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

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

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

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

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

(0<<ADPS1)|

(1<<ADPS0);

}

void read\_adc(void) {

unsigned char i =4;

adc\_temp = 0; //initialize adc\_temp variable

while (i--) {

ADCSRA |= (1<<ADSC);

while(ADCSRA & (1<<ADSC));

adc\_temp+= ADC;

*\_delay\_ms*(50);

}

adc\_temp = adc\_temp / 4; // Average a few samples

}

/\* Initialize USART communication (RS-232) \*/

void USART\_init( unsigned int ubrr ) {

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // Enable receiver, transmitter & RX interrupt

UCSR0C |= (1<<UCSZ01) | (1 << UCSZ00);

}

/\* USART string sender (RS-232) \*/

void USART\_tx\_string( char \*data ) {

while ((\*data != '\0')) {

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

UDR0 = \*data;

data++;

}

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

adc\_init(); // Setup Analog to Digital Converter on PC4

USART\_init(BAUD\_PRESCALLER); // Setup the USART (RS232 interface)

*\_delay\_ms*(500); // Delay to allow hardware to initialize

while(1)

{

// Setup AT commands for communicating to thingspeak

//AT connect check

unsigned char AT[] = "AT\r\n";

//Set device mode, 1 = Station mode

unsigned char AT\_CWMODE[] = "AT+CWMODE=1\r\n";

//Perform Wifi connection, provide SSID and Password // Used Clayton Higbee's Mobil Hotspot

unsigned char AT\_CWJAP[] = "AT+CWJAP=\"Higbee\_Pixel\",\"Higbee19\"\r\n";

//Set device for single IP Address Mode

unsigned char AT\_CIPMUX[] = "AT+CIPMUX=0\r\n";

// Start TCP connection to Thingspeak.com at port 80

unsigned char AT\_CIPSTART[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n";

// Give upcoming string length

unsigned char AT\_CIPSEND[] = "AT+CIPSEND=100\r\n";

// Send commands listed from above

*\_delay\_ms*(200);

USART\_tx\_string(AT);

*\_delay\_ms*(5000);

USART\_tx\_string(AT\_CWMODE);

*\_delay\_ms*(5000);

USART\_tx\_string(AT\_CWJAP);

*\_delay\_ms*(15000);

USART\_tx\_string(AT\_CIPMUX);

*\_delay\_ms*(10000);

USART\_tx\_string(AT\_CIPSTART);

*\_delay\_ms*(10000);

USART\_tx\_string(AT\_CIPSEND);

*\_delay\_ms*(5000);

PORTC^=(1<<5);

// Read next raw ADC value from LM35

read\_adc();

// Convert raw into Celsius and Fahrenheit

adc\_temp = (adc\_temp\*500)/1023;

tempC = adc\_temp;

tempF = (tempC\*1.8)+32;

// Print Data to Thingspeak using provided link, website channel key, and field location

*snprintf*(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api\_key=EQ3IBZ6YL429LH17&field1=%3d\r\n", tempF);

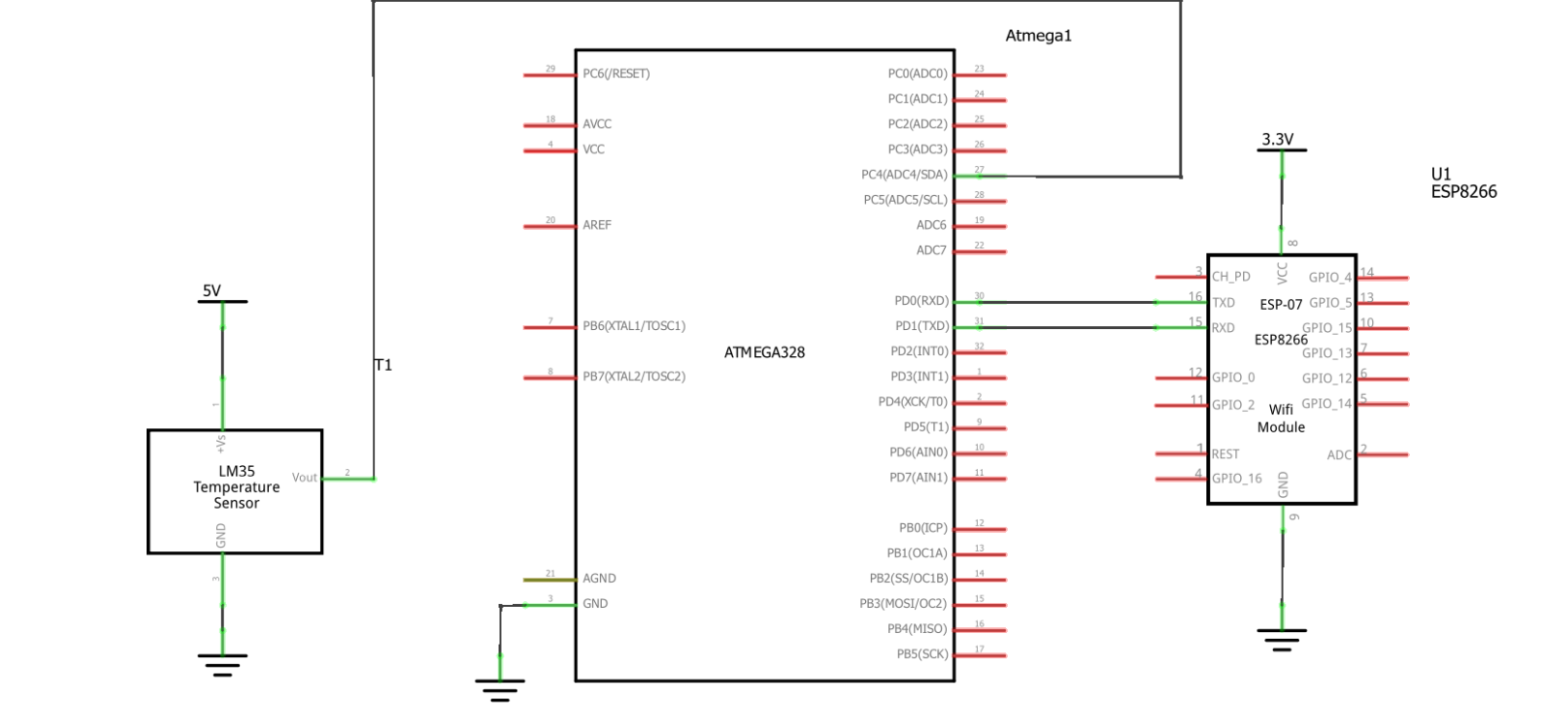
USART\_tx\_string(outs);//send data

*\_delay\_ms*(10000);

}

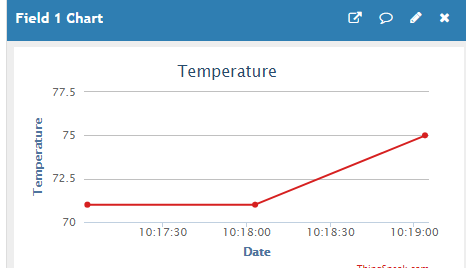
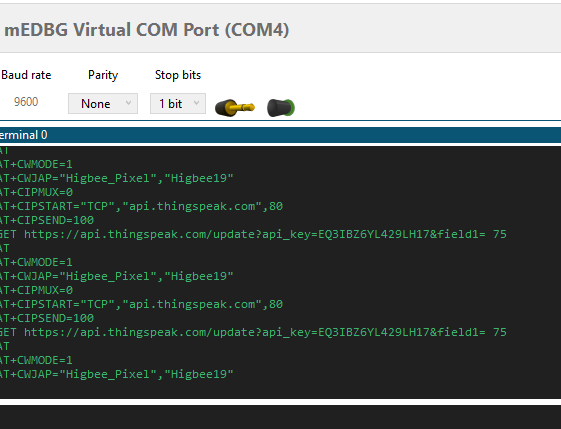
}

1. **SCHEMATICS**

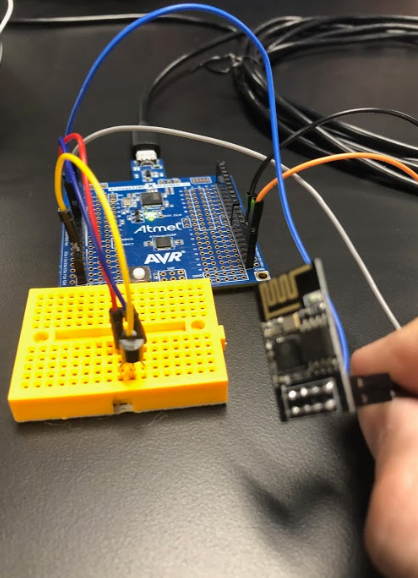


Use fritzing.org

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**

Demo1:

<https://youtu.be/JNd6IuN8Vys>

1. **GITHUB LINK OF THIS DA**

https://github.com/eed911/class\_proj.git

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT