CPE301 – SPRING 2019

MIDTERM 1

Student Name: Eric Smith

Student #: 2000543940

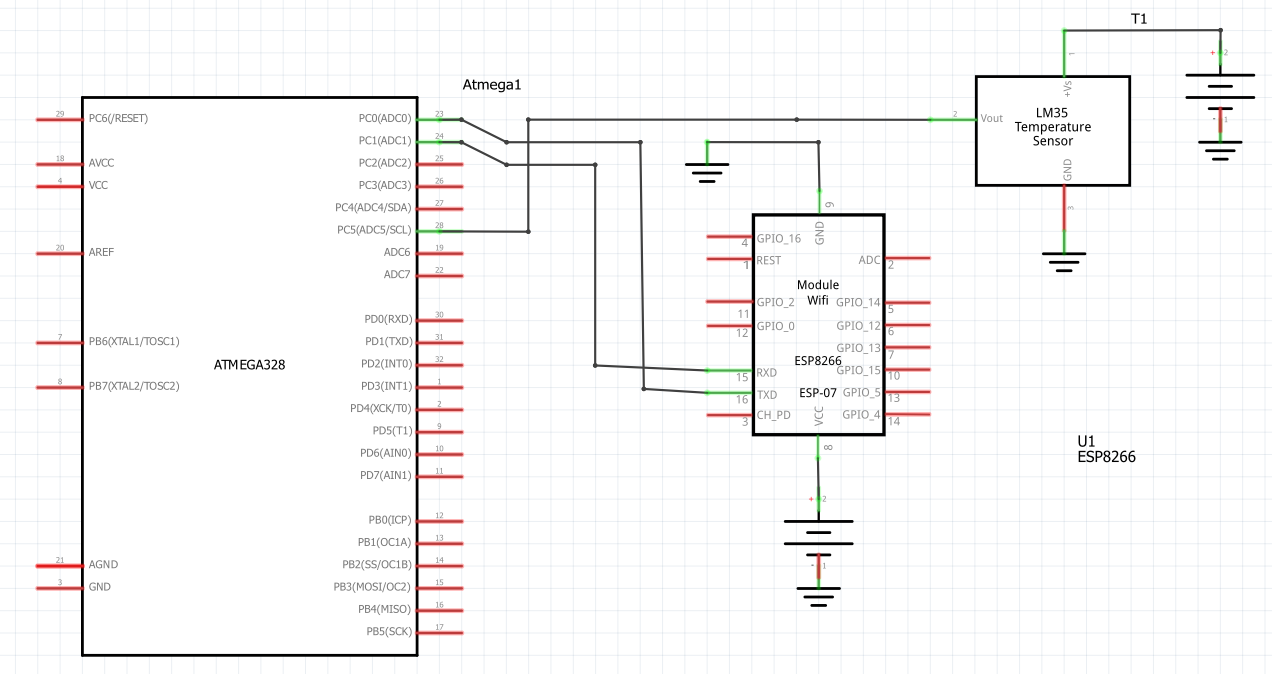
Student Email: smithe4@unlv.nevada.edu

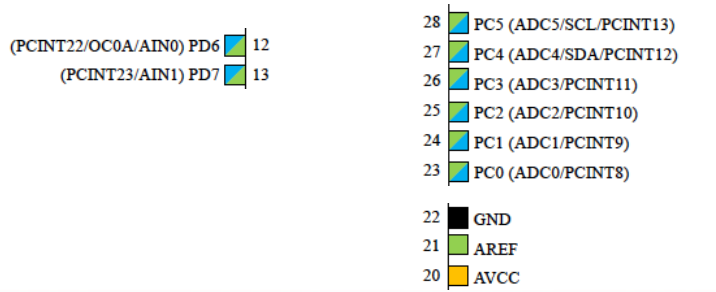
Primary Github username: smitheas95

Directory: Repository

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

LIST: ESP-01 module, UNLV ECE board, LM35 temperature sensor, 5V, 3.3V, GND, PC4





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

/\*

\* Midterm1.c

\*

\* Created: 4/7/2019 10:23:02 AM

\* Author : Eric Smith

Q: Write, simulate, and demonstrate using Atmel Studio 7 a C code for the AVR

ATMEGA328p microcontroller that performs the following functions:

1. Program the ADC of ATmega328/p to read the LM34/35 temperature sensor.

2. Display the value to UART.

3. Make sure the AT Firmware is downloaded into the ESP-01/ESP32 module.

4. Register for a free Thingspeak account with MATHWORK. Setup and get the

channel Key.

5. Transmit temperature sensor value to ESP-01/ESP32 through UART port using AT

Commands.

6. Display the temperature sensor value as a graph in Thingspeak

\*/

#define *F\_CPU* 16000000UL // SET FREQUENCY

#define BAUD\_PRESCALLER *F\_CPU*/16/BAUD-1 // PRESCALLER FOR BAUDRATE

#define BAUD 9600 // SETTING FOR BAUDRATE

// NEEDED LIBRARIES

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdio.h>

#include <stdlib.h>

// DECLARATION OF FUNCTIONS

void read\_adc(void); // ADC TO READ LM35

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

void USART\_init( unsigned int ubrr ); // USART COMMUNICATION SET UP

void USART\_tx\_string(char \*data); // USART PRINT STRING

char outs[256]; // ARRAY STRING FOR SENDING USART COMMANDS

volatile char received\_data; // ARRAY STRING FOR RECIEVING USART INFO

volatile unsigned int adc\_temp; // TEMP VARIABLE, RAW VOLATILE

volatile unsigned int tempF; // TEMP VARIABLE, VOLATILE FAHRENHEIT

volatile unsigned int tempC; // TEMP VARIABLE, VOLATILE CELSIUS

int main(void) {

adc\_init(); // ANALOG TO DIGITAL CONVERTER

USART\_init(BAUD\_PRESCALLER); // USART INITIALIZATION (RS232 INTERFACE)

*\_delay\_ms*(500); // DELAY IN ORDER FOR HARDWARE TO WORK PROPERLY

// WHILE LOOP WORKS TO READ TEMP VALUE FROM ADC AND PRINT ONTO THINGSPEAK

while(1)

{

unsigned char AT[] = "AT\r\n"; // 1 = STATION MODE

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

unsigned char AT\_CWJAP[] = "AT+CWJAP=\"SSID\",\"PASSWORD\"\r\n"; // SSID, PW

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

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

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

// DELAYS FOR PURPOSE OF SEND COMMANDS

*\_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);

// KEY PORT

PORTC ^= (1 << 5);

read\_adc();

adc\_temp = (adc\_temp\*500)/1023; // CONVERSION INTO CELSIUS AND FAHRENHEIT

tempC = adc\_temp;

tempF = (tempC\*1.8)+32;

// PRINT DATA TO THINGSPEAK PROCESS

*snprintf*(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api\_key=6Q0VYEFULHGEADRA&field1=0", tempF);

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

*\_delay\_ms*(10000);

}

}

// INITIALIZE ADC

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

// MUX ADJUSTMENT

(1<<MUX2)| // ANALOG CHANNEL SELECTION BITS

(0<<MUX1)| // (PC2 PIN25) ADC2

(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);

}

// READING VALUE FROM ADC METHOD

void read\_adc(void)

{

unsigned char i =4;

adc\_temp = 0; // ADC\_TEMP VARIABLE SET TO ZERO, INITIALIZED

while (i--) {

ADCSRA |= (1<<ADSC);

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

adc\_temp+= ADC;

*\_delay\_ms*(50);

}

adc\_temp = adc\_temp / 4; // AVG SAMPLES, CHOOSE 4

}

// USART STRING SENDER (RS-232)

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

{

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

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

UDR0 = \*data;

data++;

}

}

// INITIALIZE USART COMMUNICATION

void USART\_init( unsigned int ubrr )

{

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

UBRR0L = (unsigned char)ubrr;

UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // ENABLE RECEIVER, RX INTERRUPT AND TRANSMITTER

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

}

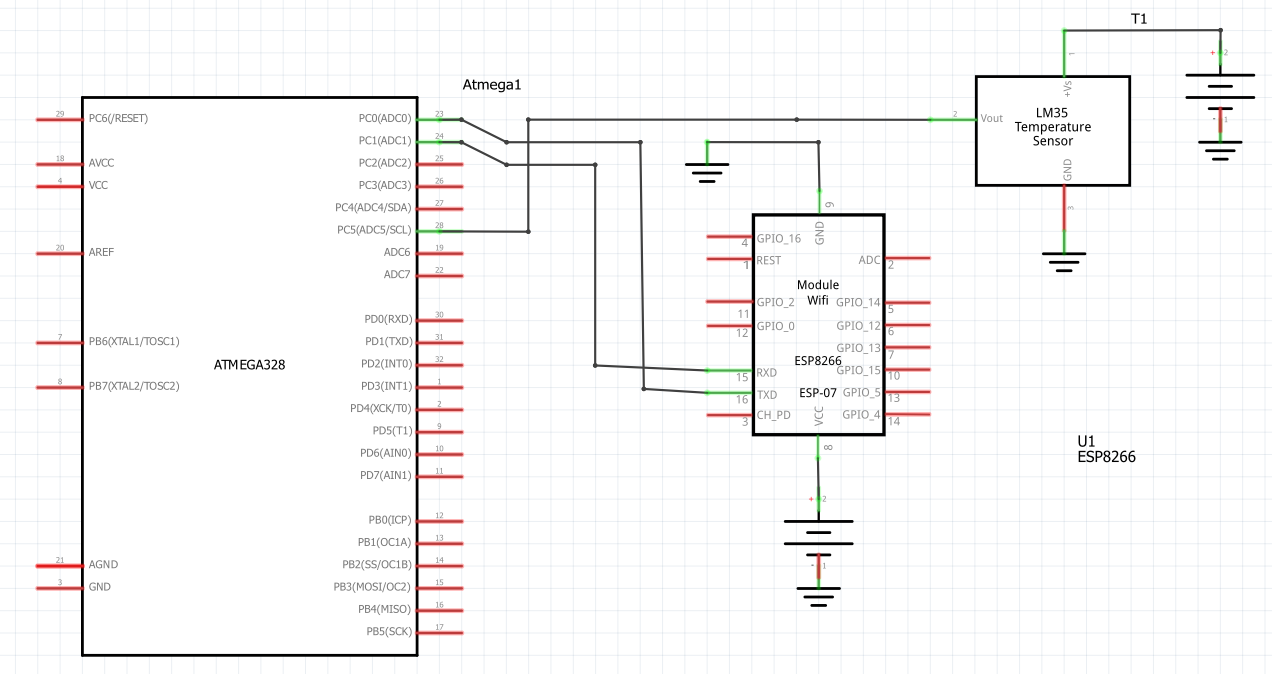
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

(N/A)

Not needed, #2 covers this portion

1. **SCHEMATICS**

Key schematic:

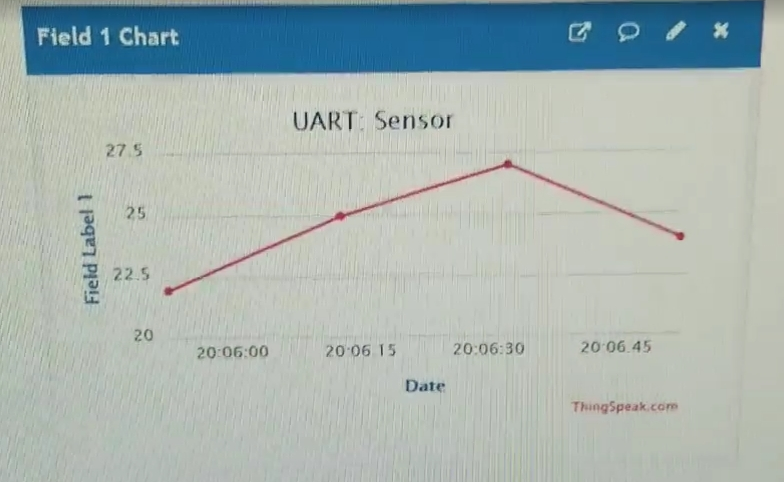


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL/THINGSPEAK OUTPUT)**

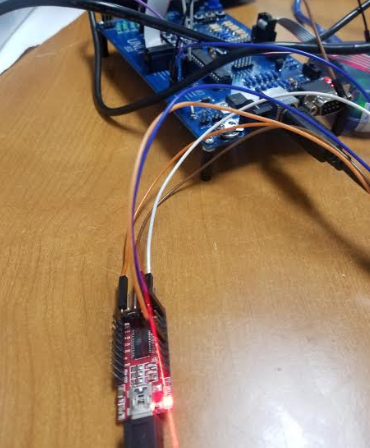
Thingspeak username:

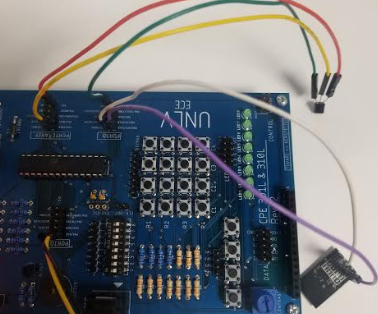


RESULTS:



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





1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/tXExTmjkwJY>

1. **GITHUB LINK OF THIS DA**

<https://github.com/smitheas95/submission_da>

**Student Academic Misconduct Policy**

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

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

Eric Smith