CPE301 – SPRING 2019

MIDTERM 1

Student Name: Cody McDonald

Student #:5000382538

Student Email: mcdonc4@unlv.nevada.edu

Primary Github address: <https://github.com/elev8rProcrastinator>

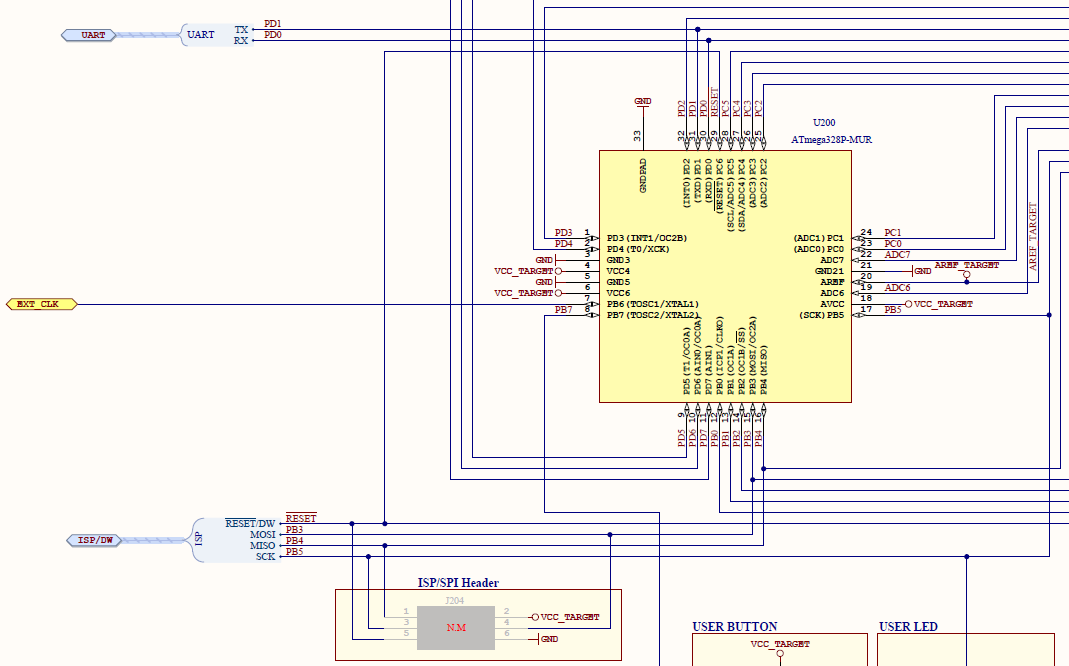
Directory: https://github.com/elev8rProcrastinator/submission\_da/DA\_Midterm

Submit the following for all Labs:

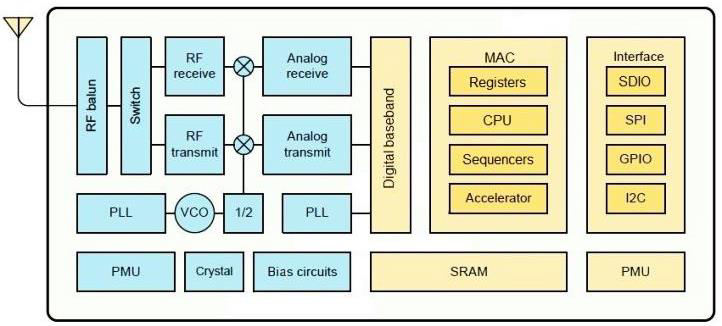
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/Midterm, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

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

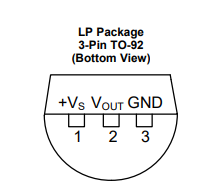
AtMini xplained:



ESP8266:



LM35 temperature sensor:



Connected setup will be located in schematic description below.

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

/\*

\* CPE301\_Midterm\_V3.c

\*

\* Created: 4/6/2019 4:12:34 PM

\* Author : Cody

\*/

#define *F\_CPU* 16000000UL

//Include Header Files

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdint.h>

#include <stdlib.h>

// Global constants for uart

#define BAUD 115200

#define FOSC 16000000

#define UBRR FOSC/8/BAUD-1

// Global variables

volatile *uint8\_t* ADCdata;

volatile unsigned char temp[10];

//AT COMMAND STRINGS

volatile unsigned char AT[] = "AT\r\n"; // Test

volatile unsigned char CWMODE[] = "AT+CWMODE=3\r\n"; // Set Wi-Fi mode

volatile unsigned char CWJAP[] = "AT+CWJAP=\"Monarchs Palace\",\"quackquack\"\r\n"; // Get Wi-Fi info

volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n"; // Establish connection with ThingSpeak

volatile unsigned char CIPSEND[] = "AT+CIPSEND=50\r\n"; // Send Temperature

volatile unsigned char CIPMUX[] = "AT+CIPMUX=0\r\n"; // Enable connection

volatile unsigned char SEND\_DATA[] = "GET /update?key=ZDG1BP942G9NVEWD&field1="; // Get Write Key

volatile unsigned char RESET[] = "AT+RST\r\n"; // Get AT Firmware info

volatile unsigned char LINEBREAK[] = "\r\n"; // end of temperature transmission

volatile unsigned char CLOSE[] = "AT+CIPCLOSE\r\n";

//initialize functions

void ADC\_init();

void usart\_init();

void usart\_send(volatile unsigned char data[]);

int main(void)

{

// initialize modes

ADC\_init(); //initialize ADC

usart\_init(); //initialize usart

sei();

// Establish initial connection the thingspeak

*\_delay\_ms*(1000);

usart\_send(RESET); //reset ESP

*\_delay\_ms*(1000);

usart\_send(AT); //confirm communication

*\_delay\_ms*(1000);

usart\_send(CWMODE); //WiFi mode = 3

*\_delay\_ms*(1000);

usart\_send(CWJAP); //Send wifi login

*\_delay\_ms*(1000);

usart\_send(CIPMUX); //Single connection point

while (1)

{

*\_delay\_ms*(1000);

usart\_send(CIPSTART); // Connect to ThingSpeak

*\_delay\_ms*(1000);

usart\_send(CIPSEND); // Declare send length 50

*\_delay\_ms*(1000);

usart\_send(SEND\_DATA); // Connect to proper key

usart\_send(temp); // Send adc data

usart\_send(CLOSE); // close connection

usart\_send(LINEBREAK); // line break

}

return 0;

}

void ADC\_init(){

ADMUX |= (1 << REFS0); // use AVcc

ADMUX |= (1 << ADLAR); // Right adjust

ADCSRA = (1 << ADEN) // Enable

|(1 << ADPS1)

|(1 << ADPS0) // 128 prescaler for 16Mhz

|(1 << ADATE) // ADC Auto Trigger

|(1 << ADIE) // Enable Interrupts

|(1 << ADSC); // Start ADC

}

void usart\_init() {

UBRR0H = ((UBRR) >> 8);

UBRR0L = UBRR;

UCSR0A |= (1<< U2X0); // divisor baud = 8

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

UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // 8 bits

}

// ADC value Interrupt subroutine

ISR(ADC\_vect) {

unsigned char i;

char dummy[10];

ADCdata = (ADCH << 1) \* 1.8 + 32; // Convert Celsius to Fahrenheit

*itoa*(ADCdata, dummy, 10); //convert char to ascii

for(i = 0 ; i < 10 ; i++){

temp[i] = dummy[i]; //move converted ascii

}

}

void usart\_send(volatile unsigned char data[]) {

volatile unsigned char length = 0;

volatile unsigned char j;

// write char array to UART register

while(data[length] != 0){

length++;

for(j = 0 ; j < length ; j++)

{

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

UDR0 = data[j]; // send char

}

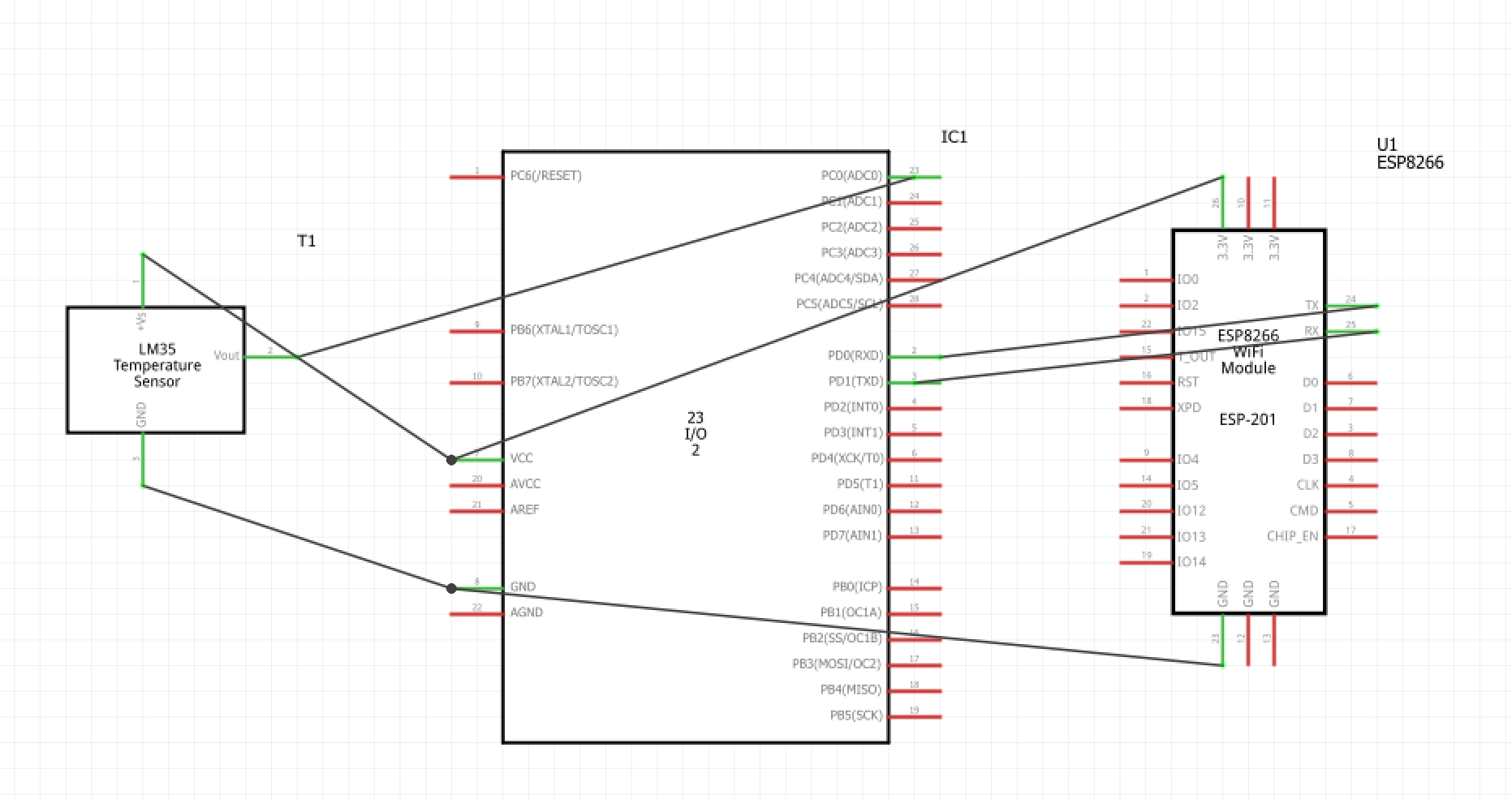
}

}

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

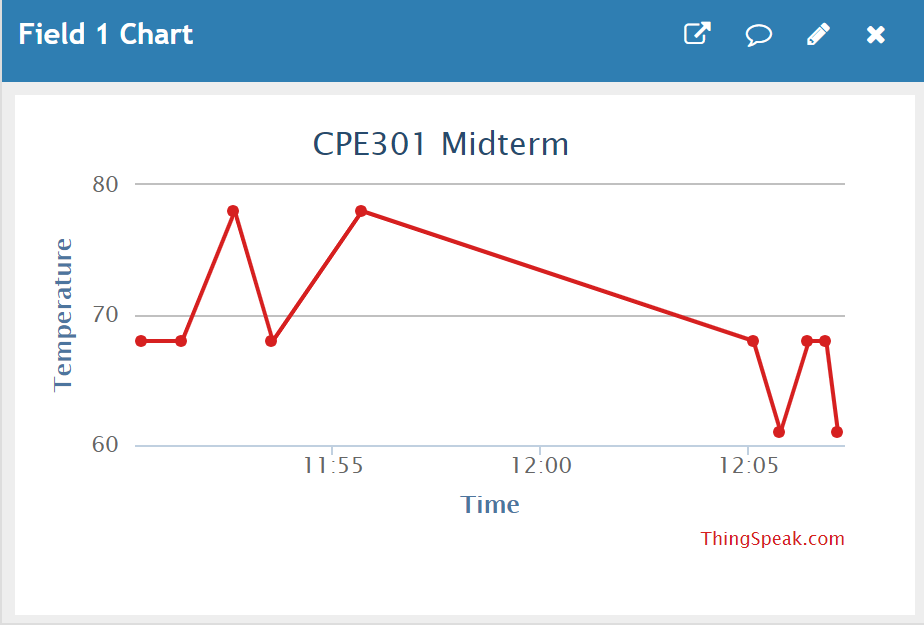
N/A there was no modified code.

1. **SCHEMATICS**

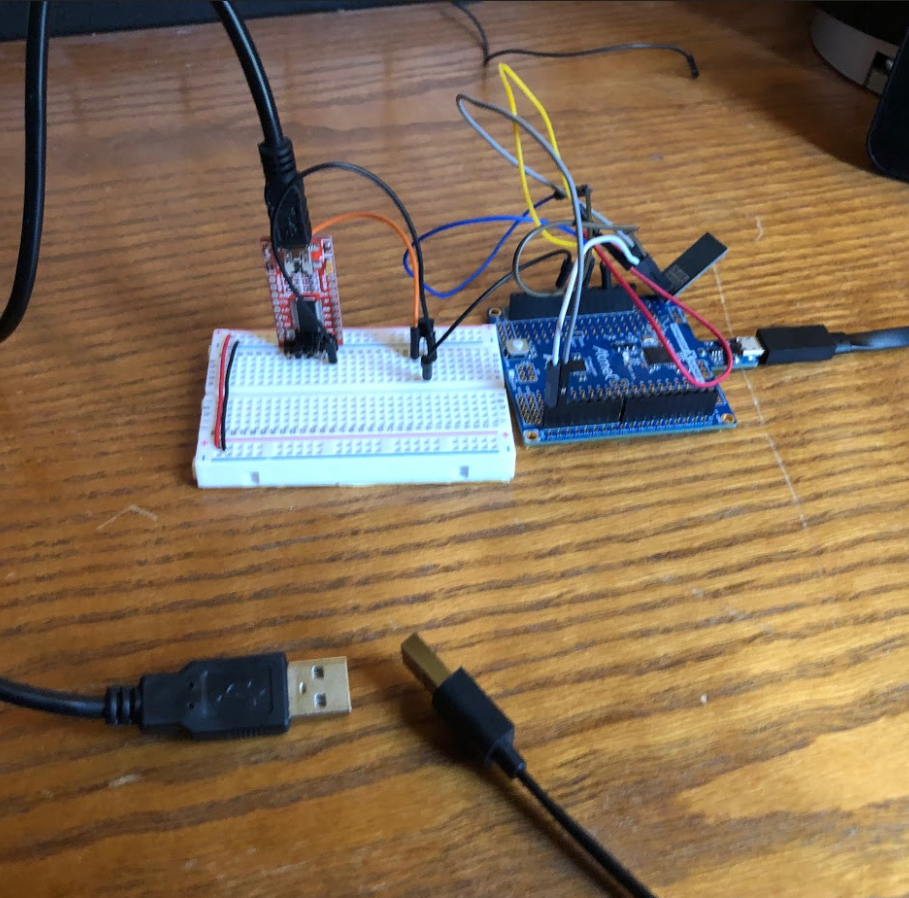


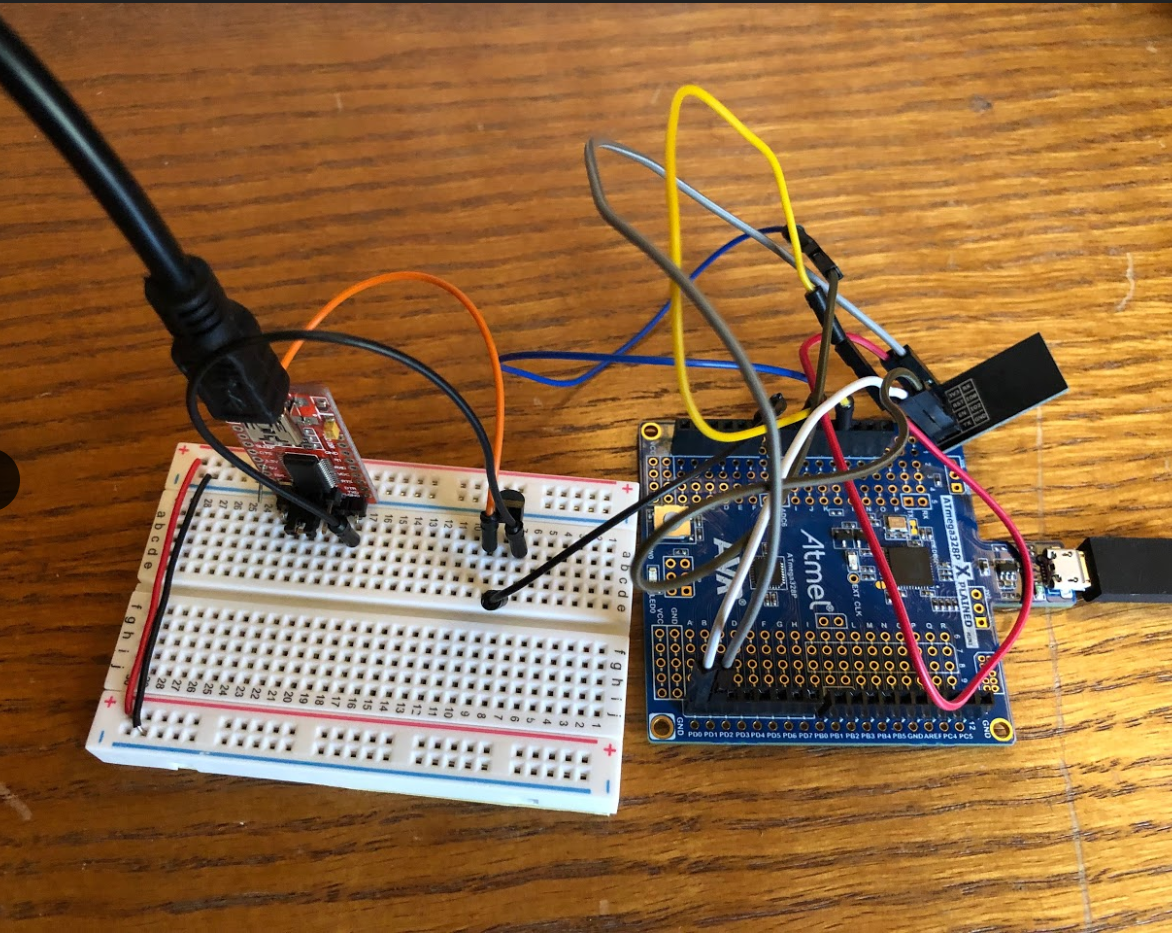
1. **SCREENSHOTS OF EACH TASK OUTPUT (THINGSPEAK OUTPUT)**

Temperature data sent to Thingspeak:



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





1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/Z6EL1btv30s

1. **GITHUB LINK OF THIS DA**

https://github.com/elev8rProcrastinator/submission\_da/DA\_Midterm

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT