CPE301 – FALL 2019

MIDTERM 1

Student Name: Dillon Archibald
Student #: 5004439916
Student Email: Archid1@unlv.nevada.edu
Primary Github address: https://github.com/Dil-bert/Alabaster.git
Directory: Mid_Term_I

Submit the following for all Labs:

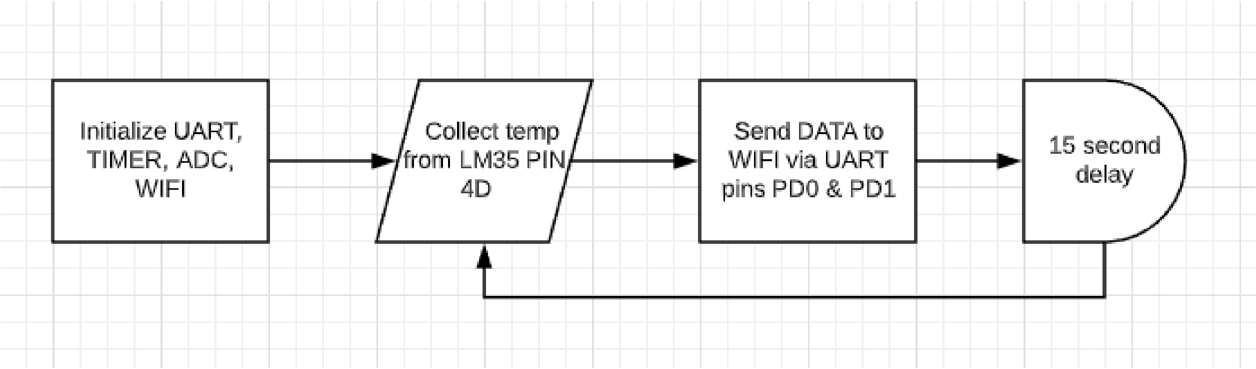
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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

List of Components used

1 x Atmega 328p Xplained mini
1 x Multi-function sheild
1 x ESP8266-01 Wifi module
1 x LM35 temperature sensor

Block diagram with pins used in the Atmega328P


Initialize UART, TIMER, ADC, WIFI
Collect temp from LM35 PIN 4D
Send DATA to WIFI via UART pins PD0 & PD1
15 second delay

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

```
/*
* DA3B.c
*
* Created: 10/24/2019 9:12:14 PM
* Author : Dilbert
*/
#define F_CPU 16000000UL

#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define USART_BAUDRATE 9600
#define UBRR_VALUE (((F_CPU/(USART_BAUDRATE*16UL)))-1)

// Global Variables
volatile uint8_t tempL = 0;
volatile uint8_t tempH = 0;
volatile float tempOut = 0;

// Function definitions
void InitPort();

void InitADC();
void SetADCChannel(uint8_t ADCchannel);
void StartADC(void);
void DisableADC(void);

void USART0Init(void);
void InitTimer0(void);
int USART0SendByte(char u8Data,FILE *stream);
void StartTimer0(void);
void StopTimer(void);

ISR(ADC_vect);

// Set Stream Pointer
FILE usart0_str = FDEV_SETUP_STREAM(USART0SendByte, NULL, _FDEV_SETUP_WRITE);

//------------------------------------------------------------------------------
//    **********MAIN**********
```

```
//-----------------------------------------------------------------------------
int main(void){
    // Initialize USART0
    USART0Init();
    // Initialize ports
    InitPort();
    // Assign our stream to standard I/O streams
    stdout=&usart0_str;
    // Initialize ADC
    InitADC();
    // Select ADC channel
    SetADCChannel(5);
    // Initialize timer0
    InitTimer0();
    // Start timer 0
    StartTimer0();
    // Start conversion
    StartADC();
    // Enable global interrupts
    sei();
    while (1){
        // 1 Second Delay between displaying temp
        _delay_ms(1000);
        // Temp set up
        tempOut = ((tempOut * 0.488));
        // Temp Conversion from Celsius to Fahrenheit
        tempOut = (((9*tempOut)/5) + 32);
        printf("Temp = %.1f F\r\n",tempOut);
    }
}

//-----------------------------------------------------------------------------
//    *********Port Initialization function*********
//-----------------------------------------------------------------------------
void InitPort(){
    // Set pin C5 as an input pin
    DDRC |= (0<<PINC5);
    // Ensure pin C5 pull up resistor is off
    PORTC = (0<<PINC5);
    // Turn off Digital logic on pin C5
    DIDR0 |= (1<<ADC5D);
}

//-----------------------------------------------------------------------------
//    *********ADC Initialization function*********
//-----------------------------------------------------------------------------
void InitADC(){
    // Select Vref=Avcc and set (left = ADLAR = 1)(right = ADLAR = 0) justified result
    ADMUX |=(1<<REFS0)|(0<<ADLAR);
    // Set prescaller to 32, enable auto triggering, enable ADC interrupt
    //  and enable ADC
    ADCSRA |=(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE)|(1<<ADIE)|(1<<ADEN);
    // Set ADC trigger source - Timer0 compare match A
    ADCSRB |=(1<<ADTS1)|(1<<ADTS0);
}

//-----------------------------------------------------------------------------
//    *********ADC PORT SELECT function*********
```

```
//-----------------------------------------------------------------------------
void SetADCChannel(uint8_t ADCchannel){
    // Select ADC channel with safety mask
    ADMUX = (ADMUX & 0xF0)|(ADCchannel & 0x0F);
}

//-----------------------------------------------------------------------------
//     *********ADC start function*********
//-----------------------------------------------------------------------------
void StartADC(void){
    ADCSRA |=(1<<ADSC);
}

//-----------------------------------------------------------------------------
//     *********ADC end function*********
//-----------------------------------------------------------------------------
void DisableADC(void){
    ADCSRA &= ~((1<<ADEN)|(1<<ADIE));
}

//-----------------------------------------------------------------------------
//     *********USART Initialization function*********
//-----------------------------------------------------------------------------
void USART0Init(void){
    // Set baud rate
    UBRR0H = (uint8_t)(UBRR_VALUE>>8);
    UBRR0L = (uint8_t)UBRR_VALUE;
    // Set frame format to 8 data bits no parity, 1 stop bit
    UCSR0C |= (1<<UCSZ01)|(UCSZ00);
    // Enable transmission and reception
    UCSR0B |= (1<<RXEN0) | (1<<TXEN0);
}

//-----------------------------------------------------------------------------
//     *********Timer0 Initialization function*********
//-----------------------------------------------------------------------------
void InitTimer0(void){
    // Set Initial Timer value
    TCNT0 = 0;
    // Place TOP timer value to Output compare register
    OCR0A = 99;
    // Set CTC mode
    //  and make toggle PD6/OC0A pin on compare match
    TCCR0A |= (1<<COM0A0)|(1<<WGM01);
}

//-----------------------------------------------------------------------------
//     *********USART Send function*********
//-----------------------------------------------------------------------------
int USART0SendByte(char u8Data,FILE *stream){
    // Wait while previous byte is completed
    while(!(UCSR0A & (1<<UDRE0))){};
    // Transmit Data
    UDR0 = u8Data;
    return 0;
}

//-----------------------------------------------------------------------------
```

```
//    *********Timer start function*********
//------------------------------------------------------------------------------
void StartTimer0(void){
    // Set prescaller 8 and start timer
    TCCR0B |= (1<<CS01);
}

//------------------------------------------------------------------------------
//    *********Timer end function*********
//------------------------------------------------------------------------------
void StopTimer(void){
    TCCR0B &= ~(1<<CS01);
    TIMSK0 &= ~(1<<OCIE0A);
}

//------------------------------------------------------------------------------
//    *********ADC conversion complete ISR*********
//------------------------------------------------------------------------------
ISR(ADC_vect){
    // clear timer compare match flag
    TIFR0 = (1<<OCF0A);
    // Toggle pin PD2 to track the end of ADC conversion
    PIND = (1<<PD2);
    tempL = ADCL;
    tempH = ADCH;
    tempOut = ((tempH << 8)|(tempL));
}
```

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

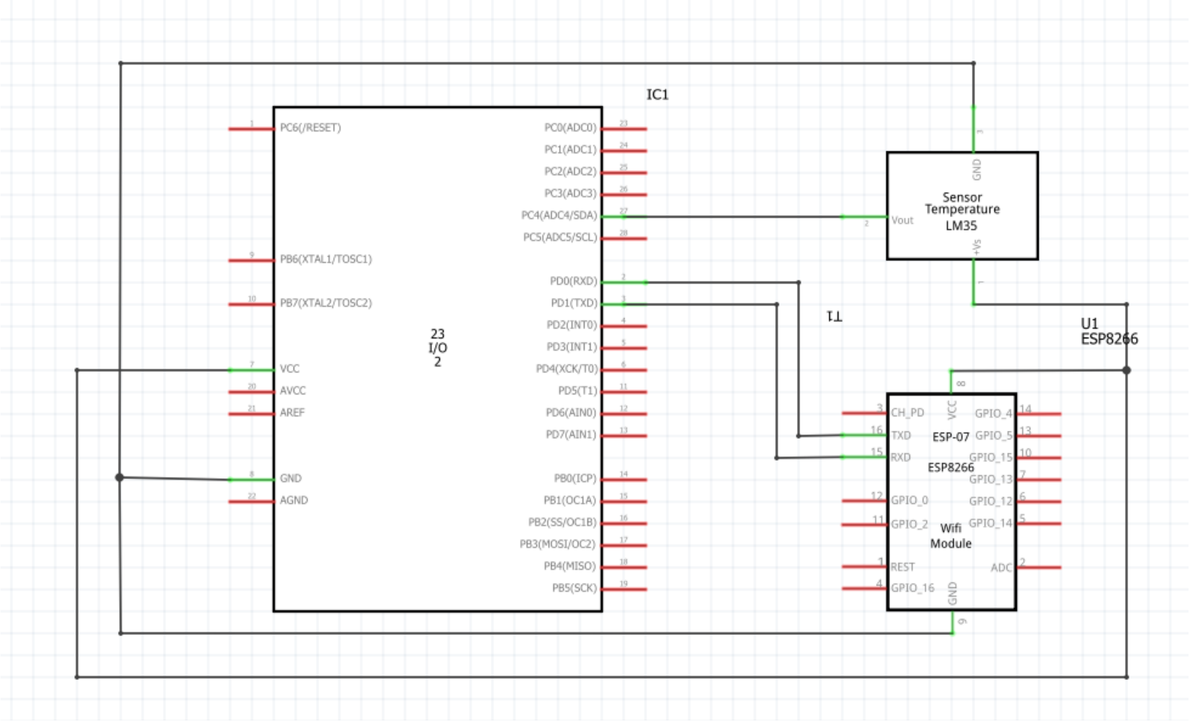
```
/*
* Mid_Term.c
*
* Created: 10/29/2019 9:43:44 AM
* Author : Dilbert
*/

#define F_CPU 16000000UL
#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <string.h>
#include <stdlib.h>

#define USART_BAUDRATE 115200
#define UBRR_VALUE (int)round((((F_CPU/(USART_BAUDRATE*8UL)))-1))

// Global Variables
volatile uint8_t tempL = 0;
volatile uint8_t tempH = 0;
volatile float tempOut = 0;
char TEMP[50];
char WIFI[] = "";
char PASSWORD[] = "";
```

```c
// Function definitions
void InitPort();

void InitADC();
void SetADCChannel(uint8_t ADCchannel);
void StartADC(void);
void DisableADC(void);

void USART0Init(void);
void InitTimer0(void);
int USART0SendByte(char u8Data,FILE *stream);
void StartTimer0(void);
void StopTimer(void);
void check_OK(void);
void UART_sendString(char * AT);
char * UART_ReciveString(void);

ISR(ADC_vect);

// Set Stream Pointer
FILE usart0_str = FDEV_SETUP_STREAM(USART0SendByte, NULL, _FDEV_SETUP_WRITE);

//------------------------------------------------------------------------------
//    **********MAIN**********
//------------------------------------------------------------------------------
int main(void){
    // Initialize USART0
    USART0Init();
    // Initialize ports
    InitPort();
    // Assign our stream to standard I/O streams
    stdout=&usart0_str;
    // Initialize ADC
    InitADC();
    // Select ADC channel
    SetADCChannel(4);
    // Initialize timer0
    InitTimer0();
    // Start timer 0
    StartTimer0();
    // Start conversion
    StartADC();
    // Enable global interrupts
    sei();
    UART_sendString("AT\r\n");
    _delay_ms(20);
    // Check if ok returned
    check_OK();
    _delay_ms(20);
    // Select WIFI mode
    UART_sendString("AT+CWMODE=1\r\n");
    _delay_ms(20);
    // Connect to local WIFI
    UART_sendString("AT+CWJAP=\"NoMore\",\"Ch@ng3l!ng\"\r\n");
    _delay_ms(40);
    // Enable connection
    UART_sendString("AT+CIPMUX=0\r\n");
    _delay_ms(40);
    while (1){
        // 15 Second Delay between displaying temp
        _delay_ms(15000);

        // Start a connection as client to Thingspeak

UART_sendString("AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n");//"AT+CIPSTART=\"TCP\",\"184.106
.153.149\",80\r\n"
        _delay_ms(40);
        // Specify the size of the data
        UART_sendString("AT+CIPSEND=51\r\n");
        _delay_ms(40);
        // Temp set up
        tempOut = ((tempOut * 0.488));
        // AT data send command set up
        snprintf(TEMP, sizeof(TEMP), "GET /update?key=XUHJJ4KV38XFPTT7&field1=%2f", tempOut);
        // Send temperature data
        UART_sendString(TEMP);
        _delay_ms(20);
        UART_sendString("\r\n\r\n");
        }
}

//-----------------------------------------------------------------------------
//    *********Port Initialization function*********
//-----------------------------------------------------------------------------
void InitPort(){
    // Set pin C4 as an input pin
    DDRC |= (0<<PINC4);
    // Ensure pin C4 pull up resistor is off
    PORTC = (0<<PINC4);
    // Turn off Digital logic on pin C4
    DIDR0 |= (1<<ADC4D);
}

//-----------------------------------------------------------------------------
//    *********ADC Initialization function*********
//-----------------------------------------------------------------------------
void InitADC(){
    // Select Vref=Avcc and set (left = ADLAR = 1)(right = ADLAR = 0) justified result
    ADMUX |=(1<<REFS0)|(0<<ADLAR)|(0x04); // (1<< 2 or 3?)
    // Set prescaller to 32, enable auto triggering, enable ADC interrupt
    //  and enable ADC
    ADCSRA |=(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE)|(1<<ADIE)|(1<<ADEN);
    // Set ADC trigger source - Timer0 compare match A
    ADCSRB |=(1<<ADTS1)|(1<<ADTS0);
}

//-----------------------------------------------------------------------------
//    *********ADC PORT SELECT function*********
//-----------------------------------------------------------------------------
void SetADCChannel(uint8_t ADCchannel){
    // Select ADC channel with safety mask
    ADMUX = (ADMUX & 0xF0)|(ADCchannel & 0x0F);
}

//-----------------------------------------------------------------------------
```

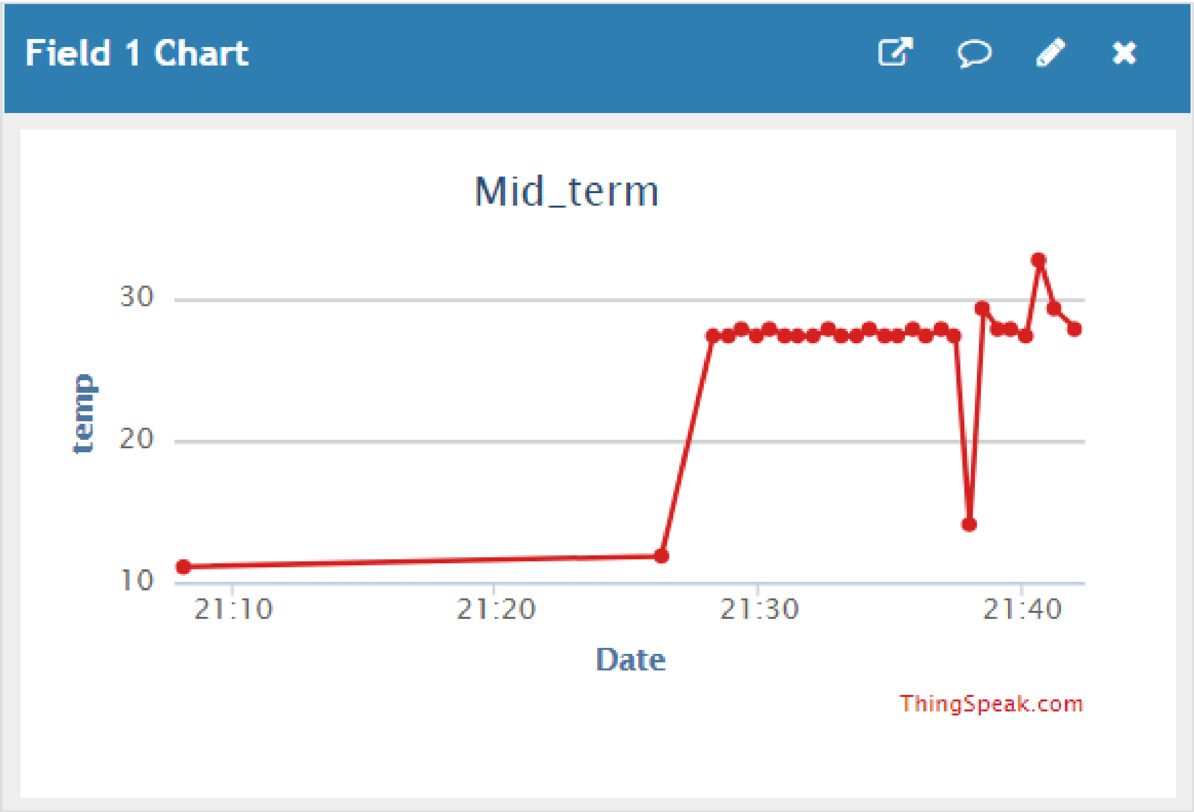
```
//    *********ADC start function*********
//-----------------------------------------------------------------------------
void StartADC(void){
    ADCSRA |=(1<<ADSC);
}

//-----------------------------------------------------------------------------
//    *********ADC end function*********
//-----------------------------------------------------------------------------
void DisableADC(void){
    ADCSRA &= ~((1<<ADEN)|(1<<ADIE));
}

//-----------------------------------------------------------------------------
//    *********USART Initialization function*********
//-----------------------------------------------------------------------------
void USART0Init(void){
    // Set baud rate
    UBRR0H = (uint8_t)(UBRR_VALUE>>8);
    UBRR0L = (uint8_t)UBRR_VALUE;
    // Set frame format to 8 data bits no parity, 1 stop bit
    UCSR0C |= (1<<UCSZ01)|(UCSZ00);
    // Enable transmission and reception
    UCSR0B |= (1<<RXEN0) | (1<<TXEN0);
    // Run double speed
    UCSR0A |= (1<<U2X0);
}

//-----------------------------------------------------------------------------
//    *********Timer0 Initialization function*********
//-----------------------------------------------------------------------------
void InitTimer0(void){
    // Set Initial Timer value
    TCNT0 = 0;
    // Place TOP timer value to Output compare register
    OCR0A = 99;
    // Set CTC mode
    //  and make toggle PD6/OC0A pin on compare match
    TCCR0A |= (1<<COM0A0)|(1<<WGM01);
}

//-----------------------------------------------------------------------------
//    *********USART Send function*********
//-----------------------------------------------------------------------------
void UART_sendString(char * AT)
{
    volatile unsigned char len= 0;
    volatile unsigned char i;
    while(AT[len] != 0)
    {
        len++;
    }
    for(i= 0x00; i< len; i++){
        // Wait for the transmitter to finish
        while(!(UCSR0A & (1<<UDRE0)));
        UDR0 = AT[i];
    }
}
```

```
//-----------------------------------------------------------------------------
//    *********USART Send function*********
//-----------------------------------------------------------------------------
int USART0SendByte(char u8Data,FILE *stream){
    // Wait while previous byte is completed
    while(!(UCSR0A & (1<<UDRE0))){};
    // Transmit Data
    UDR0 = u8Data;
    return 0;
}

//-----------------------------------------------------------------------------
//    *********Timer start function*********
//-----------------------------------------------------------------------------
void StartTimer0(void){
    // Set prescaller 8 and start timer
    TCCR0B |= (1<<CS01);
}

void check_OK(void){
    uint8_t len = 0;
    char *returned_str = UART_ReciveString();
    while(returned_str[0] == ’\0’){
        returned_str = UART_ReciveString();
    }
    len = strlen(returned_str);
    if(len > 3){ // error
        while(1);
        UART_sendString(returned_str);
        _delay_ms(1000);
        UART_sendString(″AT\r\n″);
        _delay_ms(1000);
    }
    if(len < 3){
        UART_sendString(returned_str);
        _delay_ms(1000);
    }
}

char * UART_ReciveString(void){
    char *str = (char *) (sizeof(char)*20);
    volatile uint8_t i = 0;
    do{
        // Wait for the transmitter to finish
        while(!(UCSR0A & (1<<UDRE0)));
        str[i] = UDR0;
        i++;
        } while(!RXC0);
    return str;
}

//-----------------------------------------------------------------------------
//    *********Timer end function*********
//-----------------------------------------------------------------------------
void StopTimer(void){
    TCCR0B &= ~(1<<CS01);
    TIMSK0 &= ~(1<<OCIE0A);
}

//-----------------------------------------------------------------------------
//    *********ADC conversion complete ISR*********
//-----------------------------------------------------------------------------
ISR(ADC_vect){
    // clear timer compare match flag
    TIFR0 = (1<<OCF0A);
    // Toggle pin PD2 to track the end of ADC conversion
    PIND = (1<<PD2);
    tempL = ADCL;
    tempH = ADCH;
    tempOut = ((tempH << 8)|(tempL));
}
```

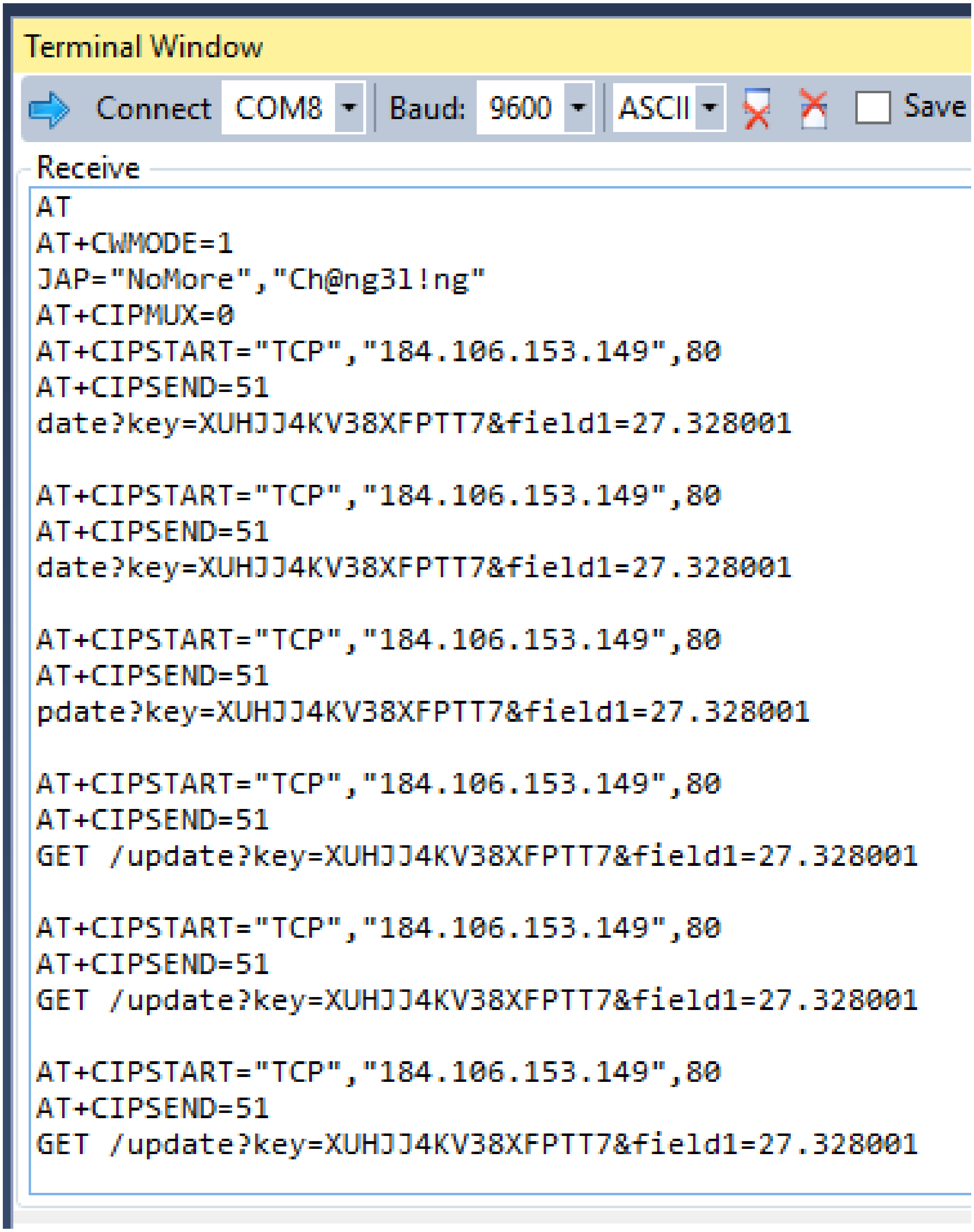
4. SCHEMATICS



5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



COM8 Baud: 9600 ASCII

```
AT
AT+CWMODE=1
JAP="NoMore","Ch@ng31!ng"
AT+CIPMUX=0
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
date?key=XUHJJ4KV38XFPTT7&field1=27.328001

AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
date?key=XUHJJ4KV38XFPTT7&field1=27.328001

AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
pdate?key=XUHJJ4KV38XFPTT7&field1=27.328001

AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
GET /update?key=XUHJJ4KV38XFPTT7&field1=27.328001

AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
GET /update?key=XUHJJ4KV38XFPTT7&field1=27.328001

AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=51
GET /update?key=XUHJJ4KV38XFPTT7&field1=27.328001
```

7. VIDEO LINKS OF EACH DEMO

https://youtu.be/e0PbBv4xrsc

8. GITHUB LINK OF THIS DA

https://github.com/Dil-bert/Alabaster/tree/master/Midterms/Mid_Term_I

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

"This assignment submission is my own, original work".
Dillon Archibald