CPE301 – SPRING 2019

MIDTERM 1

Student Name: Armon Latifi
Student #: 2000698173
Student Email: latifa1@unlv.nevada.edu
Primary Github address: https://github.com/armonlatifi
Directory: https://github.com/armonlatifi/sub_da/tree/master/MIDTERM%201

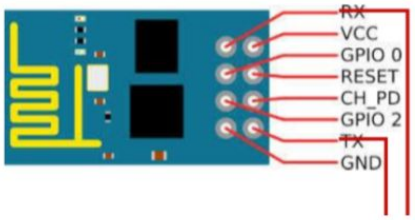
Submit the following for all Labs:

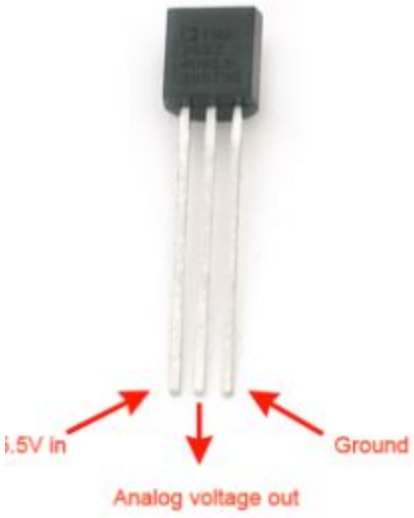
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder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

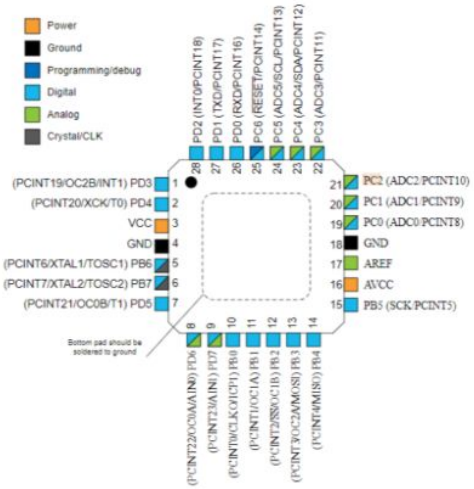
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:

- Assembler
- Simulator
- Debugger
- Breadboard
- Atmega328P
- Wires (male/female)
- Esp-01/firmware
- esp-01 adapter
- Atmel Studio
- Thinkspeak
- FTDI
- LM34







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

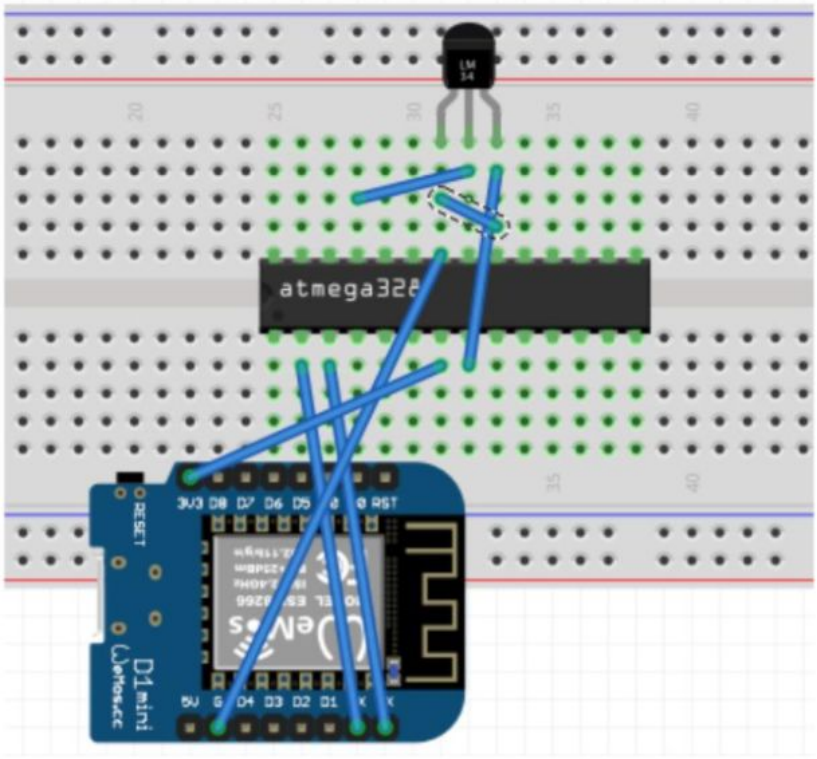
```
//set baud rate to 9600
//set clock rate at 16 MHz
//set the UBRR number

#define BAUD 9600
#define F_CPU 16000000UL
#define MYUBRR F_CPU/16/BAUD-1
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include <avr/interrupt.h>

void read_adc(void);
void adc_init(void);
void USART_init( unsigned int ubrr );
void USART_tx_string(char *data);

volatile unsigned int r_value;
char outs[20];

void usart_send( unsigned char ascii)
{
	while(!(UCSR0A & (1<<UDRE0)));
	UDR0 = ascii;
}


unsigned char usart_receive(void)
{
	while (!(UCSR0A & (1<<RXCIE0)));
	return UDR0;
}


void send_AT( unsigned char message[])
{
	unsigned char i=0;
	while(message[i] != '\0')
	{
		usart_send(message[i]);
		i++;
	}
}


int main(void) {

	unsigned char CIPMUX[] = "AT+CIPMUX=1\r\n";
	unsigned char CIPSTART[] = "AT+CIPSTART=0,\"TCP\",\"api.thingspeak.com\",80\r\n";
	unsigned char AT[] = "AT\r\n";
	unsigned char CIPSEND[] = "AT+CIPSEND=0,110\r\n";
	unsigned char GET_DATA[] = "GET
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=SBYXUXDC8TLMA50V\r\n";
	unsigned char SEND_DATA[] = "GET
https://api.thingspeak.com/update?api_key=RLIBH668P4MZTRB6=50\r\n";
```

```
	//initialize and wait
	USART_init(MYUBRR);
	USART_tx_string("Connected!\r\n");
	_delay_ms(325);
	send_AT(AT);
	_delay_ms(2000);
	send_AT(CIPMUX);
	_delay_ms(2000);
	send_AT(CIPSTART);
	_delay_ms(2000);
	send_AT(GET_DATA);
	_delay_ms(2000);
	send_AT(SEND_DATA);

	adc_init();

	//interrupt
	sei();

	while(1)
	{
	}
}

void adc_init(void)
{
	/** Setup & enable ADC **/
	ADMUX = (0<<REFS1)| // Reference Selection Bits

	(1<<REFS0)| // AVcc - external cap at AREF
	(0<<ADLAR)| // ADC Left Adjust Result
	(0<<MUX2)| // ANalog Channel Selection Bits
	(1<<MUX1)| // ADC2 (PC2 PIN25)
	(0<<MUX0);

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

	(0<<ADSC)| //Start Conversion
	(0<<ADATE)| //Auto Trigger Enable
	(0<<ADIF)| //Interrupt Flag
	(0<<ADIE)| //Interrupt Enable
	(1<<ADPS2)| //Prescaler Select Bits
	(0<<ADPS1)|
	(1<<ADPS0);

	// Timer/Counter1 Interrupt Mask Register

	TIMSK1 |= (1<<TOIE1); // enable overflow interrupt
	TCCR1B |= (1<<CS12)|(1<<CS10); // native clock
	TCNT1 = 49911;		//set Timer counter

}

/* READ PINS */
void read_adc(void) {
	unsigned char i =4;
```

```
	r_value = 0;
	while (i--) {
		ADCSRA |= (1<<ADSC);
		while(ADCSRA & (1<<ADSC));
		r_value+= ADC;
		_delay_ms(50);
	}
	//take average
	r_value = r_value / 8;
	r_value = r_value /2;
}

/* INIT USART */
void USART_init( unsigned int ubrr ) {
	UBRR0H = (unsigned char)(ubrr>>8);
	UBRR0L = (unsigned char)ubrr;
	UCSR0B = (1 << TXEN0);
	UCSR0C = (3 << UCSZ00);
}

void USART_tx_string( char *data ) {
	while ((*data != '\0')) {
		while (!(UCSR0A & (1 <<UDRE0)));
		UDR0 = *data;
		data++;
	}
}

ISR(TIMER1_OVF_vect)
{
	read_adc();
	snprintf(outs,sizeof(outs),"%3d\r\n", r_value);
	USART_tx_string(outs);
	//reset
	TCNT1 = 49911;
}
```

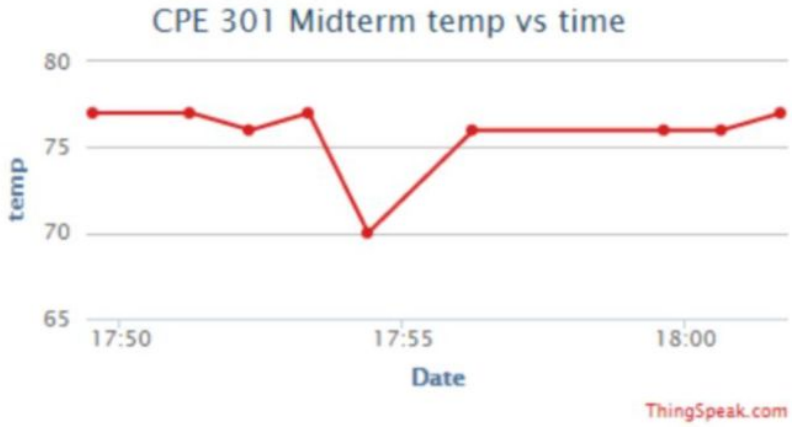
3. SCHEMATICS

Use fritzing.org

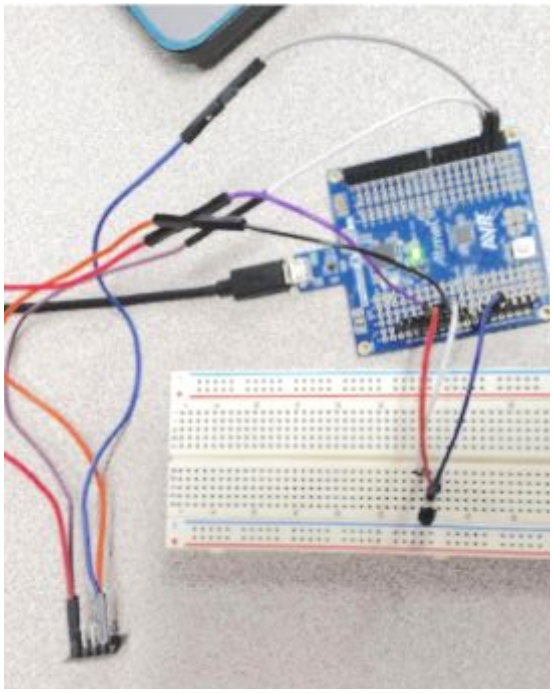


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

```
GET https://api.thingspeak.com/apps/thinghttp/send_request?api_key=SBYXUXDC8TLMA50
GET https://api.thingspeak.com/update?api_key=RLIBH668P4MZTRB6=50
 95
 89
 86
```



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. GITHUB LINK OF THIS DA

https://github.com/armonlatifi/sub_da/tree/master/MIDTERM%201

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

"*This assignment submission is my own, original work*".
Armon Latifi