Elizabeth Heider

CPE301 – SPRING 2018

Midterm 1

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
1	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
2.	INITIAL CODE OF TASK 1/A		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
4.	SCHEMATICS		
5.	SCREENSHOTS OF EACH TASK OUTPUT		
5.	SCREENSHOT OF EACH DEMO		
6.	VIDEO LINKS OF EACH DEMO		
7.	GOOGLECODE LINK OF THE DA		

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmega328P
FTDI Chip
LM34 Temp Sensor
ESP 8266
Attempted to use NodeMCU Chip - It started smoking

2. INITIAL/DEVELOPED CODE OF TASK 1/A

Transmit data using USART with the ESP chip – display results on Thingspeak (unsuccessful).

```
#define F_CPU 16000000UL

#include <stdlib.h>
#include <avr/io.h>
#include <stdint.h>
#include <util/delay.h>
#include <avr/interrupt.h>


#define FOSC 16000000                           // Clock speed 16Mhz
#define BAUD 9600                               // BAUD Rate Defined
#define MYUBRR FOSC/8/BAUD-1


volatile unsigned char AT[] = "AT\r\n";
volatile unsigned char CIPMUX[] = "AT+CIPMUX=0\r\n";
volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n";
//ip of thingspeak
volatile unsigned char SEND_DATA[] = "GET /update?key=RLJISPGVOR0OD77R&field1="; //update
thingspeak

volatile unsigned char CIPSIZE[] = "AT+CIPSEND=45\r\n"; //send data
volatile unsigned char CWMODE[] = "AT+CWMODE=3\r\n"; //wifi mode
volatile unsigned char CONNECTWIFI[] = "AT+CWJAP=\"Liz\",\"*********\"\r\n"; //connect to
ap
volatile unsigned char FIRMWARE[] = "AT+GMR\r\n";
volatile unsigned char BREAK[] = "\r\n\r\n";

//global variables
volatile uint8_t adc_val; // Value read from Temperature Sensor in ADC
volatile unsigned char temp[5];

//prototypes
void init_USART(); // Initialize USART function
void send_AT(volatile unsigned char AT[]);// Send commands

int main(void)
{
	ADMUX = 0; // read from ADC 0
	ADMUX |= (1 << REFS0); // use AVcc as the reference
	ADMUX |= (1 << ADLAR); // Right adjust for 8 bit resolution

	ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // 128 as prescalar
	ADCSRA |= (1 << ADATE); // Set ADC Auto Trigger Enable
	ADCSRB = 0;
	ADCSRA |= (1 << ADEN); // Enable ADC
	ADCSRA |= (1 << ADIE); // Enable Interrupts
	ADCSRA |= (1 << ADSC); // ADC Conversion

	init_USART();

	_delay_ms(1500);
	send_AT(AT); //at
```

```
	_delay_ms(1500);
	send_AT(FIRMWARE); //firmware
	_delay_ms(1500);
	send_AT(CWMODE); //wifi mode
	_delay_ms(1500);
	send_AT(CONNECTWIFI); //connect with WiFi
	_delay_ms(5000);
	send_AT(CIPMUX); //enable

	sei();
	while (1)
	{
		_delay_ms(500);
		send_AT(CIPSTART); // start connection
		_delay_ms(500);
		send_AT(CIPSIZE); // size
		_delay_ms(500);
		send_AT(SEND_DATA);
		send_AT(temp); //temperature
		send_AT(BREAK);
	}
	return 0;
}

void init_USART() {
	// Setting BAUD rate
	UBRR0H = ((MYUBRR) >> 8);
	UBRR0L = MYUBRR;

	UCSR0A |= (1<< U2X0);
	UCSR0B |= (1 << TXEN0); // Enable transmitter
	UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // Set frame: 8data, 1 stp
}

// Interrupt subroutine for ADC value
ISR(ADC_vect) {
	unsigned char i;
	char tmptemp[5];

	adc_val = (ADCH << 1);
	itoa(adc_val, tmptemp, 10);
	for(i = 0 ; i < 5 ; i++)
	temp[i] = tmptemp[i];
}

void send_AT(volatile unsigned char AT[]) {
	volatile unsigned char a;
	volatile unsigned char length = 0;

	while(AT[length] != 0)
	length++; // find length

	for(a = 0 ; a < length ; a++)
	{
		while(!(UCSR0A & (1<<UDRE0)));
		UDR0 = AT[a];
	}
}
```

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

"*This assignment submission is my own, original work*".
Elizabeth Heider