Eilat Avidan

CPE301 – SPRING 2018

Design Assignment (MID-TERM)

**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
* Mini USB to USB cable
* Breadboard
* ESP8266 chip
* Resistors (20 ohms and 40 ohms)

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

#define *F\_CPU* 8000000UL

#define UBRR\_115200 3 // for 8Mhz with 8.5% error

#define Domain "api.thingspeak.com"

#define API\_Write\_Key "K6HINW5F75HTUPI8"

#define Channel\_ID "467279"

#define SSID ""

#define Password ""

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <stdint.h>

// Function Declarations

void read\_adc(void);

void adc\_init(void);

void USART\_init( unsigned int ubrr );

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

volatile unsigned int adc\_temp;

char outs[30];

int main(void)

{

adc\_init(); //Initialize the ADC (Analog / Digital Converter)

USART\_init(UBRR\_115200); //Initialize the USART (RS232 interface)

*\_delay\_ms*(125); //wait a bit

*snprintf*(outs,sizeof(outs),"AT\r\n");

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

*snprintf*(outs,sizeof(outs),"AT+CWMODE=3\r\n"); //wifi mode

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

*snprintf*(outs,sizeof(outs),"AT+CIPMUX=0\r\n"); //single connection

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

*snprintf*(outs,sizeof(outs),"AT+CIPMODE=0\r\n"); //normal mode

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

*snprintf*(outs,sizeof(outs),"AT+CWJAP=\"%s\",\"%s\"\r\n", SSID, Password); //connect to wifi network

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

*snprintf*(outs,sizeof(outs),"AT+CPISTART=\"TCP\",\"%s\",80\r\n",Domain); //connect to thingspeak

USART\_tx\_string(outs);

*\_delay\_ms*(2000);

while(1)

{

read\_adc();

*snprintf*(outs,sizeof(outs),"GET /update?api\_key=%s&field1=1\r\n",API\_Write\_Key); //send value

USART\_tx\_string(outs);

*\_delay\_ms*(150);

*snprintf*(outs,sizeof(outs),"AT+CISEND=%3d\r\n",adc\_temp);

USART\_tx\_string(outs);

*\_delay\_ms*(15000); //wait 15 seconds

}

}

void adc\_init(void)

{

/\*\* Setup and enable ADC \*\*/

ADMUX = 0; //select ADC0 Pin as input

ADMUX = (0<<REFS1)| //Reference Selection Bits

(1<<REFS0)| //AVcc - external cap at AREF

(1<<ADLAR); //ADC left Adjust Result

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

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

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

(0<<ADIF)| //ADC Interrupt Flag

(0<<ADIE)| //ADC Interrupt Enable

(1<<ADPS2)| //ADC Pre-scaler of 64

(1<<ADPS1)|

(0<<ADPS0);

}

/\* READ ADC PINS\*/

void read\_adc(void)

{

unsigned char i = 4; //set for 4 ADC reads

adc\_temp = 0; //initialize temp to 0

while (i--)

{

ADCSRA |= (1<<ADSC); //start the conversion

while((ADCSRA & (1<<ADIF)) == 0); //wait for conversion to finish

adc\_temp += ADCH\*2; //get temp value

*\_delay\_ms*(50); //wait a bit

}

adc\_temp = adc\_temp / 4; // Average a few samples

}

/\* INIT USART (RS-232) \*/

void USART\_init( unsigned int ubrr )

{

UBRR0H = (unsigned char)(ubrr>>8); //set baud rate

UBRR0L = (unsigned char)ubrr;

UCSR0B = (1 << TXEN0) | (1 <<RXEN0); // Enable receiver, transmitter

UCSR0C = (1 << UCSZ00) | (1 << UCSZ01); //asynchronous 8-bit data 1 stop bit

}

/\* SEND A STRING TO THE RS-232\*/

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

{

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

{

while (!(UCSR0A & (1 <<UDRE0))); //wait for the transmit buffer to empty

UDR0 = \*data; //put the data into the empty buffer, which sends the data

*\_delay\_ms*(125); // wait a bit

data++;

}

}

1. **Flow chart**

Display values in Putty using WIFI (AT commends)

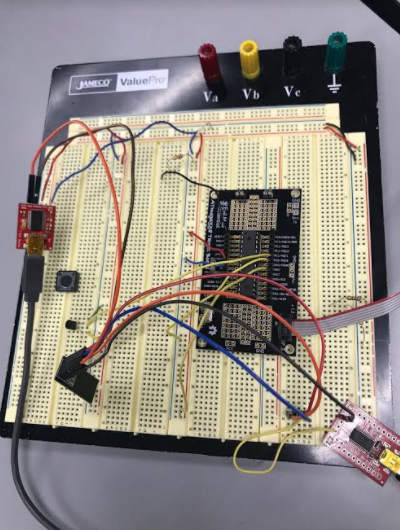
ESP8266 (send value through AT commends)

FTDI (3.3V)

LM34 (temp sensor)

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

I could not get the terminal to display the values. I tried even to connect the esp8266 directly to the FTDI chip to display values and it still did not work.

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
2. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/b5qD7vp0Rd0>

1. **GITHUB LINK OF THIS DA**

[git@github.com:EilatAvidan/microcon.git](mailto:git@github.com:EilatAvidan/microcon.git)

**Student Academic Misconduct Policy**

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

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

Eilat Avidan