CPE301 – SPRING 2019

MIDTERM 1

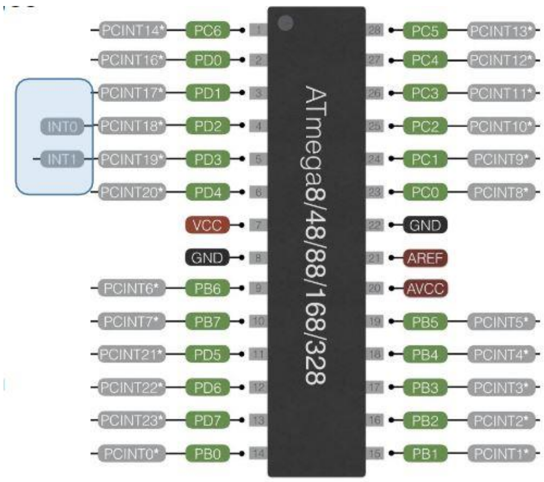
Student Name: Jett Guerrero
Student #: 5002377713
Student Email: guerrj1@unlv.nevada.edu
Primary Github address: https://github.com/guerrj1
Directory: https://github.com/guerrj1/Submission_Midterms/tree/master/Midterm1

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

- Atmega328P
- FTDI Basic
- ESP-01
- LM34
- Breadboard
- Micro USB Cable
- Mini USB Cable
- Male to male wires
- Female to male wires



Pins on the Atmega328p that were used are PD0, PD1, GND, 5v and 3.3v

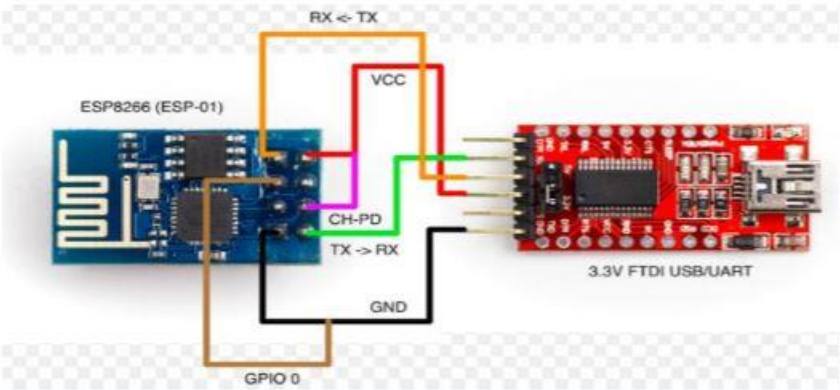
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1

```
//Midterm1

#define F_CPU 16000000UL
#define BAUD 115200
#define FOSC 16000000
#define UBRR_VALUE (FOSC/8/BAUD -1)  //USART Baud rate equation
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdint.h>

volatile uint8_t ADCvalue;              //Global variable, set to volatile if used with ISR
//AT array
volatile unsigned char ADCtemp[5];
volatile unsigned char CWMODE[] = "AT+CWMODE=3\r\n"; //Mode 3 for wifi mode
volatile unsigned char WIFI[] =   "AT+CWJAP=\"JIJJG\", \"xxxxxx\"\r\n"; //Connect to wifi with SSID and password
volatile unsigned char ENABLE[] = "AT+CIPMUX=0\r\n"; //enable single connection
volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n";
//Transmission control protocol, remote IP address, port 80
```

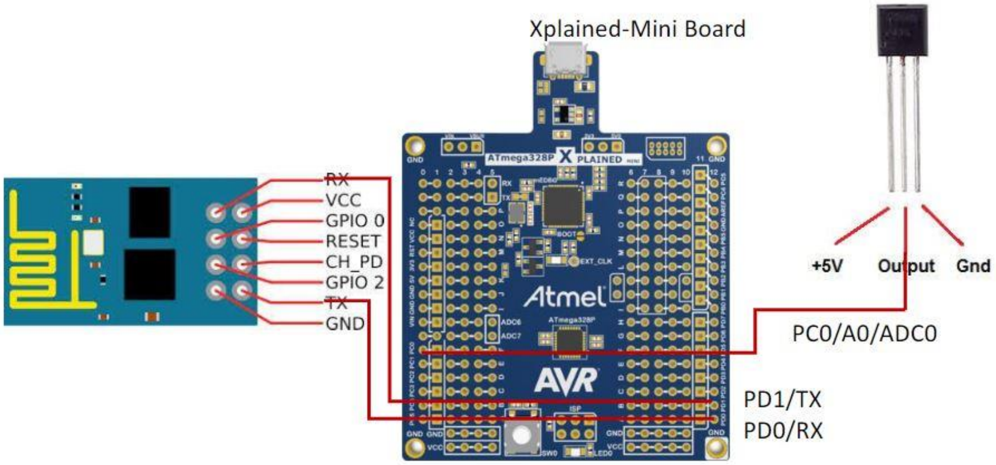
```
volatile unsigned char CIPSEND[] =  "AT+CIPSEND=45\r\n"; //45 is the length of the data to be
sent
volatile unsigned char SEND_DATA[] = "GET /update?key=ML87AXCZLB8AGULF&field1="; //Write API
key: ML87AXCZLB8AGULF from thingspeak
volatile unsigned char PAUSE_DELAY[] =    "\r\n\r\n";  //pause delay

//prototypes
void usart_init (void);  //usart function
void send_ATdata (volatile unsigned char c[]);  //sends the arrays to this function

int main(void)
{
        //ADC intializations
        ADMUX = 0;              //use ADC0
        ADMUX |= (1 << REFS0);  //use AVcc as the reference
        ADMUX |= (1 << ADLAR);  //Right adjust for 8 bit resolution
        ADCSRA |= (1 << ADATE); //Set ADC Auto Trigger Enable
        ADCSRB = 0;             //0 for free running mode
        ADCSRA |= (1 << ADEN);  //Enable the ADC
        ADCSRA |= (1 << ADIE);  //Enable Interrupts
        ADCSRA |= (1 << ADSC);  //Start the ADC conversion
        ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);  //set prescaler to 128

        usart_init();      //usart function call

        _delay_ms(1000);
        send_ATdata(CWMODE);
        _delay_ms(1000);
        send_ATdata(WIFI);
        _delay_ms(2000);
        send_ATdata(ENABLE);

        //while always
        while(1)
        {
                _delay_ms(1000);  //delay function
                send_ATdata(CIPSTART); //thingspeak connection in port 80
                _delay_ms(1000);  //delay function
                send_ATdata(CIPSEND); //data length
                _delay_ms(1000);  //delay function
                send_ATdata(SEND_DATA); //api key
                send_ATdata(ADCtemp);  //sends the temperature data from sensor
                send_ATdata(PAUSE_DELAY);  //delay for display output
        }
}

//interrupt subroutine
ISR(ADC_vect)
{
        volatile unsigned int j=0;
        char temp[5];

        ADCvalue = (ADCH<<1);  //shifts ADCH by 1 to the left
        itoa(ADCvalue, temp, 10);

        while(j<5)
        {
                ADCtemp[j] = temp[j]; //set temp as ADCtemp
                j++;  //increment the j temp
        }
}
```

```
void usart_init(void)  //usart initialization
{
                //set baud rate
                UBRR0H = ((UBRR_VALUE) >> 8); //shift by 8 bits
         UBRR0L = UBRR_VALUE; //set UBRRO low to the UBRR_VALUE

                UCSR0A |= (1 << U2X0); //USART transmission speed
                UCSR0B |= (1 << TXEN0); //enable transmission and reception
                UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); //set frame to 8 data bits, no parity,
1 stop bit

                sei();   //enable global interrupt
}

void send_ATdata(volatile unsigned char c[])
{
        //counter variable initializations
        volatile unsigned int i=0;
        volatile unsigned int j=0;

        j = 0; //initialize counter j to 0

        //while still not at the end
        while (c[j] != 0x00)
        {
                j++;   //increment j counter
        }

        //while i is less than j counter
        while (i<j)
         {
                 //wait whike prevoius byte is completed
                while(!(UCSR0A & (1 << UDRE0)));  //buffer is empty and can be written to when
UDREO is 1
                UDR0 = c[i]; //transmit data
                i++;  //increment i counter
         }
}
```

3. SCHEMATICS

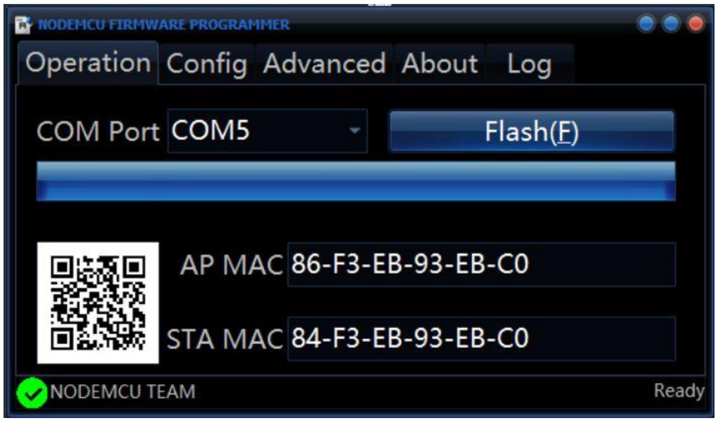


Schematic wire connection for flashing firmware into ESP-01



Schematic wire connection for connecting LM34 temperature sensor to the Atmega328p and ESP-01

4. SCREENSHOTS OF EACH TASK OUTPUT (NODEMCU FIRMWARE PROGRAMMER, ESPLORER, ATMEL STUDIO OUTPUT, & THINGSPEAK)

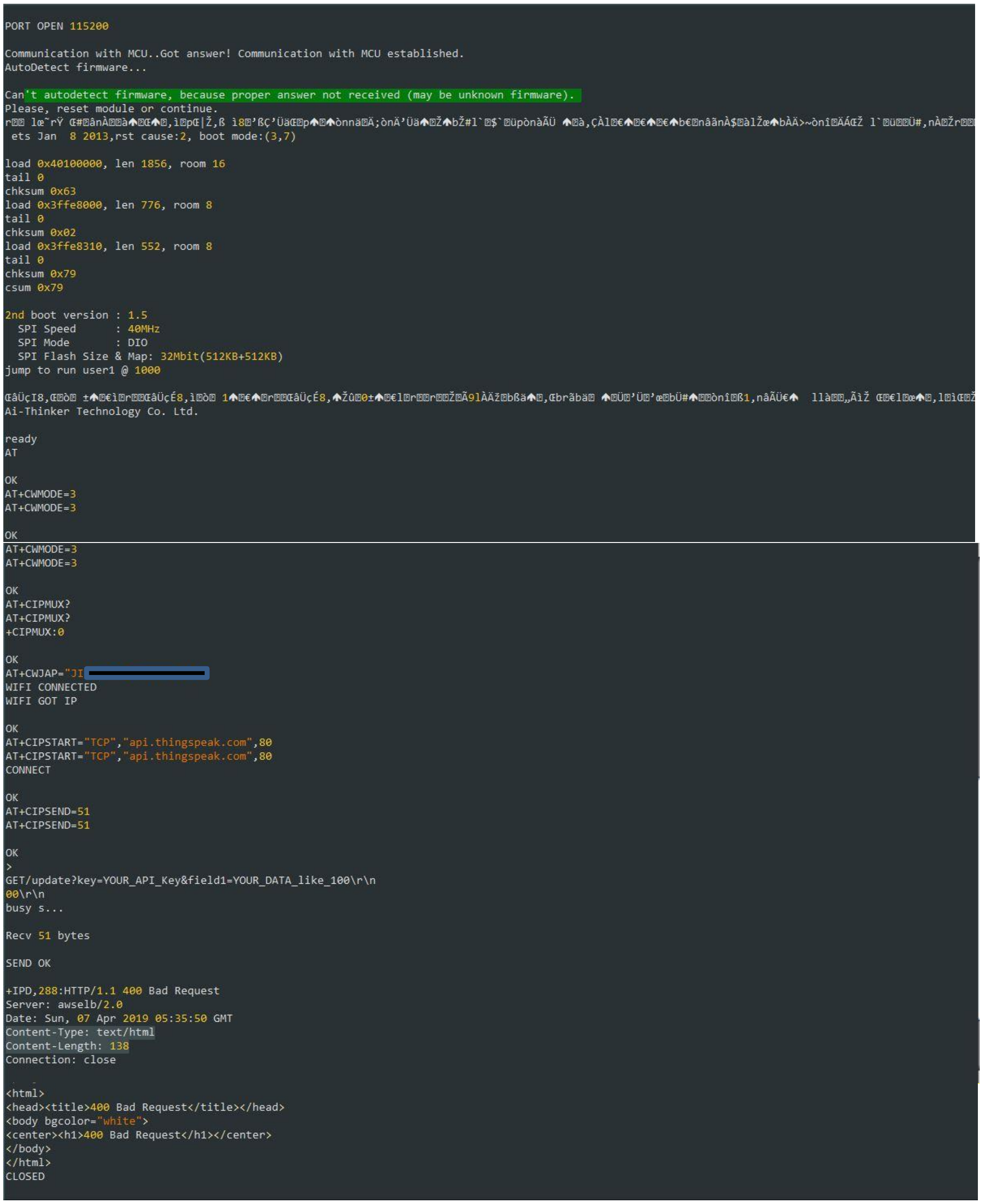


COM Port COM5

AP MAC 86-F3-EB-93-EB-C0

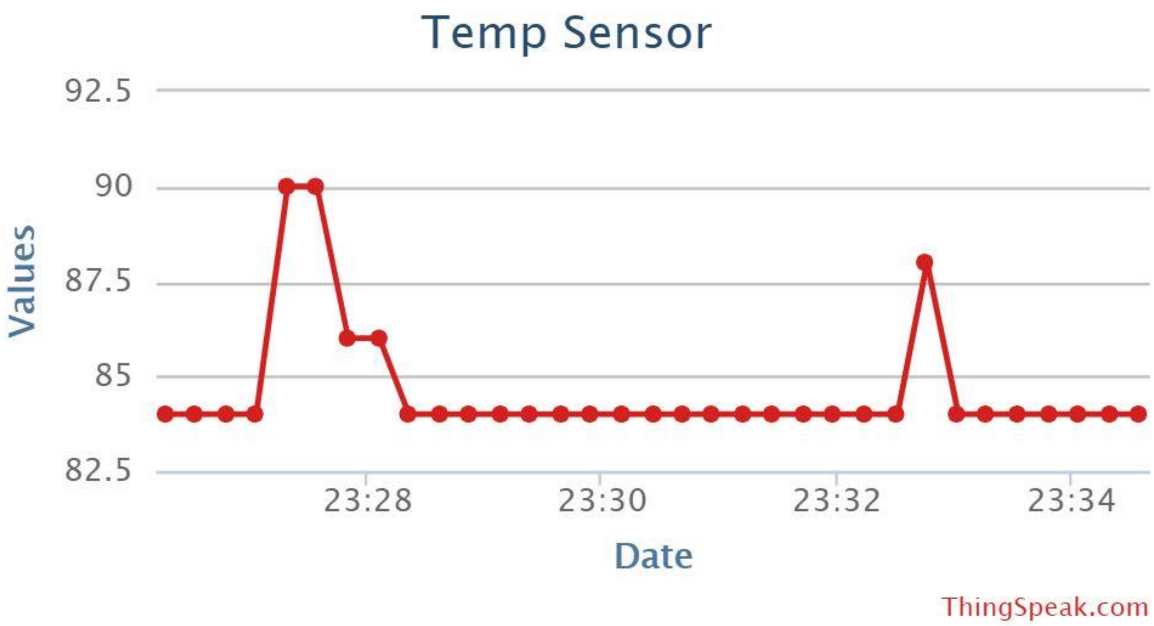
STA MAC 84-F3-EB-93-EB-C0

NodeMCU Firmware Programmer



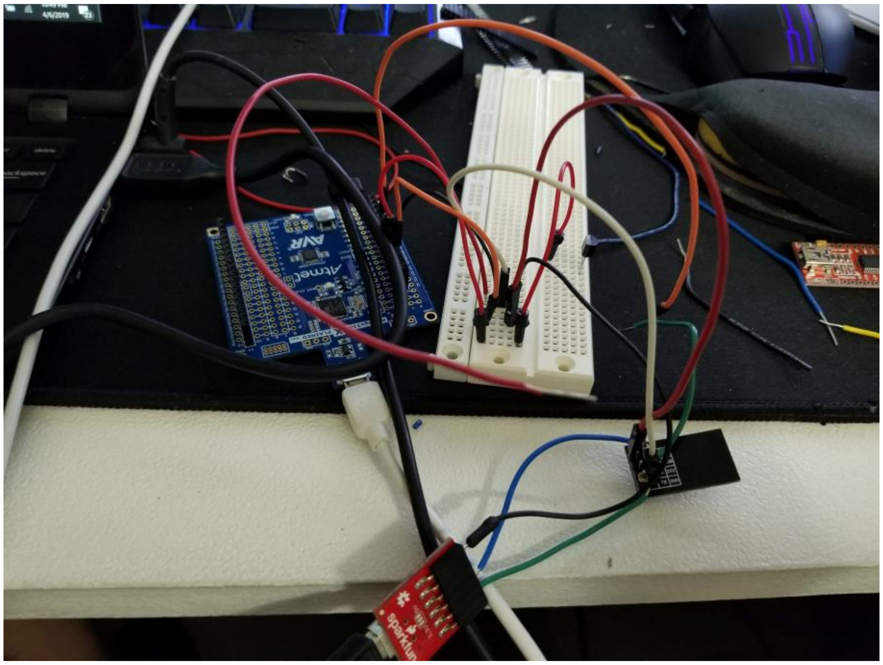
```
PORT OPEN 115200

Communication with MCU..Got answer! Communication with MCU established.
AutoDetect firmware...

Can't autodetect firmware, because proper answer not received (may be unknown firmware).
Please, reset module or continue.
r␦␦ lœˇrŸ Œ#␦ânÀ␦␦à⬆␦Œ⬆␦,ì␦pŒ|Ž,ß ì8␦’ßÇ’ÜäŒ␦p⬆␦⬆ònnä␦Ä;ònÄ’Üä⬆␦Ž⬆bŽ#l`␦$`␦üpònàÃÜ ⬆␦à,ÇÀl␦€⬆␦€⬆␦€⬆b€␦nâãnÀ$␦àlŽœ⬆bÀÄ>~ònî␦ÄÁŒŽ l`␦ü␦␦Ü#,nÀ␦Žr␦␦
 ets Jan  8 2013,rst cause:2, boot mode:(3,7)

load 0x40100000, len 1856, room 16
tail 0
chksum 0x63
load 0x3ffe8000, len 776, room 8
tail 0
chksum 0x02
load 0x3ffe8310, len 552, room 8
tail 0
chksum 0x79
csum 0x79

2nd boot version : 1.5
  SPI Speed      : 40MHz
  SPI Mode       : DIO
  SPI Flash Size & Map: 32Mbit(512KB+512KB)
jump to run user1 @ 1000

ŒâÜçI8,Œ␦ò␦ ±⬆␦€ì␦r␦␦ŒâÜçÉ8,ì␦ò␦ 1⬆␦€⬆␦r␦␦ŒâÜçÉ8,⬆Žû␦0±⬆␦€l␦r␦␦r␦␦Ž␦Ã9lÀÄž␦bßä⬆␦,Œbrãbä␦ ⬆␦Ü␦’Ü␦’œ␦bÜ#⬆␦␦ònî␦ß1,nâÃÜ€⬆  llà␦␦„ÃìŽ Œ␦€l␦œ⬆␦,l␦ìŒ␦Ž
Ai-Thinker Technology Co. Ltd.

ready
AT

OK
AT+CWMODE=3
AT+CWMODE=3

OK
AT+CWMODE=3
AT+CWMODE=3

OK
AT+CIPMUX?
AT+CIPMUX?
+CIPMUX:0

OK
AT+CWJAP="JI[redacted]
WIFI CONNECTED
WIFI GOT IP

OK
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSTART="TCP","api.thingspeak.com",80
CONNECT

OK
AT+CIPSEND=51
AT+CIPSEND=51

OK
>
GET/update?key=YOUR_API_Key&field1=YOUR_DATA_like_100\r\n
00\r\n
busy s...

Recv 51 bytes

SEND OK

+IPD,288:HTTP/1.1 400 Bad Request
Server: awselb/2.0
Date: Sun, 07 Apr 2019 05:35:50 GMT
Content-Type: text/html
Content-Length: 138
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
</body>
</html>
CLOSED
```

Espolrer AT Command Window

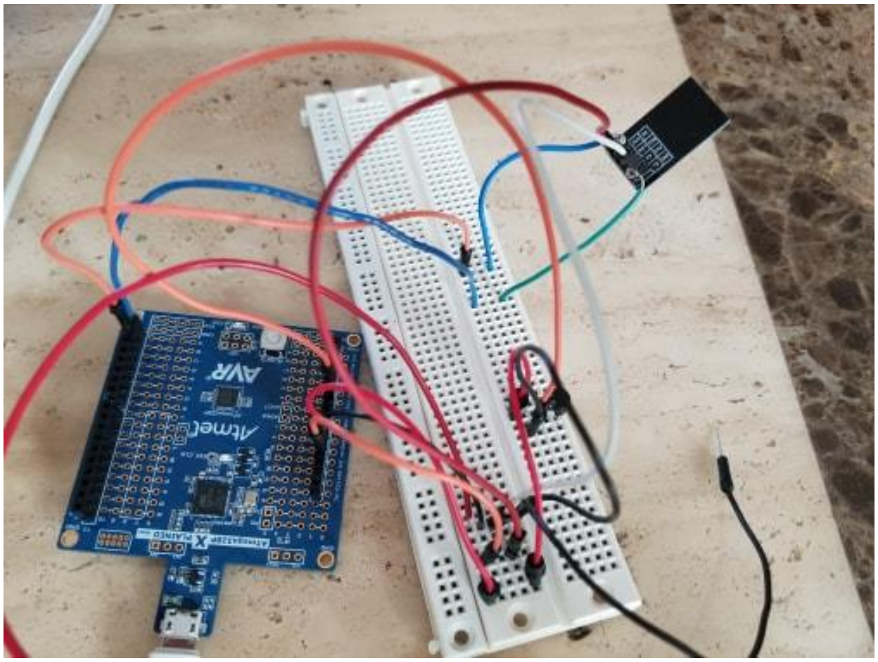


ThingSpeak Temperature Sensor Graph from LM34

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



Wiring of ESP-01 for firmware flashing


Wiring for ESP-01 and LM34 Temperature sensor to the Atmega328p

6. VIDEO LINKS OF EACH DEMO

https://youtu.be/rj8kNaYIeUw

7. GITHUB LINK OF THIS DA

https://github.com/guerrj1/Submission_Midterms/tree/master/Midterm1

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

"This assignment submission is my own, original work".
Jett Guerrero