CPE301 – SPRING 2025

Design Assignment 4

Student Name: Ryan Sewell
Student #: 8000473785
Student Email: sewelr2@unlv.nevada.edu
Primary Github address: https://github.com/sewelr2
Directory: DA-Submissions/DA4
Video Playlist:
https://youtube.com/playlist?list=PLt45mEFhRV6ffOYRcGHhoI5aDeP3Zgqt5&feature=shared

The goal of this assignment is to write, implement and demonstrate using Microchip Studio 7 a C code for the AVR ATMEGA328pb

- Read the ADC value from the POT connected to AC0/PC0. Keep displaying the voltage value UART terminal every 0.01 sec. The resolution of the oscilloscope should be 0.1V. Use Timer auto-trigger for this implementation.
- Using a GUI Python script display the ADC values as waveform (using tkinter).

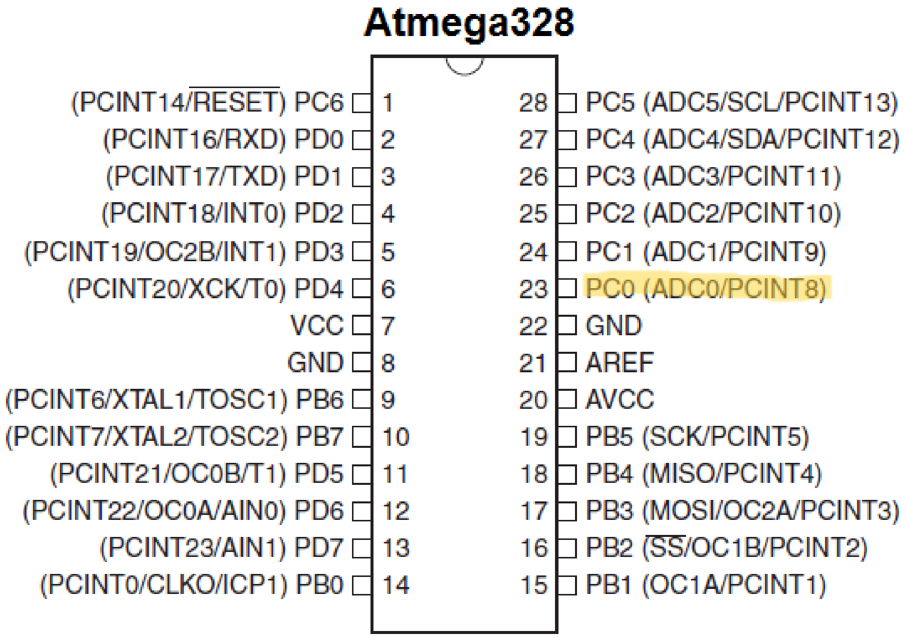
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Microchip Studio
- Assembler
- Simulator
- Debugger

Atmega328PB-Xmini PC

Multi-Function Shield
- Potentiometer

Tauno Serial Plotter



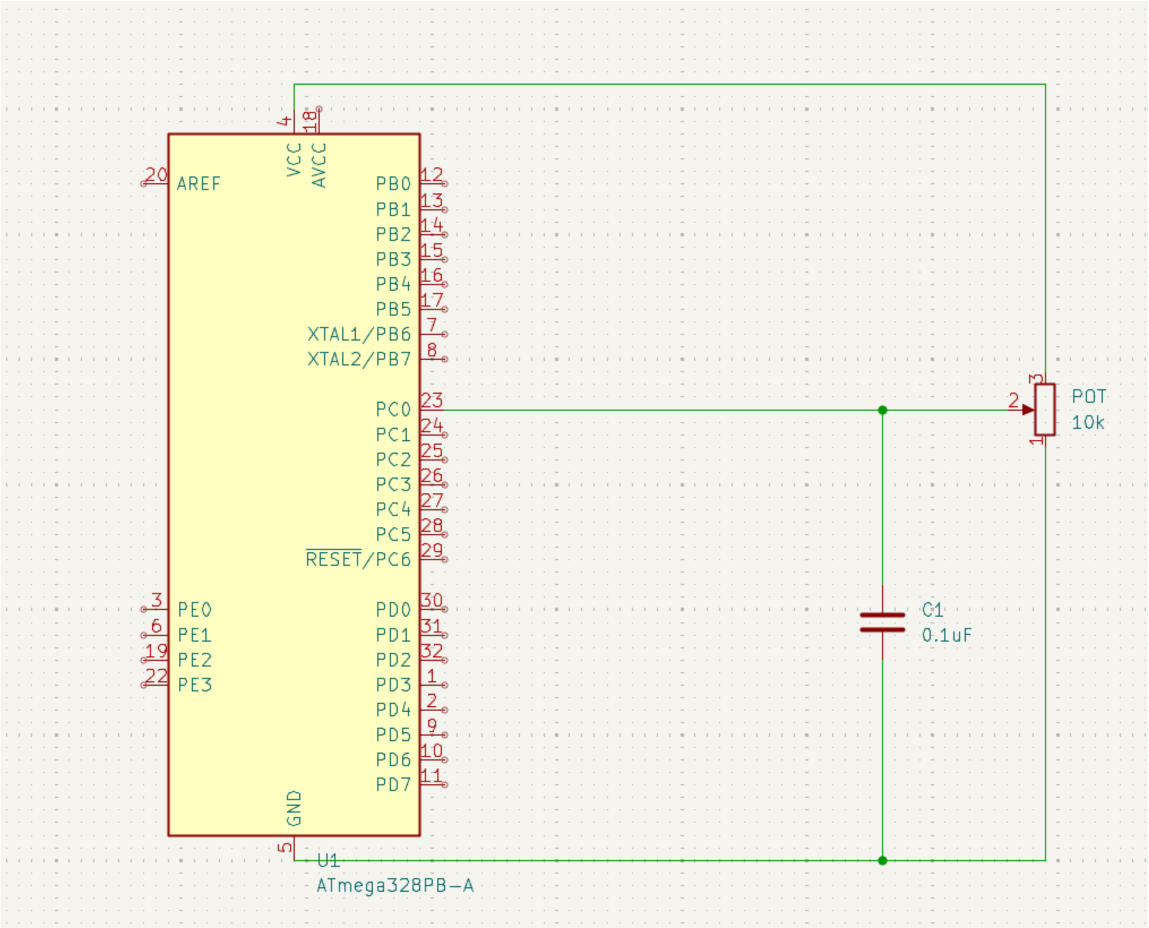
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>

#define F_CPU 16000000UL
#define BAUD 9600

#define BAUD_PRESCALER ((F_CPU / (16UL * BAUD)) - 1)

// Global variable to store the latest ADC reading
volatile uint16_t adc_value;

void USART_init()
{
        UBRR0H = (uint8_t)(BAUD_PRESCALER >> 8);
        UBRR0L = (uint8_t)(BAUD_PRESCALER);
        UCSR0B = (1 << RXEN0) | (1 << TXEN0);
        UCSR0C = (3 << UCSZ00);
}

void USART_send(unsigned char data)
{
        // Wait for empty transmit buffer
        while (!(UCSR0A & (1 << UDRE0)));
        UDR0 = data;
}

void USART_putstring(char* StringPtr)
{
        while(*StringPtr != 0x00)
        {
                USART_send(*StringPtr);
                StringPtr++;
        }
}

void adc_init(void){
        ADCSRA |= ((1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0));    //16Mhz/128 = 125Khz the ADC reference 
clock
        ADMUX |= (1<<REFS0);                //Voltage reference from Avcc (5v)
        ADCSRA |= (1<<ADEN);                //Turn on ADC
        ADCSRA |= (1<<ADSC);                //Do an initial conversion because this one is the 
slowest and to ensure that everything is up and running
}

void timer_init (void)
{
        TCCR1B |= 5; //(1 << CS12) | (1 << CS10); // Sets prescaler to 1024
        TIMSK1 = (1 << TOIE1); // Enables overflow flag
        TCNT1 = 49911; // 1 second delay = (0xFFFF) - TCNT = 65535 - 15624 = 49911
        sei();
}

int main (void)
{
        timer_init ();
        USART_init ();
        adc_init ();

        while (1)
        {
                // main loop
        }
        return 0;
}
```

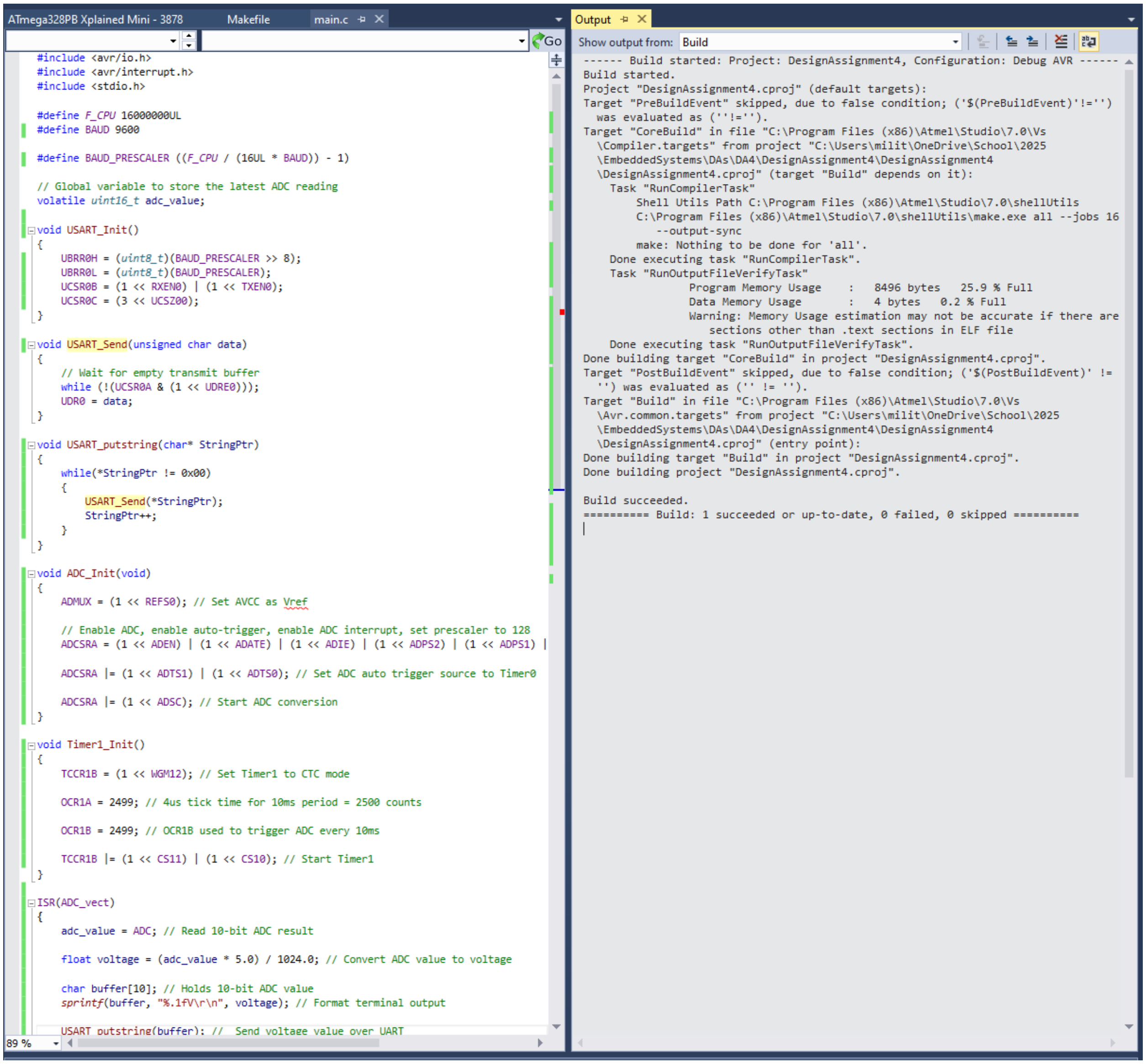
3. DEVELOPED/MODIFIED CODE OF TASK 2/A from TASK 1

```
void ADC_Init(void)
{
        ADMUX = (1 << REFS0); // Set AVCC as Vref

        // Enable ADC, enable auto-trigger, enable ADC interrupt, set prescaler to 128
        ADCSRA = (1 << ADEN) | (1 << ADATE) | (1 << ADIE) |  (1 << ADPS2) | (1 << ADPS1) | (1 <<
ADPS0);

        ADCSRA |= (1 << ADTS1) | (1 << ADTS0); // Set ADC auto trigger source to Timer0

        ADCSRA |= (1 << ADSC); // Start ADC conversion
}

void Timer1_Init()
{
        TCCR1B = (1 << WGM12); // Set Timer1 to CTC mode

        OCR1A = 2499; // 4us tick time for 10ms period = 2500 counts

        OCR1B = 2499; // OCR1B used to trigger ADC every 10ms

        TCCR1B |= (1 << CS11) | (1 << CS10); // Start Timer1
}

ISR(ADC_vect)
{
        adc_value = ADC; // Read 10-bit ADC result

        float voltage = (adc_value * 5.0) / 1024.0; // Convert ADC value to voltage

        char buffer[10]; // Holds 10-bit ADC value
        sprintf(buffer, "%.1fV\r\n", voltage); // Format terminal output

        USART_putstring(buffer); //  Send voltage value over UART
}

int main(void)
{
        USART_Init(BAUD_PRESCALER); // Initialize UART

        ADC_Init(); // Initialize ADC to read PC0

        Timer1_Init(); // Initialize Timer1

        sei(); // Enable global interrupts

        while (1)
        {
                // Wait for ADC ISR to trigger
        }

        return 0;
}
```

4. SCHEMATICS



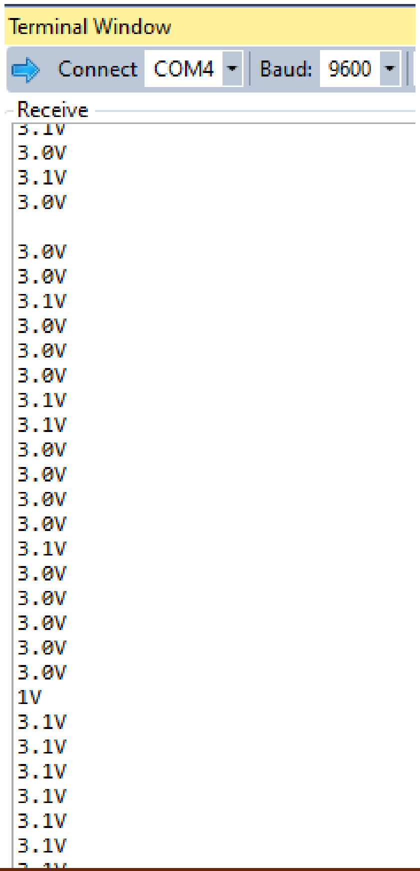
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



```c
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>

#define F_CPU 16000000UL
#define BAUD 9600

#define BAUD_PRESCALER ((F_CPU / (16UL * BAUD)) - 1)

// Global variable to store the latest ADC reading
volatile uint16_t adc_value;

void USART_Init()
{
    UBRR0H = (uint8_t)(BAUD_PRESCALER >> 8);
    UBRR0L = (uint8_t)(BAUD_PRESCALER);
    UCSR0B = (1 << RXEN0) | (1 << TXEN0);
    UCSR0C = (3 << UCSZ00);
}

void USART_Send(unsigned char data)
{
    // Wait for empty transmit buffer
    while (!(UCSR0A & (1 << UDRE0)));
    UDR0 = data;
}

void USART_putstring(char* StringPtr)
{
    while(*StringPtr != 0x00)
    {
        USART_Send(*StringPtr);
        StringPtr++;
    }
}

void ADC_Init(void)
{
    ADMUX = (1 << REFS0); // Set AVCC as Vref

    // Enable ADC, enable auto-trigger, enable ADC interrupt, set prescaler to 128
    ADCSRA = (1 << ADEN) | (1 << ADATE) | (1 << ADIE) | (1 << ADPS2) | (1 << ADPS1) |

    ADCSRA |= (1 << ADTS1) | (1 << ADTS0); // Set ADC auto trigger source to Timer0

    ADCSRA |= (1 << ADSC); // Start ADC conversion
}

void Timer1_Init()
{
    TCCR1B = (1 << WGM12); // Set Timer1 to CTC mode

    OCR1A = 2499; // 4us tick time for 10ms period = 2500 counts

    OCR1B = 2499; // OCR1B used to trigger ADC every 10ms

    TCCR1B |= (1 << CS11) | (1 << CS10); // Start Timer1
}

ISR(ADC_vect)
{
    adc_value = ADC; // Read 10-bit ADC result

    float voltage = (adc_value * 5.0) / 1024.0; // Convert ADC value to voltage

    char buffer[10]; // Holds 10-bit ADC value
    sprintf(buffer, "%.1fV\r\n", voltage); // Format terminal output

    USART_putstring(buffer); //  Send voltage value over UART
```

```
------ Build started: Project: DesignAssignment4, Configuration: Debug AVR ------
Build started.
Project "DesignAssignment4.cproj" (default targets):
Target "PreBuildEvent" skipped, due to false condition; ('$(PreBuildEvent)'!='')
  was evaluated as (''!='').
Target "CoreBuild" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs
  \Compiler.targets" from project "C:\Users\milit\OneDrive\School\2025
  \EmbeddedSystems\DAs\DA4\DesignAssignment4\DesignAssignment4
  \DesignAssignment4.cproj" (target "Build" depends on it):
    Task "RunCompilerTask"
        Shell Utils Path C:\Program Files (x86)\Atmel\Studio\7.0\shellUtils
        C:\Program Files (x86)\Atmel\Studio\7.0\shellUtils\make.exe all --jobs 16
           --output-sync
        make: Nothing to be done for 'all'.
    Done executing task "RunCompilerTask".
    Task "RunOutputFileVerifyTask"
                Program Memory Usage    :   8496 bytes   25.9 % Full
                Data Memory Usage       :   4 bytes   0.2 % Full
                Warning: Memory Usage estimation may not be accurate if there are
                   sections other than .text sections in ELF file
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "DesignAssignment4.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' !=
  '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs
  \Avr.common.targets" from project "C:\Users\milit\OneDrive\School\2025
  \EmbeddedSystems\DAs\DA4\DesignAssignment4\DesignAssignment4
  \DesignAssignment4.cproj" (entry point):
Done building target "Build" in project "DesignAssignment4.cproj".
Done building project "DesignAssignment4.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

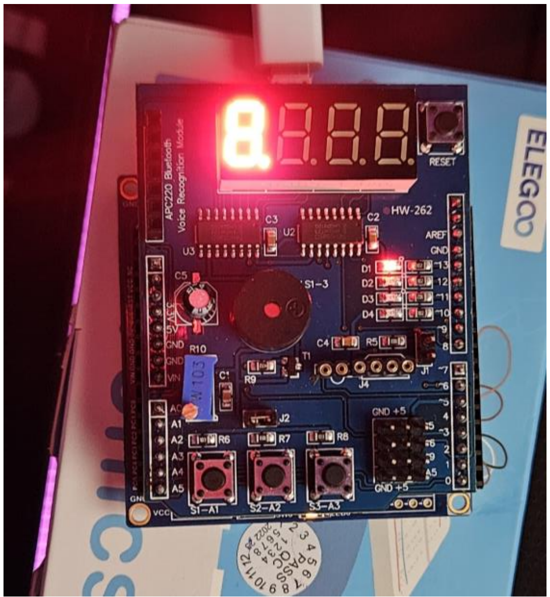


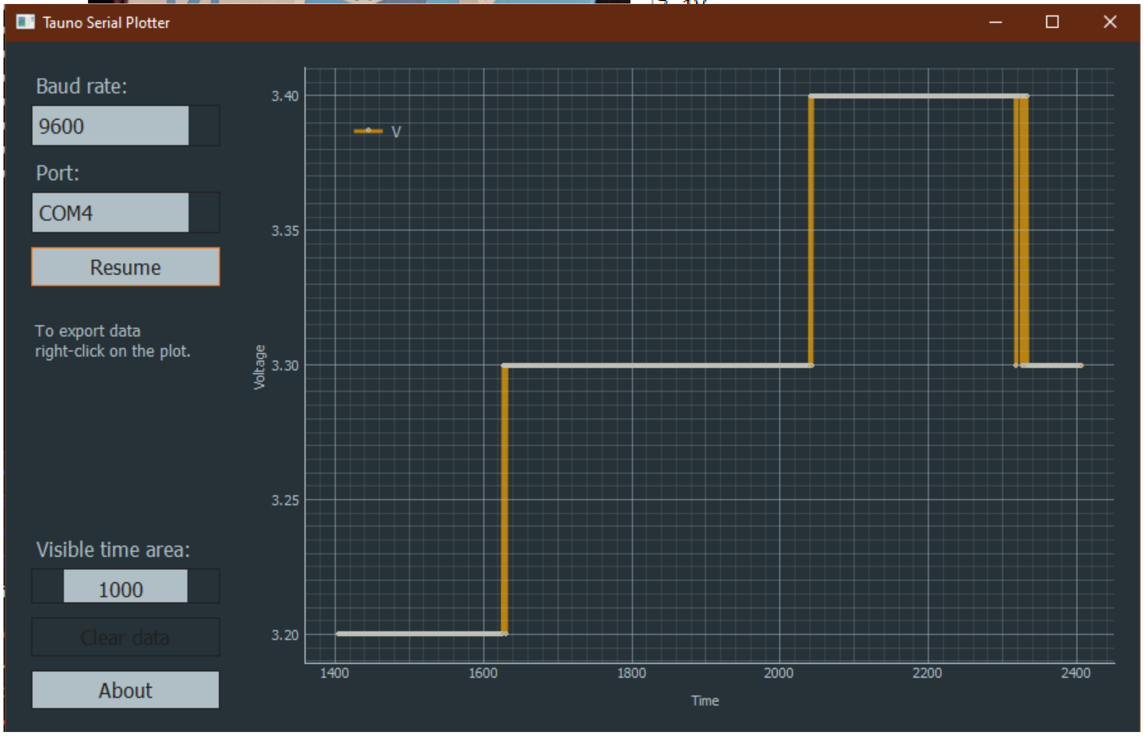
Terminal Window

Connect COM4 Baud: 9600

Receive

```
3.1V
3.0V
3.1V
3.0V

3.0V
3.0V
3.1V
3.0V
3.0V
3.0V
3.1V
3.1V
3.0V
3.0V
3.0V
3.0V
3.1V
3.0V
3.0V
3.0V
3.0V
3.0V
1V
3.1V
3.1V
3.1V
3.1V
3.1V
3.1V
```





Tauno Serial Plotter

Baud rate: 9600

Port: COM4

Resume

To export data right-click on the plot.

Visible time area: 1000

Clear data

About

7. VIDEO LINKS OF EACH DEMO

https://youtu.be/iYqzKRNEMDY

8. GITHUB LINK OF THIS DA

https://github.com/sewelr2/DA-Submissions/tree/master/DA4

Student Academic Misconduct Policy

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

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

Ryan Sewell