CPE 301 - 1001 DESIGN ASSIGNMENT 4

The goal of the assignment is to write, implement and demonstrate using Microchip Studio 7 a C code for the AVR ATMEGA328pb microcontroller that performs the following functions:

• Read the ADC value from the POT connected to ACO/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.

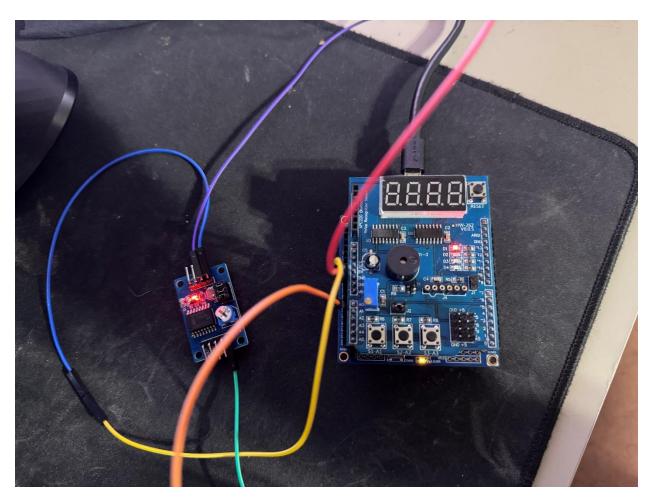
• Using a GUI Python script, display the ADC values as waveform (using tkinter).

Components Used/Connected

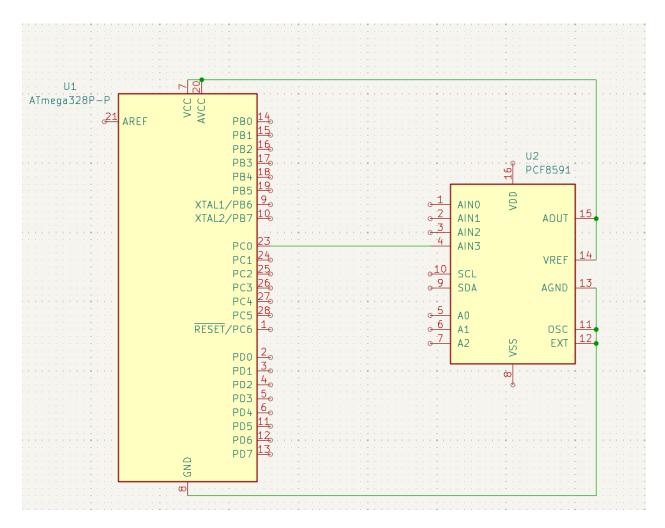
ATMega328P and Arduino Uno Pin Mapping

Arduino function	-		Arduino function
reset	(PCINT14/RESET) PC6 1	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0 □2	27 PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1□3	26 PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2 4	25 PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3 5	24 🗆 PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4 6	23 PC0 (ADC0/PCINT8)	analog input 0
VCC	vcc □ 7	22 GND	GND
GND	GND□8	21 AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6 ☐9	20 AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7) 19 PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	18 PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	2 17 PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	3 16 PB2 (SS/OC1B/PCINT2) (digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0 ☐1/	15 ☐ PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.



Atmega328p and potentiometer setup



Schematic. Instead of an individual potentiometer I used the PCF8591T module which had a potentiometer on it. I just had to hook up AIN3 since that's where the voltage was read.

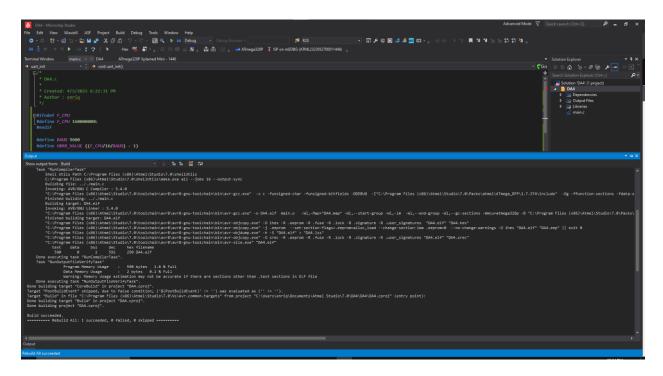
AVR C Code

```
* DA4.c
 * Created: 4/5/2025 6:22:31 PM
 * Author : enriq
#ifndef F CPU
#define F_CPU 16000000UL
#endif
#define BAUD 9600
#define UBRR_VALUE ((F_CPU/16/BAUD) - 1)
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
volatile uint16_t adc_result = 0;
void uart init() {
        UBRROH = (UBRR_VALUE >> 8);
        UBRRØL = UBRR_VALUE;
        UCSR0B = (1 << TXEN0);
        UCSROC = (1 << UCSZO1) | (1 << UCSZOO);
}
void uart_send(char c) {
        while (!(UCSR0A & (1<<UDRE0)));</pre>
                 UDR0 = c;
}
void adc_init() {
        ADMUX = (1 << REFS0);
        ADCSRA = (1 < ADEN) | (1 < ADATE) | (1 < ADIE) | (1 < ADPS2) | (1 < ADPS1) | (1 < ADPS0); //
Enable ADC, Auto Trigger, Interrupt
        ADCSRB = 0x00;
        DIDR0 = (1 << ADC0D);
        ADCSRA |= (1<<ADSC); // Start ADC conversion
}
void timer_init() {
        TCCR1B |= (1<<WGM12); // CTC Mode
        OCR1A = 12499; // OCR1A = 2499; // 10ms
        TCCR1B |= (1<<CS11)|(1<<CS10); // prescaler 64
        TIMSK1 = (1 << OCIE1A);
}
ISR(ADC vect) {
        adc_result = ADC;
}
```

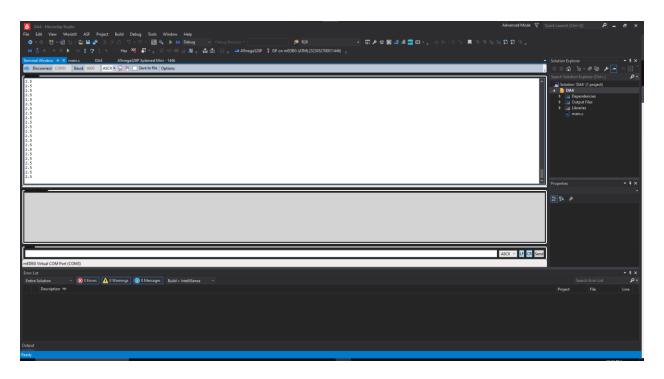
```
ISR(TIMER1_COMPA_vect) {
        uint16_t mv = (adc_result * 500) / 1023;  // Convert ADC to millivolts

        uart_send((mv / 10) + '0');  // Send whole number part
        uart_send('.');
        uart_send((mv % 10) + '0');  // Send decimal part
        uart_send('\n');
}

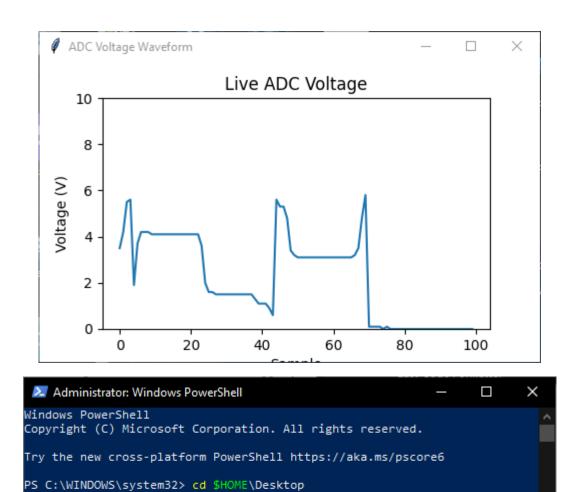
int main(void) {
        uart_init();
        adc_init();
        timer_init();
        sei();  // set global interrupt
        while (1) {
        }
}
```



Successful Compilation



Successfully reading values from potentiometer in COM3 at 9600 Baud



Running python script. Used a generic graph template for tkinter to make the script.

PS C:\Users\enriq\Desktop> python da4.py