CPE 301 - 1001

DESIGN ASSIGNMENT 5

Write, implement, and demonstrate using Atmel Studio 7 a C code for the AVR ATMEGA328p microcontroller that performs the following functions:
You'll use the ADC, and PWM/CCP Module of the ATmega328/p to set and determine the speed of the DC Motor.

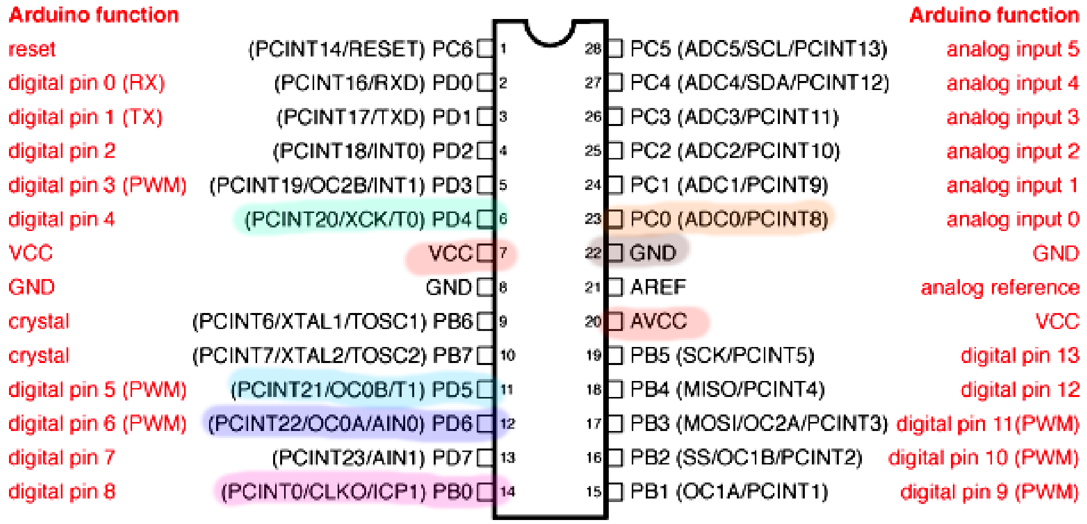
1. Using the Potentiometer connected to ADC0, translate the ADC value (0~1023) to PWM value/speed of the motor (0~255 if using Timer0/2). Verify the operation.

2. Using the CCP capture pin of PWM1, in mode 1x determine the speed of the DC Motor for a set ADC Pot value/position.

3. Develop a UART GUI interface to override the ADC speed control, the user input will control the speed of the motor. Plot the set speed and current speed using a UART GUI tool.

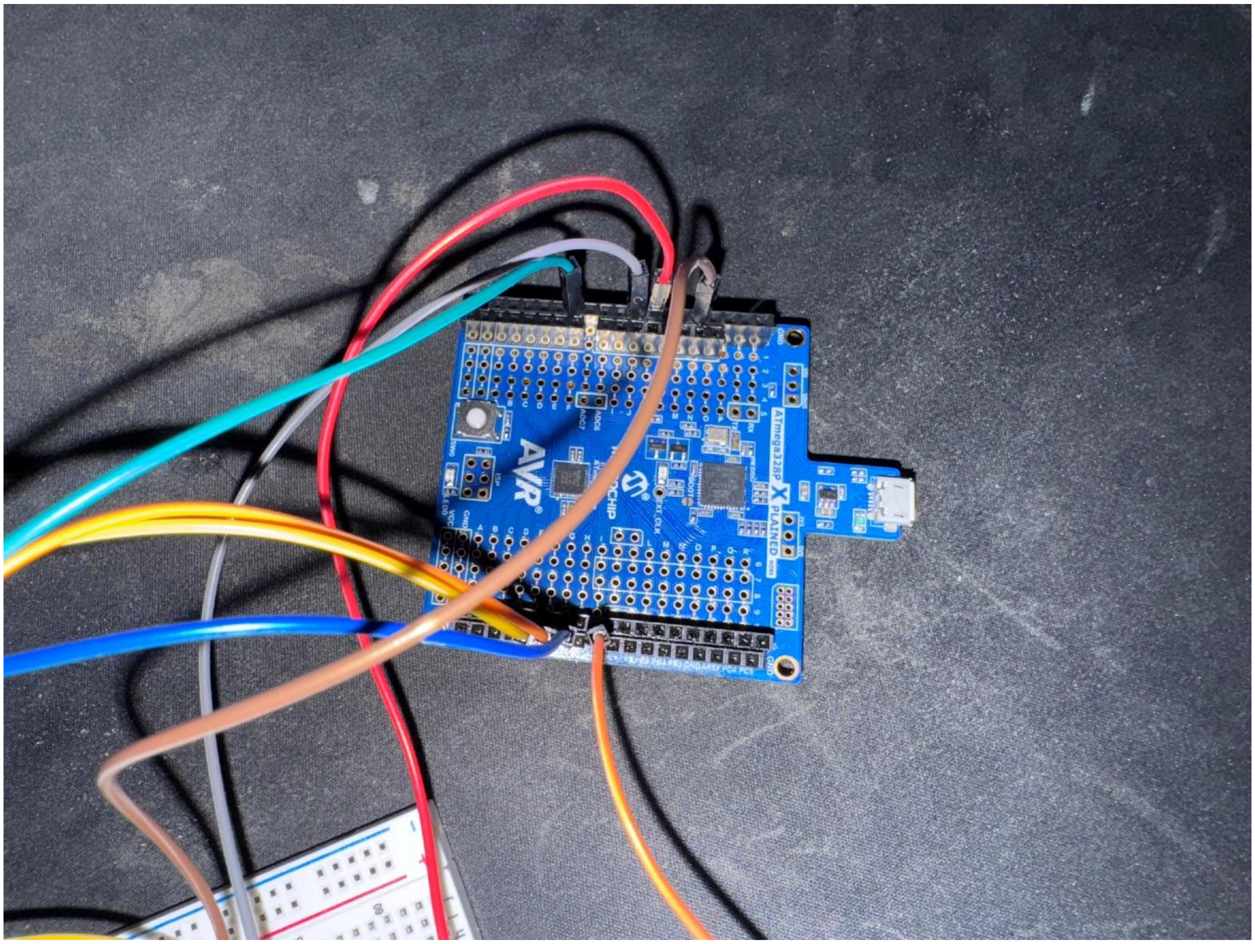
Components Used/Connected

ATMega328P and Arduino Uno Pin Mapping

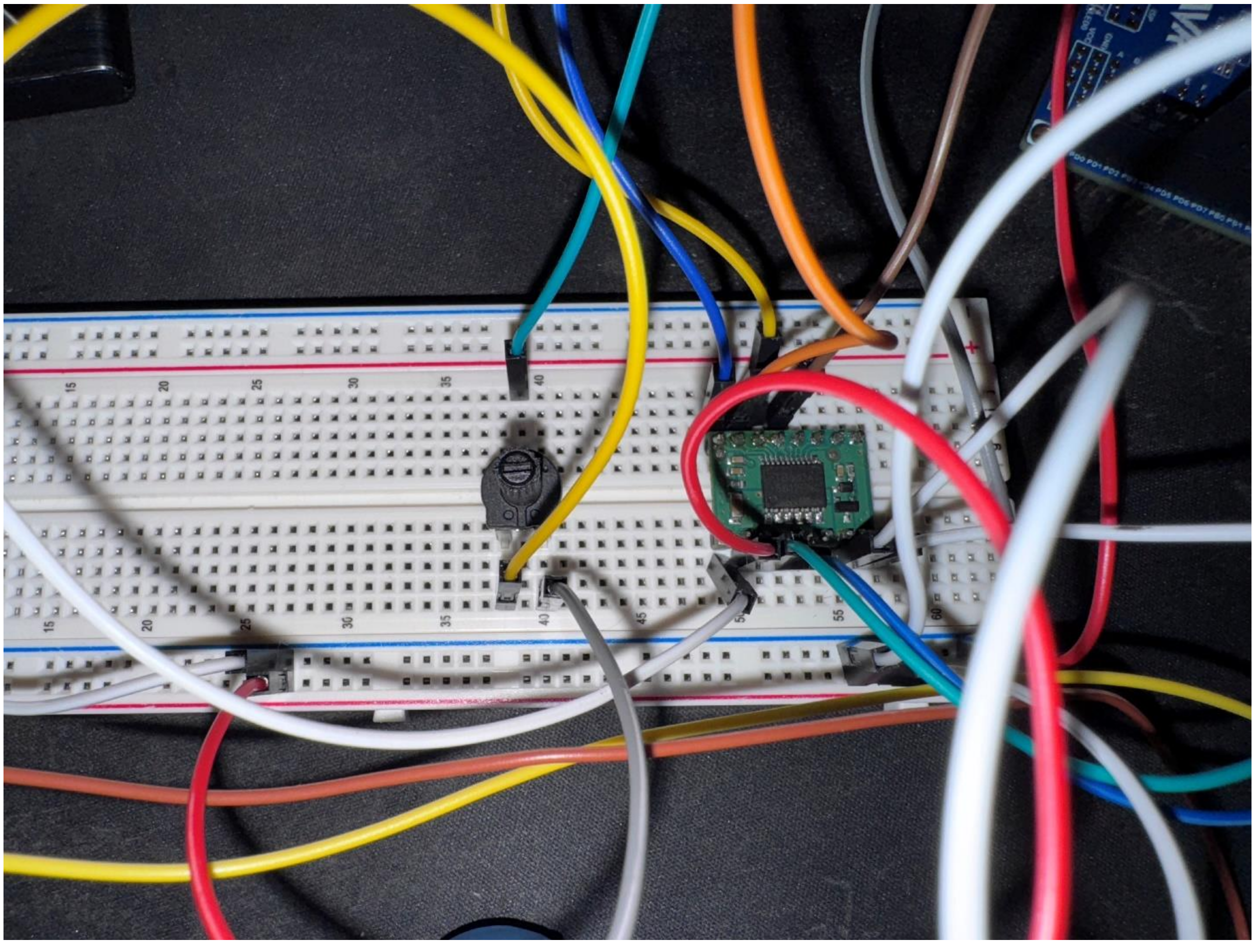

Arduino function
Arduino function
reset (PCINT14/RESET) PC6 1
28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX) (PCINT16/RXD) PD0 2
27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX) (PCINT17/TXD) PD1 3
26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2 (PCINT18/INT0) PD2 4
25 PC2 (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 10
19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM) (PCINT21/OC0B/T1) PD5 11
18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM) (PCINT22/OC0A/AIN0) PD6 12
17 PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7 (PCINT23/AIN1) PD7 13
16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8 (PCINT0/CLKO/ICP1) PB0 14
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.

DA5

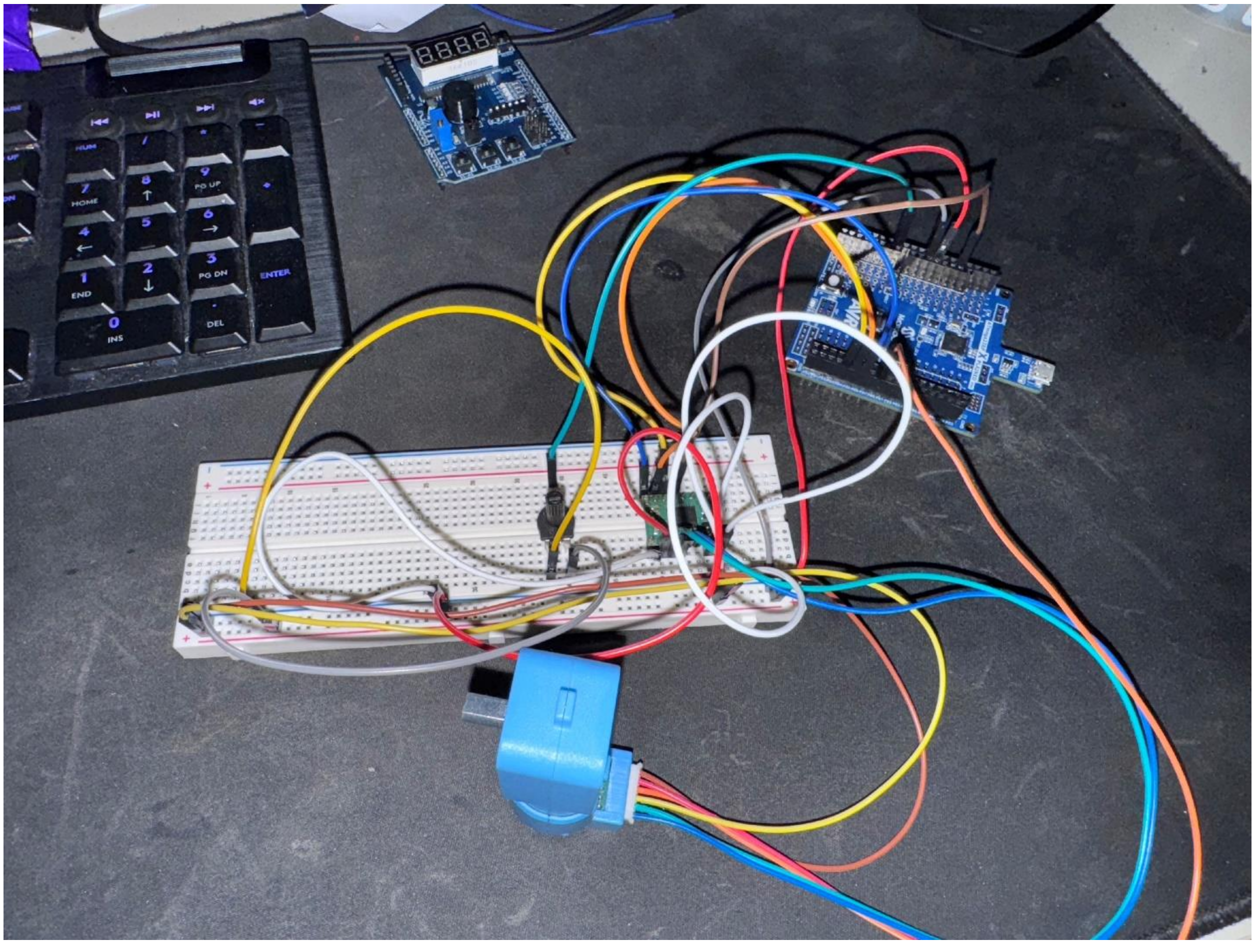


Atmega328p wiring



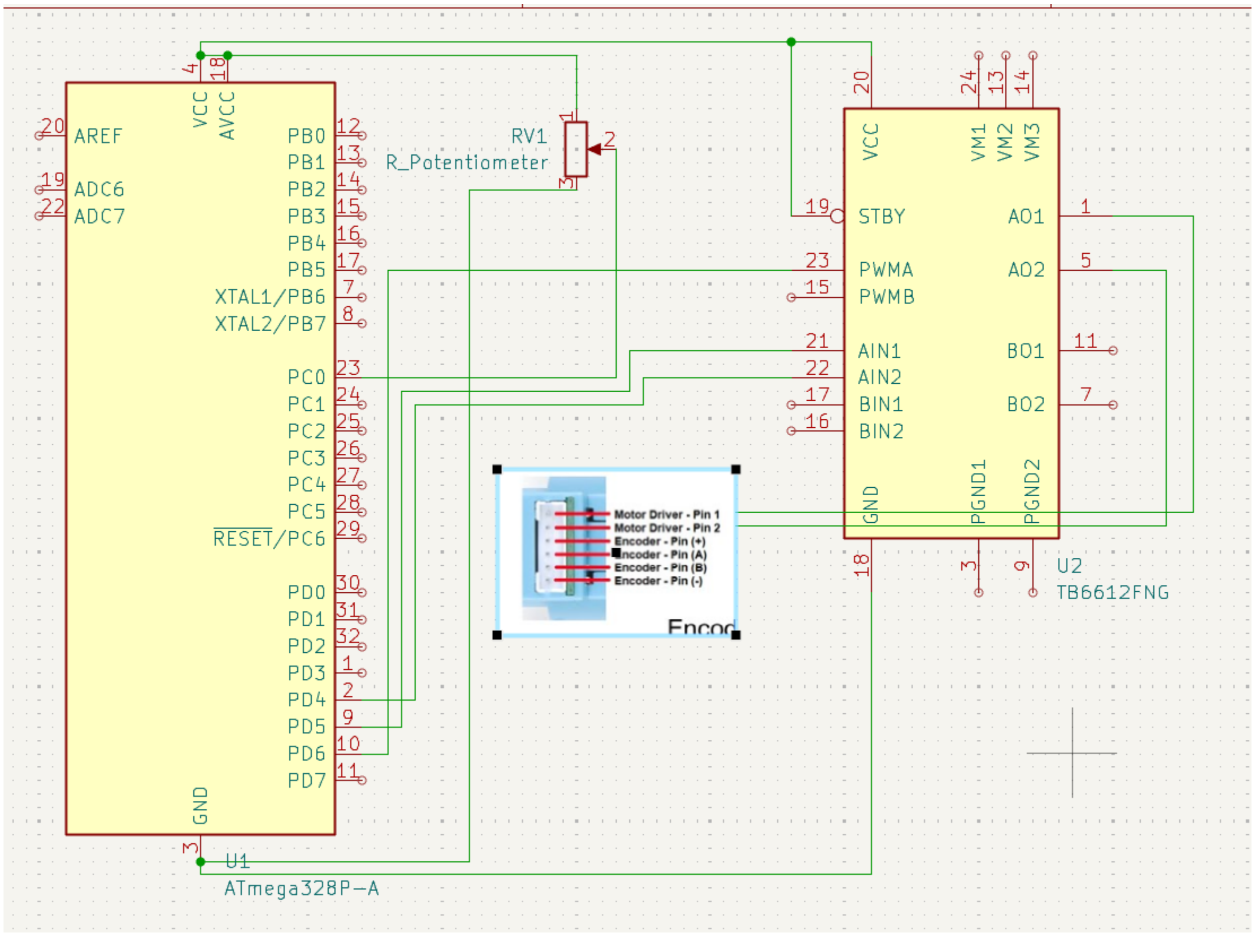
Breadboard layout.

DA5



Whole setup.

DA5

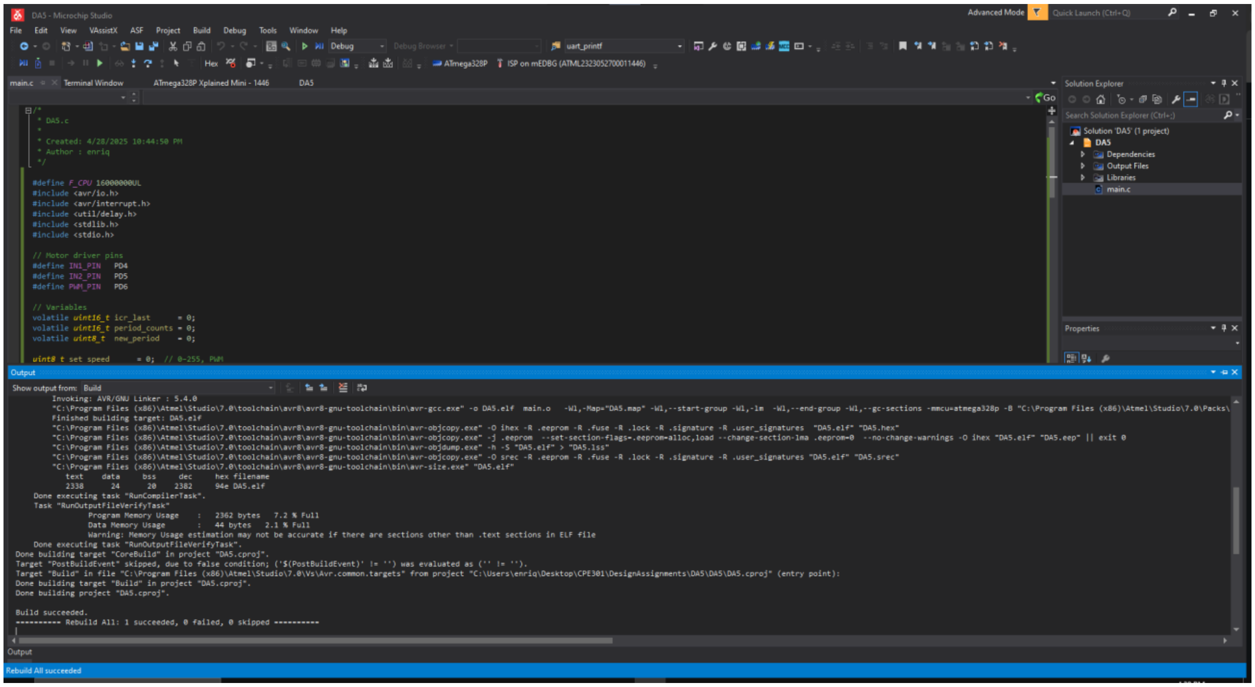


Schematic showing how everything was set up.

AVR C Code

```
/*
 * DA5.c
 *
 * Created: 4/28/2025 10:44:50 PM
 * Author : enriq
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdio.h>

// Motor driver pins
#define IN1_PIN   PD4
#define IN2_PIN   PD5
#define PWM_PIN   PD6

// Variables
volatile uint16_t icr_last      = 0;
volatile uint16_t period_counts = 0;
volatile uint8_t  new_period    = 0;

uint8_t set_speed      = 0;  // 0-255, PWM
uint8_t measured_speed = 0;  // 0-255, from measurement

// UART parsing
char    rx_buf[4];
uint8_t rx_pos         = 0;
uint8_t override_flag  = 0;
uint8_t override_speed = 0;

// UART at 9600 baud
static int uart_putchar(char c, FILE *stream) {
        while (!(UCSR0A & (1<<UDRE0)));
        UDR0 = (uint8_t)c;
        return 0;
}
static FILE uart_str = FDEV_SETUP_STREAM(uart_putchar, NULL, _FDEV_SETUP_WRITE);

void uart_init(void) {

        UBRR0  = F_CPU/16/9600 - 1;
        UCSR0B = (1<<TXEN0)|(1<<RXEN0);
        UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);
        stdout  = &uart_str;
}

// ADC
void adc_init(void) {
        ADMUX  = (1<<REFS0);
        ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1);
}
uint16_t adc_read(void) {
        ADCSRA |= (1<<ADSC);
        while (ADCSRA & (1<<ADSC));
        return ADC;
}

// PWM (PD6)
void pwm0_init(void) {
        DDRD  |= (1<<PWM_PIN);
        TCCR0A = (1<<COM0A1)|(1<<WGM01)|(1<<WGM00);
        TCCR0B =  (1<<CS01);
}

// ICP1 (PB0)
ISR(TIMER1_CAPT_vect) {
        uint16_t ic = ICR1;
        period_counts = ic - icr_last;
        icr_last      = ic;
        new_period    = 1;
}
void icap_init(void) {
        DDRB   &= ~(1<<PB0);
        TCCR1B  = (1<<ICNC1)|(1<<ICES1)|(1<<CS11);
        TIMSK1  = (1<<ICIE1);
}

// Motor direction pins (PD4/PD5)
void motor_dir_init(void) {
        DDRD  |= (1<<IN1_PIN)|(1<<IN2_PIN);
        PORTD |=  (1<<IN1_PIN);  // IN1=1
        PORTD &= ~(1<<IN2_PIN);  // IN2=0 → forward
}

int main(void) {
        motor_dir_init();
        pwm0_init();
        adc_init();
        uart_init();
        icap_init();
        sei();

        while (1) {

                if (UCSR0A & (1<<RXC0)) {
                        char c = UDR0;
                        if (c >= '0' && c <= '9' && rx_pos < sizeof(rx_buf) - 1) {
                                rx_buf[rx_pos++] = c;
                        }
                        else if ((c=='\r' || c=='\n') && rx_pos) {
                                rx_buf[rx_pos] = '\0';
                                uint16_t v = atoi(rx_buf);
                                if (v > 255) v = 255;
                                override_speed = (uint8_t)v;
                                override_flag  = 1;
                                rx_pos = 0;
                        }
                }

                // set_speed
                if (override_flag) {
                        set_speed = override_speed;
                        } else {
                        uint16_t raw = adc_read();
                        set_speed = raw >> 2;
                }
                OCR0A = set_speed;

                // Update measured_speed
                if (new_period) {
                        new_period = 0;
                        if (period_counts) {
                                uint32_t freq = F_CPU/8/period_counts;
                                measured_speed = (freq > 255 ? 255 :
(uint8_t)freq);
                                } else {
                                measured_speed = 0;
                        }
                }

                // Print on BAUD 9600
                uint16_t temp = ADC;  // last ADC reading
                printf("%u %u %u\n", temp, set_speed, measured_speed);


                _delay_ms(200);
        }
}
```

DA5



DA5 - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
main.c
Terminal Window
ATmega328P Xplained Mini - 1446
DA5
Solution Explorer
Solution 'DA5' (1 project)
DA5
Dependencies
Output Files
Libraries
main.c
Properties
Output

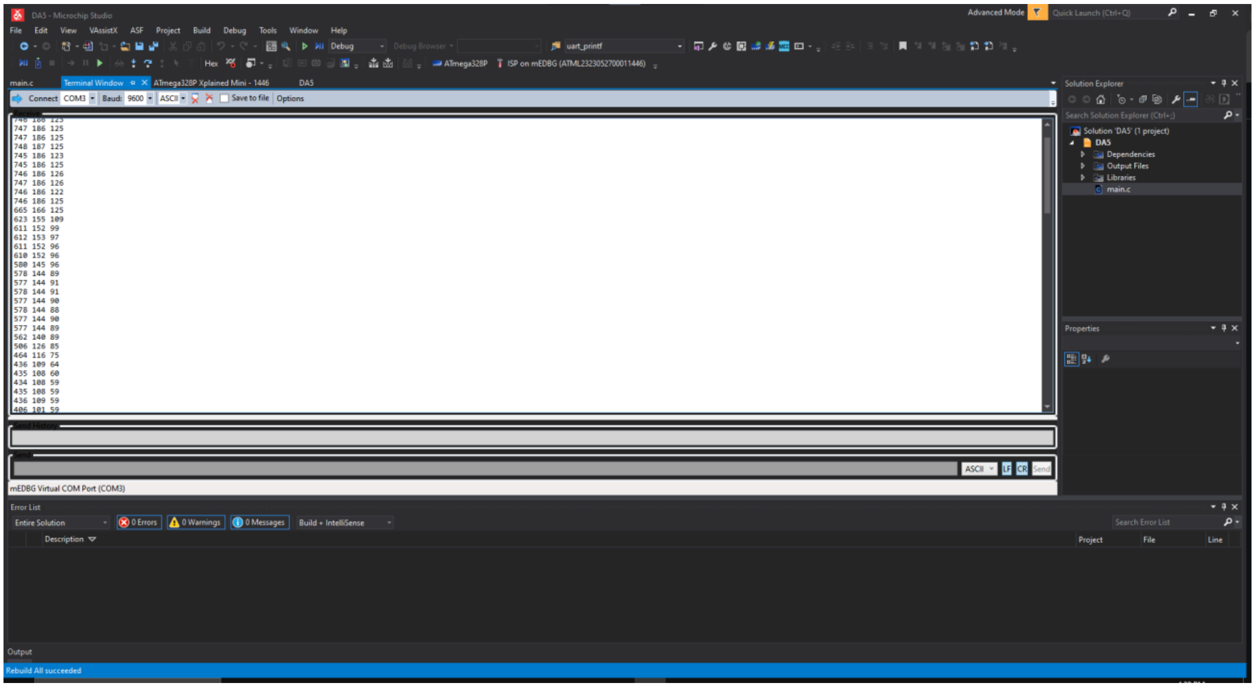
```c
/*
 * DA5.c
 *
 * Created: 4/28/2025 10:44:50 PM
 * Author : enriq
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdio.h>

// Motor driver pins
#define IN1_PIN   PD4
#define IN2_PIN   PD5
#define PWM_PIN   PD6

// Variables
volatile uint16_t icr_last      = 0;
volatile uint16_t period_counts = 0;
volatile uint8_t  new_period    = 0;

uint8_t set_speed      = 0;  // 0-255, PWM
```

Show output from: Build

```
        Invoking: AVR/GNU Linker : 5.4.0
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-gcc.exe" -o DA5.elf  main.o   -Wl,-Map="DA5.map" -Wl,--start-group -Wl,-lm  -Wl,--end-group -Wl,--gc-sections -mmcu=atmega328p -B "C:\Program Files (x86)\Atmel\Studio\7.0\Packs\
        Finished building target: DA5.elf
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-objcopy.exe" -O ihex -R .eeprom -R .fuse -R .lock -R .signature -R .user_signatures  "DA5.elf" "DA5.hex"
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-objcopy.exe" -j .eeprom  --set-section-flags=.eeprom=alloc,load --change-section-lma .eeprom=0  --no-change-warnings -O ihex "DA5.elf" "DA5.eep" || exit 0
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-objdump.exe" -h -S "DA5.elf" > "DA5.lss"
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-objcopy.exe" -O srec -R .eeprom -R .fuse -R .lock -R .signature -R .user_signatures "DA5.elf" "DA5.srec"
        "C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\bin\avr-size.exe" "DA5.elf"
           text    data     bss     dec     hex filename
           2338      24      20    2382     94e DA5.elf
    Done executing task "RunCompilerTask".
    Task "RunOutputFileVerifyTask"
                Program Memory Usage    :   2362 bytes   7.2 % Full
                Data Memory Usage       :   44 bytes   2.1 % 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 "DA5.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\enriq\Desktop\CPE301\DesignAssignments\DA5\DA5\DA5.cproj" (entry point):
Done building target "Build" in project "DA5.cproj".
Done building project "DA5.cproj".

Build succeeded.
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

Rebuild All succeeded

Successful Compilation



Connect COM3 Baud: 9600 ASCII □ Save to file Options

```
746 186 125
747 186 125
747 186 125
748 187 125
745 186 123
745 186 125
746 186 126
747 186 126
746 186 122
746 186 125
665 166 125
623 155 109
611 152 99
612 153 97
611 152 96
610 152 96
580 145 96
578 144 89
577 144 91
578 144 91
577 144 90
578 144 88
577 144 90
577 144 89
562 140 89
506 126 85
464 116 75
436 109 64
435 108 60
434 108 59
435 108 59
436 109 59
406 101 59
```

mEDBG Virtual COM Port (COM3)

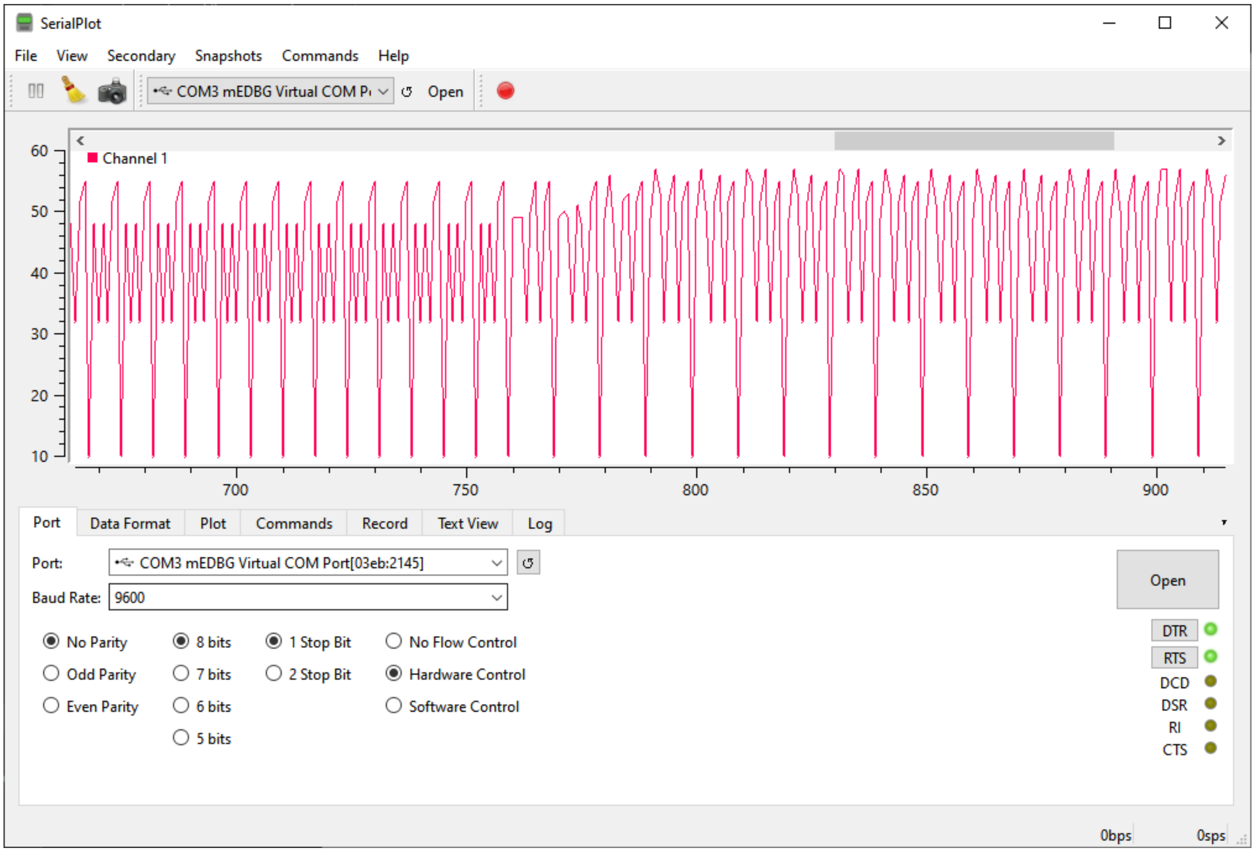
0 Errors 0 Warnings 0 Messages

Rebuild All succeeded

Successfully reading values from motor and potentiometer.

The first value is ADC, second value is the potentiometer value, and third value is the measured speed.

DA5



Port: COM3 mEDBG Virtual COM Port[03eb:2145]

Baud Rate: 9600

● No Parity ● 8 bits ● 1 Stop Bit ○ No Flow Control
○ Odd Parity ○ 7 bits ○ 2 Stop Bit ● Hardware Control
○ Even Parity ○ 6 bits ○ Software Control
○ 5 bits

SerialPlot test