CPE301 – SPRING 2025

Design Assignment 6

Student Name: Ryan Sewell
Student #: 8000473785
Student Email: sewelr2@unlv.nevada.edu
Primary Github address: https://github.com/sewelr2
Directory: DA-Submissions/DA6
Video Playlist:
https://youtube.com/playlist?list=PLt45mEFhRV6ffOYRcGHhoI5aDeP3Zgqt5&feature=shared

The core objective of this lab is to build a single AVR-based motor-control demo in Atmel Studio 7 that:

1. Reads a potentiometer on ADC0 (0–1023) and maps it to an 8-bit PWM duty (0–255) on OC0A to drive your DC motor.
2. Measures the actual motor speed using the Timer/CCP input-capture (ICP1) hardware.
3. Overrides the pot-based setpoint via a simple UART "GUI," streaming out CSV pairs of (set-value, measured-speed) so you can live-plot both traces in a PC tool.

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Microchip Studio
- Assembler
- Simulator
- Debugger

Atmega328PB-Xmini PC
- Polulu md08a
- DC Motor

Multi-Function Shield
- Potentiometer

Tauno Serial Plotter

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdint.h>

//—— your Wait() from reference (~20 ms) ———————————————————————
void Wait(void)
{
        uint8_t i;
        for (i = 0; i < 50; i++)
        {
                _delay_loop_2(0);
                _delay_loop_2(0);
                _delay_loop_2(0);
        }
}

//—— hardware definitions ————————————————————————————————————
// Servo PWM on PB1/OC1A (Timer1)
#define SERVO_DDR   DDRB
```

```
#define SERVO_PIN   PINB1

// HC-SR04 Trigger on PC1
#define TRIG_DDR    DDRC
#define TRIG_PORT   PORTC
#define TRIG_PIN    PINC1

// HC-SR04 Echo on PD6
#define ECHO_PINR   PIND
#define ECHO_PIN    PIND6

//—— USART @9600, TX only ———————————————————————————————————
#define BAUD        9600
#define UBRR_VAL    ((F_CPU/16/BAUD) - 1)

static void USART_init(void)
{
       UBRR0H = (uint8_t)(UBRR_VAL >> 8);
       UBRR0L = (uint8_t)UBRR_VAL;
       UCSR0B = (1<<TXEN0);                    // TX enable
       UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);       // 8N1
}

static void USART_send(char c)
{
       while (!(UCSR0A & (1<<UDRE0)));
       UDR0 = c;
}

static void USART_print_u16(uint16_t x)
{
       char buf[6];
       uint8_t i = 0;
       if (x == 0)
       {
              USART_send('0');
              return;
       }
       while (x && i < sizeof(buf))
       {
              buf[i++] = '0' + (x % 10);
              x /= 10;
       }
       while (i--)
       USART_send(buf[i]);
}

//—— Servo (Timer1 Fast PWM Mode 14, prescaler=64 → 50 Hz) ———————
static void servo_init(void)
{
       SERVO_DDR |= (1<<SERVO_PIN);            // PB1 output

       // COM1A1=1 non-inverting OC1A, WGM11=1
       TCCR1A = (1<<COM1A1)|(1<<WGM11);
       // WGM13=1, WGM12=1, CS11=1, CS10=1 → prescaler=64, Mode 14
       TCCR1B = (1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10);

       ICR1 = 4999;                            // TOP = 4999 → 50 Hz
```

```
}

// map 0–180° → 250–500 ticks (1 ms–2 ms @ 4 µs/tick)
static inline void servo_setAngle(uint8_t angle)
{
       OCR1A = 250 + ((uint32_t)angle * 250) / 180;
}

//── HC-SR04 setup & measurement by polling TCNT1 ─────────────
static void ultrasonic_init(void)
{
       TRIG_DDR  |=  (1<<TRIG_PIN);            // trigger pin output
       DDRD      &= ~(1<<ECHO_PIN);            // echo pin input
       PORTD     &= ~(1<<ECHO_PIN);            // no pull-up
}

static uint16_t ultrasonic_read_raw(void)
{
       uint16_t start, end;

       // 10 µs trigger pulse
       TRIG_PORT |=  (1<<TRIG_PIN);
       _delay_us(10);
       TRIG_PORT &= ~(1<<TRIG_PIN);

       // wait for echo high
       while (!(ECHO_PINR & (1<<ECHO_PIN)));
       start = TCNT1;

       // wait for echo low
       while  (ECHO_PINR &  (1<<ECHO_PIN));
       end = TCNT1;

       return end - start;  // raw ticks (4 µs each)
}

//── Main sweep ──────────────────────────────────────────
int main(void)
{
       USART_init();
       servo_init();
       ultrasonic_init();

       while (1)
       {
              // CW sweep: 0→180 in 2° steps
              for (uint8_t ang = 0; ang <= 180; ang += 2)
              {
                     servo_setAngle(ang);
                     Wait();

                     uint16_t raw = ultrasonic_read_raw();
                     USART_print_u16(ang);
                     USART_send(',');
                     USART_print_u16(raw);
                     USART_send('\n');
              }
              // CCW sweep: 180→0
```

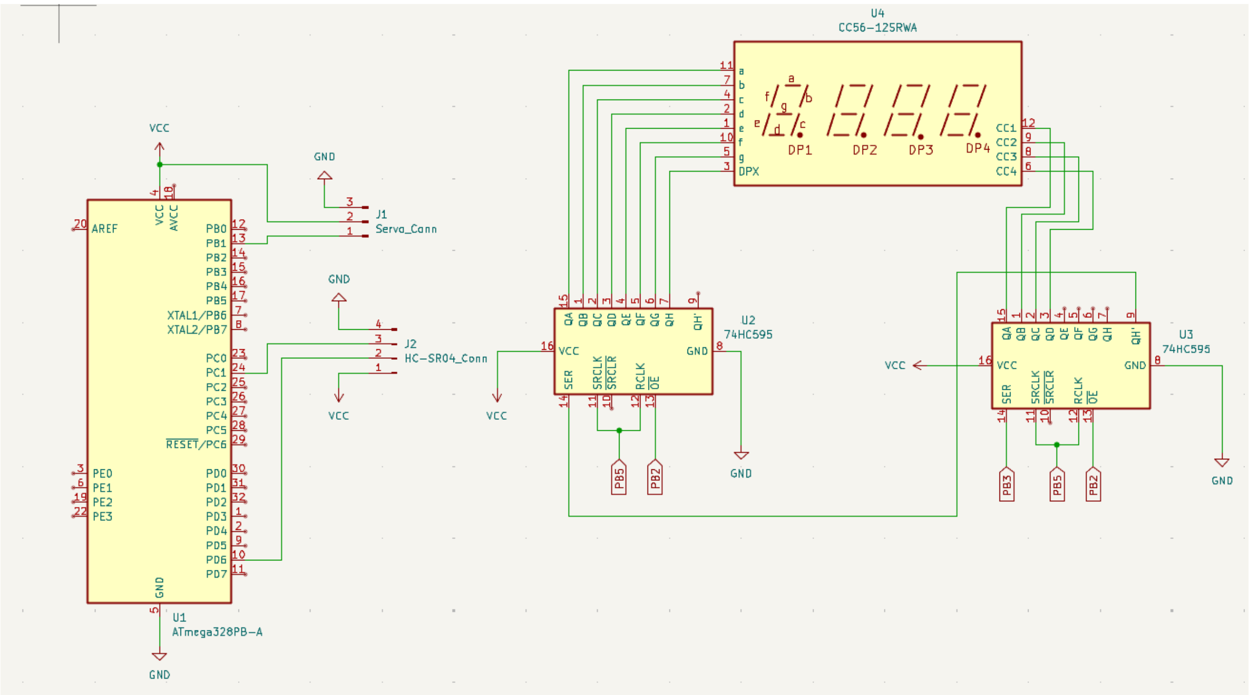
```
        for (int8_t ang = 180; ang >= 0; ang -= 2)
        {
            servo_setAngle(ang);
            Wait();

            uint16_t raw = ultrasonic_read_raw();
            USART_print_u16(ang);
            USART_send(',');
            USART_print_u16(raw);
            USART_send('\n');
        }
    }
}
```

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

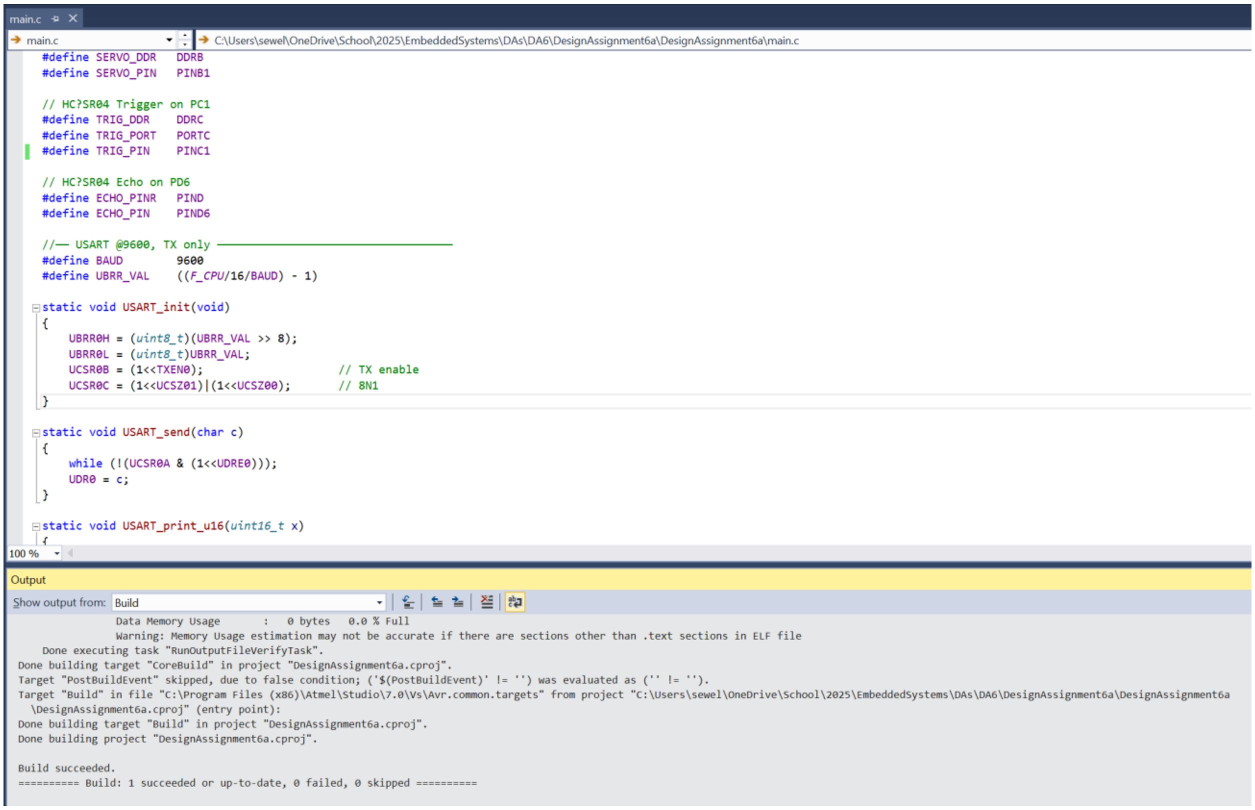
```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdint.h>

//——— your Wait() from reference (~20 ms) ——————————————————————————————————————————————
void Wait(void)
{
    uint8_t i;
    for (i = 0; i < 50; i++)
    {
        _delay_loop_2(0);
        _delay_loop_2(0);
        _delay_loop_2(0);
    }
}

//——— hardware definitions ——————————————————————————————————————————————————————————————
——————————

// Servo PWM on PB1/OC1A (Timer1)
#define SERVO_DDR   DDRB
#define SERVO_PIN   PINB1

// HC-SR04 Trigger on PC1
#define TRIG_DDR    DDRC
#define TRIG_PORT   PORTC
#define TRIG_PIN    PINC1

// HC-SR04 Echo on PD6
#define ECHO_PINR   PIND
#define ECHO_PIN    PIND6

// USART0 TX only @9600 baud
#define BAUD        9600
#define UBRR_VAL    ((F_CPU/16/BAUD) - 1)
```

```
// SPI pins for MAX7219 (7-seg driver)
#define SPI_DDR     DDRB
#define SPI_PORT    PORTB
#define SPI_MOSI    PINB3
#define SPI_SCK     PINB5
#define SPI_SS      PINB2

//——— SPI0 + MAX7219 routines ——————————————————————————————————————————————
———
void SPI_init(void)
{
    SPI_DDR |= (1<<SPI_MOSI)|(1<<SPI_SCK)|(1<<SPI_SS);
    SPI_PORT |= (1<<SPI_SS);
    SPCR0 = (1<<SPE0)|(1<<MSTR0)|(1<<SPR00);  // SPI0 enabled, Master, Fosc/16
}

void max7219_send(uint8_t reg, uint8_t data)
{
    SPI_PORT &= ~(1<<SPI_SS);
    SPDR0 = reg;
    while (!(SPSR0 & (1<<SPIF0)));
    SPDR0 = data;
    while (!(SPSR0 & (1<<SPIF0)));
    SPI_PORT |= (1<<SPI_SS);
}

void max7219_init(void)
{
    max7219_send(0x09, 0x0F);  // decode mode: digits 0 - 3
    max7219_send(0x0A, 0x0F);  // intensity
    max7219_send(0x0B, 0x03);  // scan limit: 4 digits
    max7219_send(0x0C, 0x01);  // normal operation
    max7219_send(0x0F, 0x00);  // display test: off
}

static const uint16_t pow10[4] = {1, 10, 100, 1000};

void displayNumber(uint16_t num)
{
    for (uint8_t d = 0; d < 4; d++)
    {
        uint8_t val = (num / pow10[d]) % 10;
        max7219_send(d + 1, val);
    }
}

//——— USART0 TX only ——————————————————————————————————————————————————————
————————
void USART_init(void)
{
    UBRR0H = (uint8_t)(UBRR_VAL >> 8);
    UBRR0L = (uint8_t)UBRR_VAL;
    UCSR0B = (1<<TXEN0);
    UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);
}

void USART_send(char c)
{
    while (!(UCSR0A & (1<<UDRE0)));
    UDR0 = c;
}

void USART_print_u16(uint16_t x)
{
    char buf[6];
    uint8_t i = 0;
    if (x == 0) { USART_send(’0’); return; }
    while (x && i < sizeof(buf))
    {
        buf[i++] = ’0’ + (x % 10);
        x /= 10;
    }
    while (i--) USART_send(buf[i]);
}

//── Servo (Timer1 Fast PWM Mode 14, prescaler=64 → 50 Hz) ───────
void servo_init(void)
{
    SERVO_DDR |= (1<<SERVO_PIN);
    TCCR1A = (1<<COM1A1)|(1<<WGM11);
    TCCR1B = (1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10);
    ICR1 = 4999;
}

static inline void servo_setAngle(uint8_t angle)
{
    OCR1A = 250 + ((uint32_t)angle * 250) / 180;
}

//── HC-SR04 setup & measurement by polling TCNT1 ─────────────
void ultrasonic_init(void)
{
    TRIG_DDR  |=  (1<<TRIG_PIN);
    DDRD      &= ~(1<<ECHO_PIN);
    PORTD     &= ~(1<<ECHO_PIN);
}

uint16_t ultrasonic_read_raw(void)
{
    uint16_t start, end;
    TRIG_PORT |=  (1<<TRIG_PIN);
    _delay_us(10);
    TRIG_PORT &= ~(1<<TRIG_PIN);
    while (!(ECHO_PINR & (1<<ECHO_PIN)));
    start = TCNT1;
    while  (ECHO_PINR &  (1<<ECHO_PIN));
    end   = TCNT1;

    return end - start;
}

//--- Main sweep with 7-SEG display --------------------------------------------------
int main(void)
{
    USART_init();
    SPI_init();
    max7219_init();
    servo_init();
    ultrasonic_init();

    while (1)
    {
        uint16_t min_raw = 0xFFFF;

        // CW: display each raw reading
        for (uint8_t ang = 0; ang <= 180; ang += 2)
        {
            servo_setAngle(ang);
            Wait();

            uint16_t raw = ultrasonic_read_raw();
            if (raw < min_raw) min_raw = raw;

            // log on USART
            USART_print_u16(ang);
            USART_send(’,’);
            USART_print_u16(raw);
            USART_send(’\n’);

            // show current reading on 7-seg
            displayNumber(raw);
        }

        // CCW: display lowest reading from CW scan
        for (int8_t ang = 180; ang >= 0; ang -= 2)
        {
            servo_setAngle(ang);
            Wait();
            displayNumber(min_raw);
        }
    }
```

4. SCHEMATICS


U4
CC56-12SRWA
a
b
c
d
e
f
g
DPX
DP1
DP2
DP3
DP4
CC1
CC2
CC3
CC4
VCC
GND
J1
Servo_Conn
J2
HC-SR04_Conn
U1
ATmega328PB-A
AREF
AVCC
PB0
PB1
PB2
PB3
PB4
PB5
XTAL1/PB6
XTAL2/PB7
PC0
PC1
PC2
PC3
PC4
PC5
RESET/PC6
PE0
PE1
PE2
PE3
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
U2
74HC595
U3
74HC595
QA
QB
QC
QD
QE
QF
QG
QH
QH'
SER
SRCLK
SRCLR
RCLK
OE
PB5
PB2
PB3

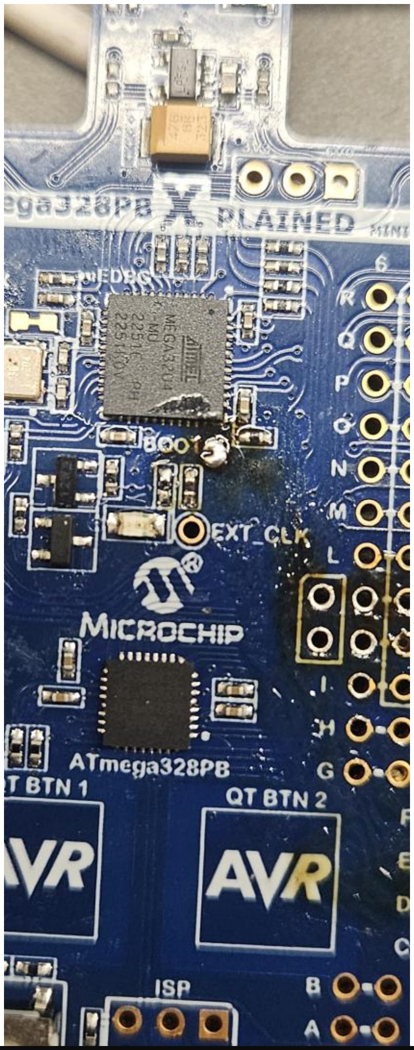
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

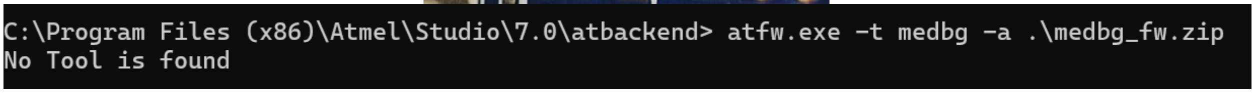

main.c
main.c
C:\Users\sewel\OneDrive\School\2025\EmbeddedSystems\DAs\DA6\DesignAssignment6a\DesignAssignment6a\main.c
100 %
Output
Show output from: Build

```
#define SERVO_DDR   DDRB
#define SERVO_PIN   PINB1

// HC?SR04 Trigger on PC1
#define TRIG_DDR    DDRC
#define TRIG_PORT   PORTC
#define TRIG_PIN    PINC1

// HC?SR04 Echo on PD6
#define ECHO_PINR   PIND
#define ECHO_PIN    PIND6

//— USART @9600, TX only ————————————————————————————
#define BAUD        9600
#define UBRR_VAL    ((F_CPU/16/BAUD) - 1)

static void USART_init(void)
{
    UBRR0H = (uint8_t)(UBRR_VAL >> 8);
    UBRR0L = (uint8_t)UBRR_VAL;
    UCSR0B = (1<<TXEN0);                 // TX enable
    UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);    // 8N1
}

static void USART_send(char c)
{
    while (!(UCSR0A & (1<<UDRE0)));
    UDR0 = c;
}

static void USART_print_u16(uint16_t x)
{
```

```
               Data Memory Usage       :   0 bytes   0.0 % 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 "DesignAssignment6a.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\sewel\OneDrive\School\2025\EmbeddedSystems\DAs\DA6\DesignAssignment6a\DesignAssignment6a
  \DesignAssignment6a.cproj" (entry point):
Done building target "Build" in project "DesignAssignment6a.cproj".
Done building project "DesignAssignment6a.cproj".

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

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)




```
C:\Program Files (x86)\Atmel\Studio\7.0\atbackend> atfw.exe -t medbg -a .\medbg_fw.zip
No Tool is found
```

Board was bricked and have been unable to unbrick it. Photos above show what I was seeing.

7. VIDEO LINKS OF EACH DEMO

8. GITHUB LINK OF THIS DA

https://github.com/sewelr2/DA-Submissions/tree/master/DA5

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

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