CPE301 – SPRING 2024

Design Assignment 6

Student Name: Johnathan Widney
Student #: 5000354900
Student Email: widney@unlv.nevada.edu
Primary Github address: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems
Directory: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main
Video Playlist:
https://www.youtube.com/playlist?list=PLoASw0sToF2VbleAjq4UsKuL5bDLANrj3

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/DA, 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).

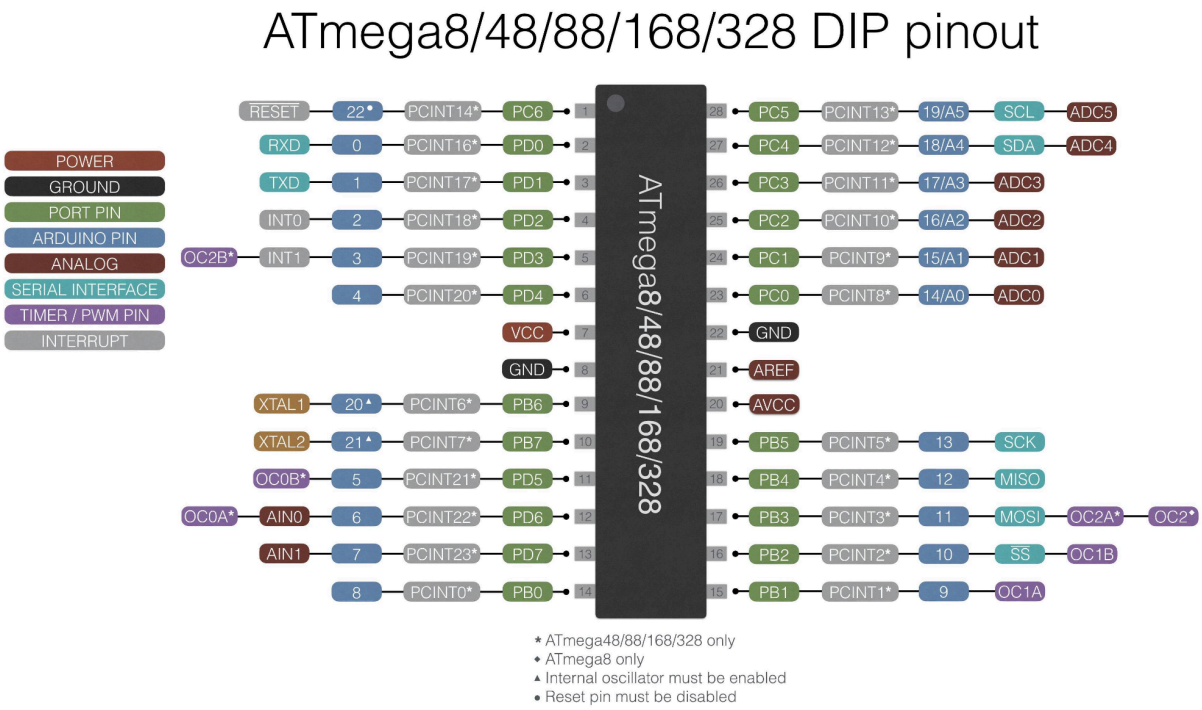
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used
Block diagram with pins used in the Atmega3PB (only)

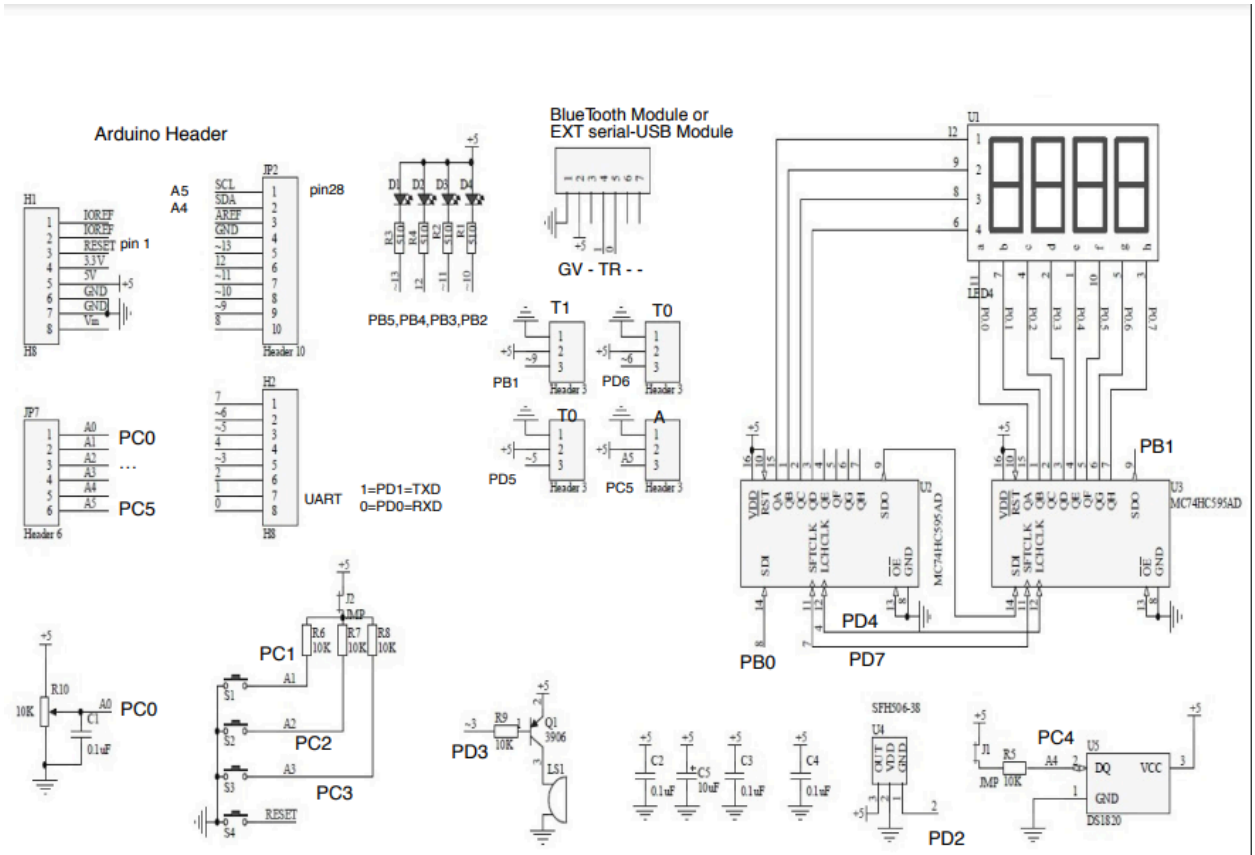
- Atmega328PB Xplained mini microcontroller board
- Arduino compatible external multifunction development shield
- Male-to-male jumper cables
- Mini breadboard
- Motor driver module
- 360 degree DC motor with encoder
- Female-to-female ribbon cable
- PC

Block diagrams and pins:

Atmega328PB Micro controller:



Arduino compatible multifunction development shield:



For assignments dealing with LED - use the pins PB5,PB4,PB3,PB2. For assignments dealing with switches, pin interrupts use pins PC1,PC2,PC3
For assignments in PWM use ~5/PD5 (T0B), ~6/PD6 (T0A), ~9/PB1 (T1A), ~10/PB2 (T1B-LED), ~11/PB3(T2A-LED),~3/PD3(T2B/Buzzer)
For assignments with analog input use A0/PC0-Potentiometer, A4/PC4/LM3X, or EXT @ PC5. PD2 is INT0 pin (external interrupt)

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1

```
/* This code contains all of tasks 1, 2, 3, and 4 */

// Initializing clock, constants, libraries, variables, etc.
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#define BUFFERSIZE 100
#define ASCII_NUM 48
#define SAMPRATE 100

#include <avr/io.h>                                    /* Include AVR std. library file */
#include <avr/interrupt.h>
#include <stdio.h>                                     /* Include std. library file */
#include <util/delay.h>                                /* Include Delay header file */
#include <avr/delay.h>
```

```
#include <stdlib.h>

// 7seg 74HC595
#define DATA (1<<PE3) //MOSI (SI)
#define LATCH (1<<PE2) //SS (RCK)
#define CLOCK (1<<PC1) //SCK (SCK)
#define DELAY_SEG 1

/* Segment byte maps for numbers 0 to 9 */
const uint8_t SEG_digit[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99,
0x92, 0x82, 0xF8, 0X80, 0X90};
/* Byte maps to select digit 1 to 4 */
const uint8_t SEG_select[] = {0xF1, 0xF2, 0xF4, 0xF8};
int dout, doutOLD;
int numDigits;
unsigned int STALL, OVERRIDE, DIRECTION, ONOFF;
float rpm;
unsigned int numBuffer[BUFFERSIZE];

// Global Vars
unsigned int count = 0;
int pwmVal;
float Analog;
char strPWM[5];
char strAnalog[100];
volatile uint8_t Direction = 0;

/* ADC Initialization function */
void ADC_Init() {
	DDRC = 0x00;				/* Make ADC port as input */
	ADCSRA = 0x87;				/* Enable ADC, with
freq/128  */
	ADMUX = 0x40;				/* Vref: Avcc, ADC channel: 0 */
}

/* ADC Read function */
int ADC_Read(char channel) {
	ADMUX = 0x40 | (channel & 0x07);	/* set input channel to read */
	ADCSRA |= (1<<ADSC);			/* Start ADC conversion */
	while (!(ADCSRA & (1<<ADIF)));		/* Wait until end of conversion by polling ADC
interrupt flag */
	ADCSRA |= (1<<ADIF);			/* Clear interrupt flag */
	_delay_us(1);				/* Wait a little bit */
	return ADCW;				/* Return ADC word */
}

ISR(INT0_vect) {
	TCCR0B |= (0<<CS00)|(0<<CS01);/* Set Fast PWM with Fosc/64 Timer0 clock */
	_delay_us(5000);				/* Software de-bouncing
control delay */
	TCCR0B |= (1<<CS00)|(1<<CS02);/* Set Fast PWM with Fosc/64 Timer0 clock */
}
```

```
void USART_init(unsigned int ubrr) {
	//Set baud rate
	UBRR0H = (unsigned char)(ubrr>>8);
	UBRR0L = (unsigned char) ubrr;
	// enable transmitter
	UCSR0B = (1<<TXEN0) | (1<<RXEN0); // Enable Transmit and Receive
	// Set frame format: async, no parity, 1 stop bit, , 8 data bits
	UCSR0C =
(0<<UMSEL01)|(0<<UMSEL00)|(0<<UPM01)|(0<<UPM00)|(0<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00);
}

void USART_transmit(const char* data) {
	while (*data) {
		//check if buffer is empty so that data can be written to transmit
		while (!(UCSR0A & (1 << UDRE0)));
		UDR0 = *data; //copy “data” to be sent to UDR0
		++data;
	}
}

void USART_transmitChar(const char data) {
	//check if buffer is empty so that data can be written to transmit
	while (!(UCSR0A & (1 << UDRE0)));
	UDR0 = data; //copy character to be sent to UDR0
}

/* receive data through serial port, but it's only a char */
unsigned char USART_receive() {
	//check if there is unread data in the buffer
	while (!(UCSR0A & (1 << RXC0)));
	//read the data from the UDR0
	return UDR0;
}

// capture Flag
volatile uint32_t revTickAvg, revTickSig;
volatile uint32_t tickv, ticks;
volatile uint32_t revTick[SAMPRATE]; // Ticks per revolution
volatile uint32_t revCtr = 0; // Total elapsed revolutions
volatile uint16_t T1Ovs2; // Overflows for small rotations
// capture ISR
ISR(TIMER1_CAPT_vect) {
	cli();
	tickv = ICR1; // save duration of last revolution
	revTickSig = (uint32_t)(tickv) + ((uint32_t)T1Ovs2 * 0x10000L);
	revTick[revCtr] = revTickSig;
	revCtr++; // add to revolution count
	if (revCtr == SAMPRATE) {
		for (int i = 0; i < SAMPRATE; i++) {
			revTickAvg += (float) revTick[i];
		}
```

```
                revTickAvg /= SAMPRATE;
                rpm = (float) (60*1000000) / (144 * revTickAvg*0.0625);
                revCtr = 0;
        }
        TCNT1 = 0; // restart timer for next revolution
        T1Ovs2 = 0;
        sei();
}
// Overflow ISR
ISR(TIMER1_OVF_vect) {
        cli();
        // increment overflow counter
        T1Ovs2++;
        if (T1Ovs2 > 10) {
                STALL = 1;
                } else {
                STALL = 0;
        }
        sei();
}
ISR (ADC_vect)
{
        cli();
        doutOLD = dout;
        dout = ADC;
        // start ADC conversion=
        ADCSRA |=(1<<ADSC);
        sei();
}
ISR(USART0_RX_vect) {
        cli();
        static unsigned int i;
        char input = UDR0;
        switch (input) {
                case '0':
                case '1':
                case '2':
                case '3':
                case '4':
                case '5':
                case '6':
                case '7':
                case '8':
                case '9':
                numBuffer[i] = (int) input - ASCII_NUM;
                i++;
                break;
                case '\n':
                case '\r':
                numDigits = i;
                i = 0;
                if (numDigits > 0) { OVERRIDE = 1; }
```

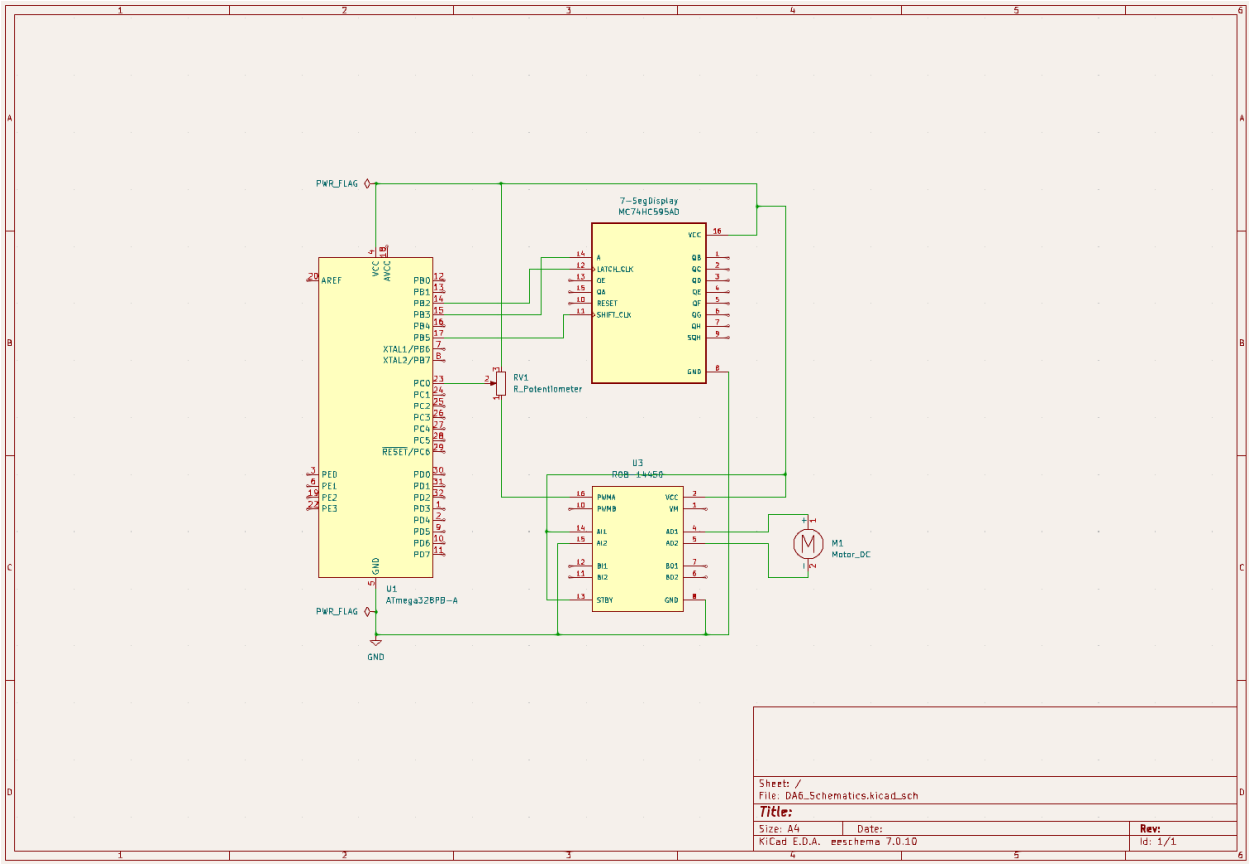
```
                break;
                /* default:*/
        }
        if (i == BUFFERSIZE) {
                i = 0;
        }
        sei();
}
void init_IO(void){
        //Setup IO
        //Set control pins as outputs
        DDRE |= (DATA | LATCH);
        DDRC |= (CLOCK);
        //Set control pins low
        PORTE &= ~(DATA | LATCH);
        PORTC &= ~CLOCK;
}
void init_SPI(void){
        //Setup SPI
        SPCR1 = (1<<SPE) | (1<<MSTR); //Start SPI as Master
}
void SPI_send(unsigned char byte){
        SPDR1 = byte; //Shift in some data
        while(!(SPSR1 & (1<<SPIF))); //Wait for SPI process to finish
}
void SEG_display(int thou, int hun, int ten, int one) {
        cli();
        PORTE &= ~LATCH;
        ten %= 10;
        one %= 10;
        // sending tens place
        SPI_send((unsigned char) SEG_digit[ten]);
        SPI_send((unsigned char) SEG_select[2]);
        PORTE |= LATCH;
        PORTE &= ~LATCH;
        _delay_ms(DELAY_SEG);
        // sending ones place
        SPI_send((unsigned char) SEG_digit[one]);
        SPI_send((unsigned char) SEG_select[3]);
        PORTE |= LATCH;
        PORTE &= ~LATCH;
        _delay_ms(DELAY_SEG);
        sei();
}

void setUP() {
        DDRD &= ~(1<<PIND2);    /* Make INT0 pin as Input */
        PORTD |= (1 << PIND2);    // turn On the Pull-up
        DDRD |= (1<<PIND6);                     /* Make OC0 pin as Output */
        EICRA |= (1 << ISC01);    // set INT0 to trigger to falling edge
        EIMSK |= (1 << INT0);     // Turns on INT0
        ADC_Init();                                             /* Initialize ADC */
```

```
	TCNT0 = 0;                                      /* Set timer0 count zero */
	TCCR0A |= (1<<WGM00)|(1<<WGM01)|(1<<COM0A1);
	TCCR0B |= (1<<CS00)|(1<<CS02);/* Set Fast PWM with Fosc/64 Timer0 clock */
	USART_init(MYUBRR);
}

// Initialize timer
void InitTimer1(void) {
	// Set PB0 as input
	DDRB &= ~(1 << DDB0);
	PORTB |= (1 << DDB0);
	// Set Initial Timer value
	TCNT1 = 0;
	////First capture on rising edge
	TCCR1A = 0;
	TCCR1B = (0 << ICNC1) | (1 << ICES1);
	TCCR1C = 0;
	// Interrupt setup
	// ICIE1: Input capture
	// TOIE1: Timer1 overflow
	TIFR1 = (1 << ICF1) | (1 << TOV1); // clear pending
	TIMSK1 = (1 << ICIE1) | (1 << TOIE1); // and enable
}
void StartTimer1(void) {
	// Start timer without pre-scaler
	TCCR1B |= (1 << CS10);
	// Enable global interrupts
	sei();
}

int main(void) {
	setUP();
	sei();
	InitTimer1();
	StartTimer1();
	ADC_Init();
	init_IO();
	init_SPI();
	DDRD &= ~((1 << PIND7) | (1 << PIND6) | (1 << PIND5));
	PORTD |= (1 << PIND7) | (1 << PIND6) | (1 << PIND5); //enable pull-up
	DDRB |= 0b00001110; //PB3, PB1, and PB2 as outputs
	//Fast PWM, non-inverted
	TCCR2A = (1<<COM2A1)|(1<<WGM21)|(1<<WGM20);
	TCCR2B = 0x02; //N = 8

	while(1) {
		pwmVal = (ADC_Read(0)/4);   // 1024/4 scales down to 256 ~ target PWM
		itoa(pwmVal,strPWM,10);
		USART_transmit("PWM Val: ");
		USART_transmit(strPWM);
		USART_transmitChar('\n');
```

```
                OCR0A = pwmVal;                                /* Read ADC and map it into 0-255 to write
in OCR0 register */
                _delay_ms(1000);
        }
}
```

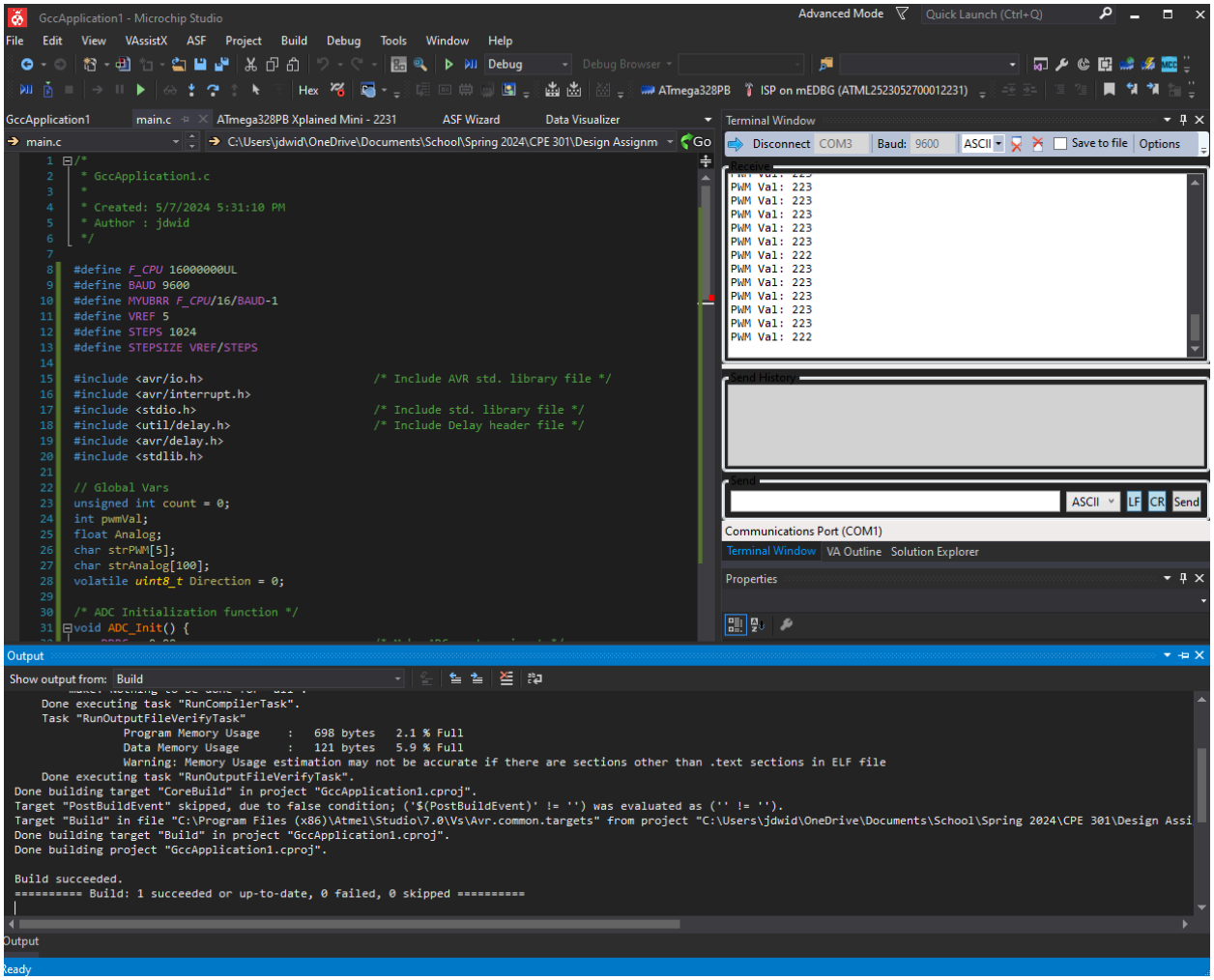
3. SCHEMATICS

Use KICAD schematics only (not required for DA1 simulation)

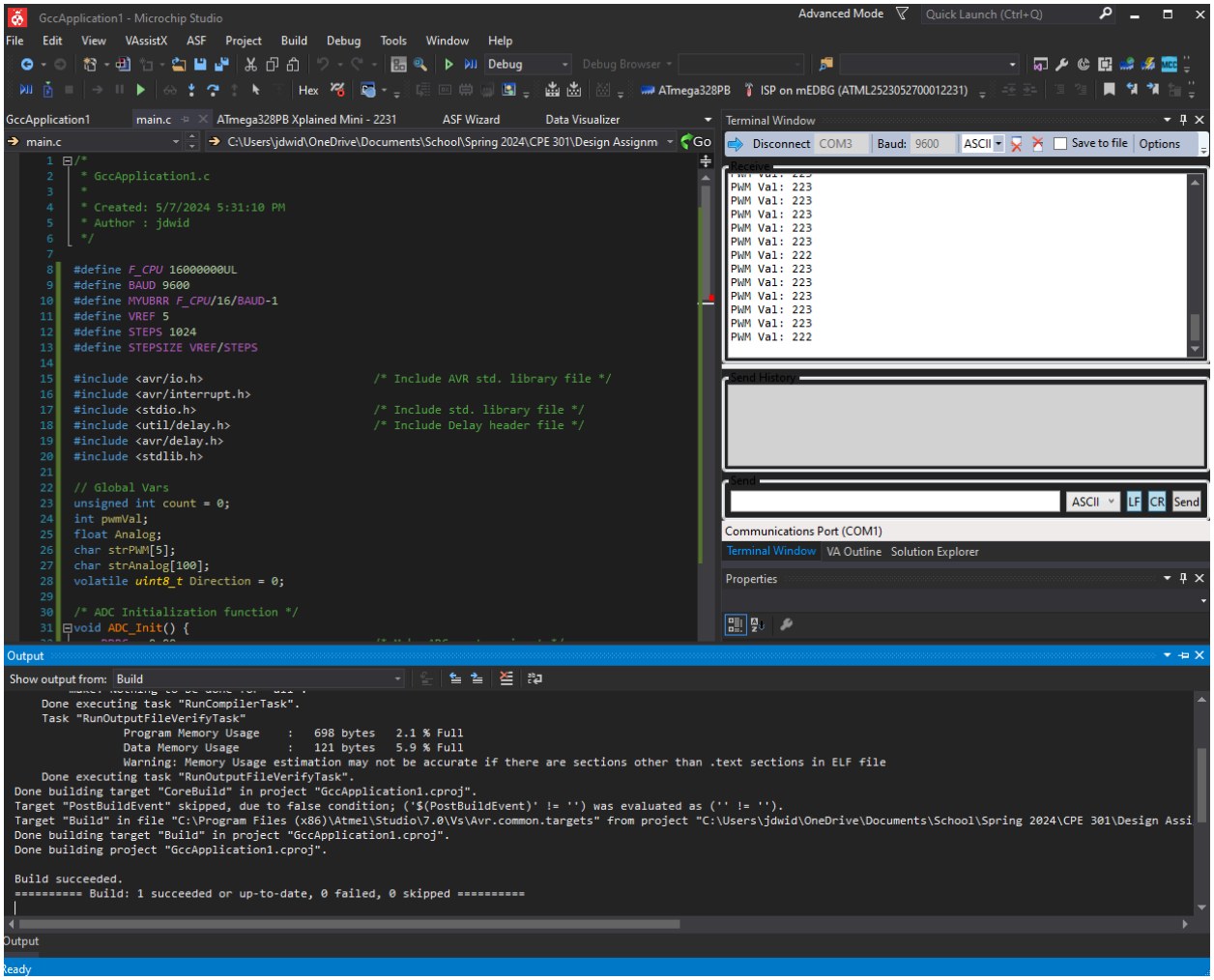


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

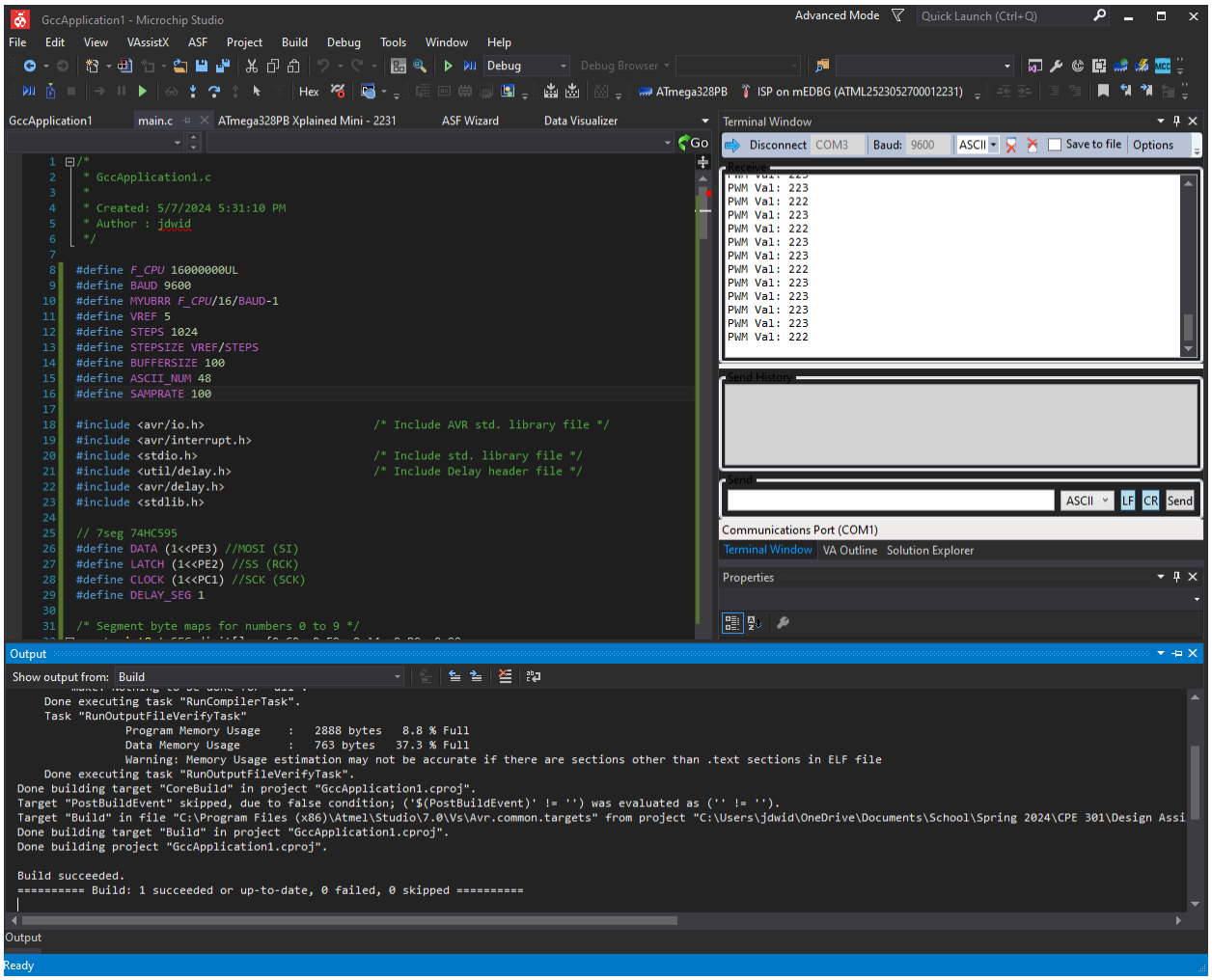
Task 1:


GccApplication1 - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
ATmega328PB
ISP on mEDBG (ATML2523052700012231)
GccApplication1
main.c
ATmega328PB Xplained Mini - 2231
ASF Wizard
Data Visualizer
C:\Users\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assignm
Go
/*
* GccApplication1.c
*
* Created: 5/7/2024 5:31:10 PM
* Author : jdwid
*/
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#include <avr/io.h> /* Include AVR std. library file */
#include <avr/interrupt.h>
#include <stdio.h> /* Include std. library file */
#include <util/delay.h> /* Include Delay header file */
#include <avr/delay.h>
#include <stdlib.h>
// Global Vars
unsigned int count = 0;
int pwmVal;
float Analog;
char strPWM[5];
char strAnalog[100];
volatile uint8_t Direction = 0;
/* ADC Initialization function */
void ADC_Init() {
Terminal Window
Disconnect COM3 Baud: 9600 ASCII Save to file Options
Receive
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 222
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 222
Send History
Send
ASCII LF CR Send
Communications Port (COM1)
Terminal Window VA Outline Solution Explorer
Properties
Output
Show output from: Build
Done executing task "RunCompilerTask".
Task "RunOutputFileVerifyTask"
Program Memory Usage : 698 bytes 2.1 % Full
Data Memory Usage : 121 bytes 5.9 % 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 "GccApplication1.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\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assi
Done building target "Build" in project "GccApplication1.cproj".
Done building project "GccApplication1.cproj".
Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
Output
Ready

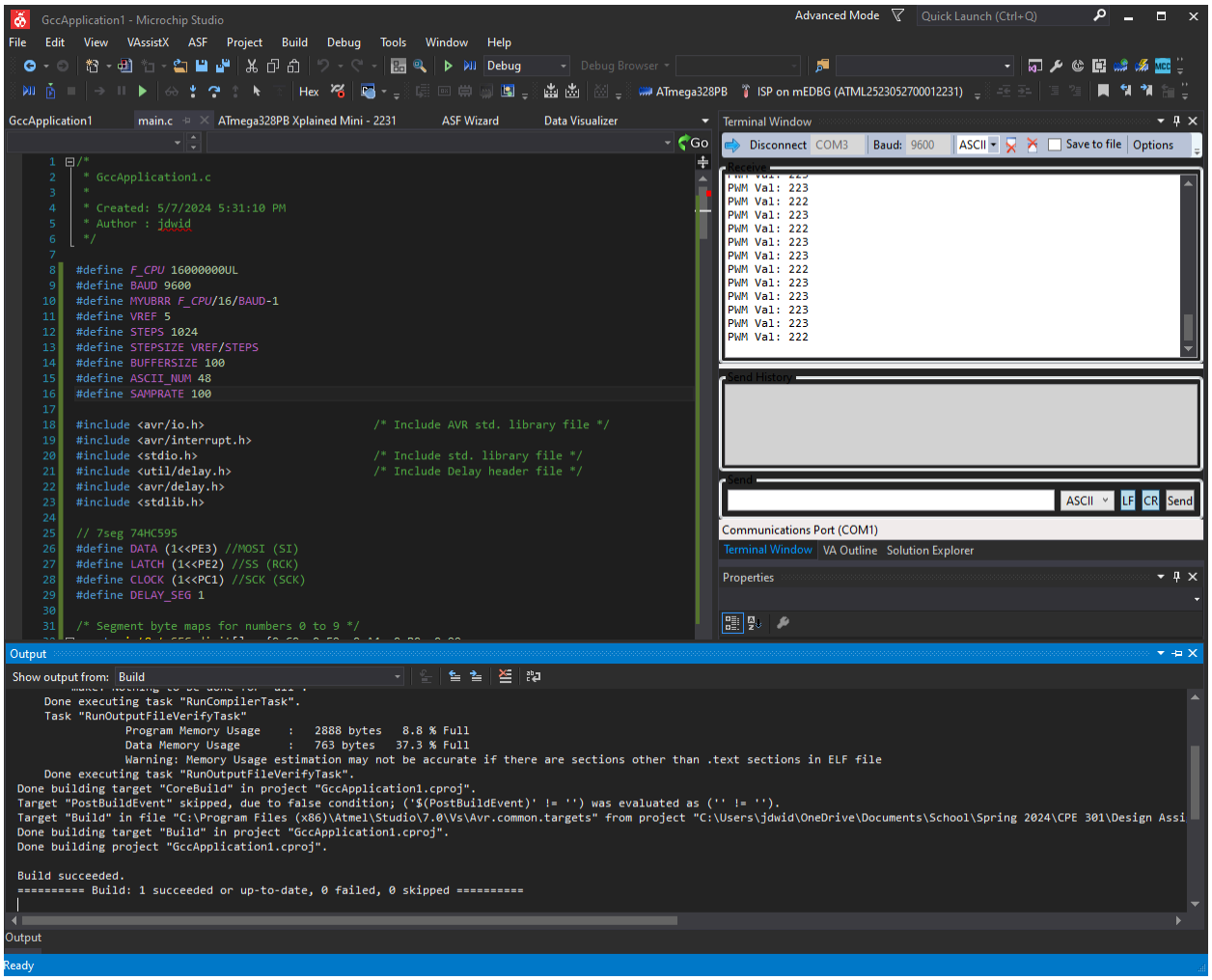
Task 2:


GccApplication1 - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
ATmega328PB
ISP on mEDBG (ATML2523052700012231)
GccApplication1
main.c
ATmega328PB Xplained Mini - 2231
ASF Wizard
Data Visualizer
C:\Users\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assignm
Go
/*
* GccApplication1.c
*
* Created: 5/7/2024 5:31:10 PM
* Author : jdwid
*/
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#include <avr/io.h> /* Include AVR std. library file */
#include <avr/interrupt.h>
#include <stdio.h> /* Include std. library file */
#include <util/delay.h> /* Include Delay header file */
#include <avr/delay.h>
#include <stdlib.h>
// Global Vars
unsigned int count = 0;
int pwmVal;
float Analog;
char strPWM[5];
char strAnalog[100];
volatile uint8_t Direction = 0;
/* ADC Initialization function */
void ADC_Init() {
Terminal Window
Disconnect COM3 Baud: 9600 ASCII Save to file Options
Receive
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 222
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 222
Send History
Send
ASCII LF CR Send
Communications Port (COM1)
Terminal Window VA Outline Solution Explorer
Properties
Output
Show output from: Build
Done executing task "RunCompilerTask".
Task "RunOutputFileVerifyTask"
Program Memory Usage : 698 bytes 2.1 % Full
Data Memory Usage : 121 bytes 5.9 % 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 "GccApplication1.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\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assi
Done building target "Build" in project "GccApplication1.cproj".
Done building project "GccApplication1.cproj".
Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
Output
Ready

Task 3:

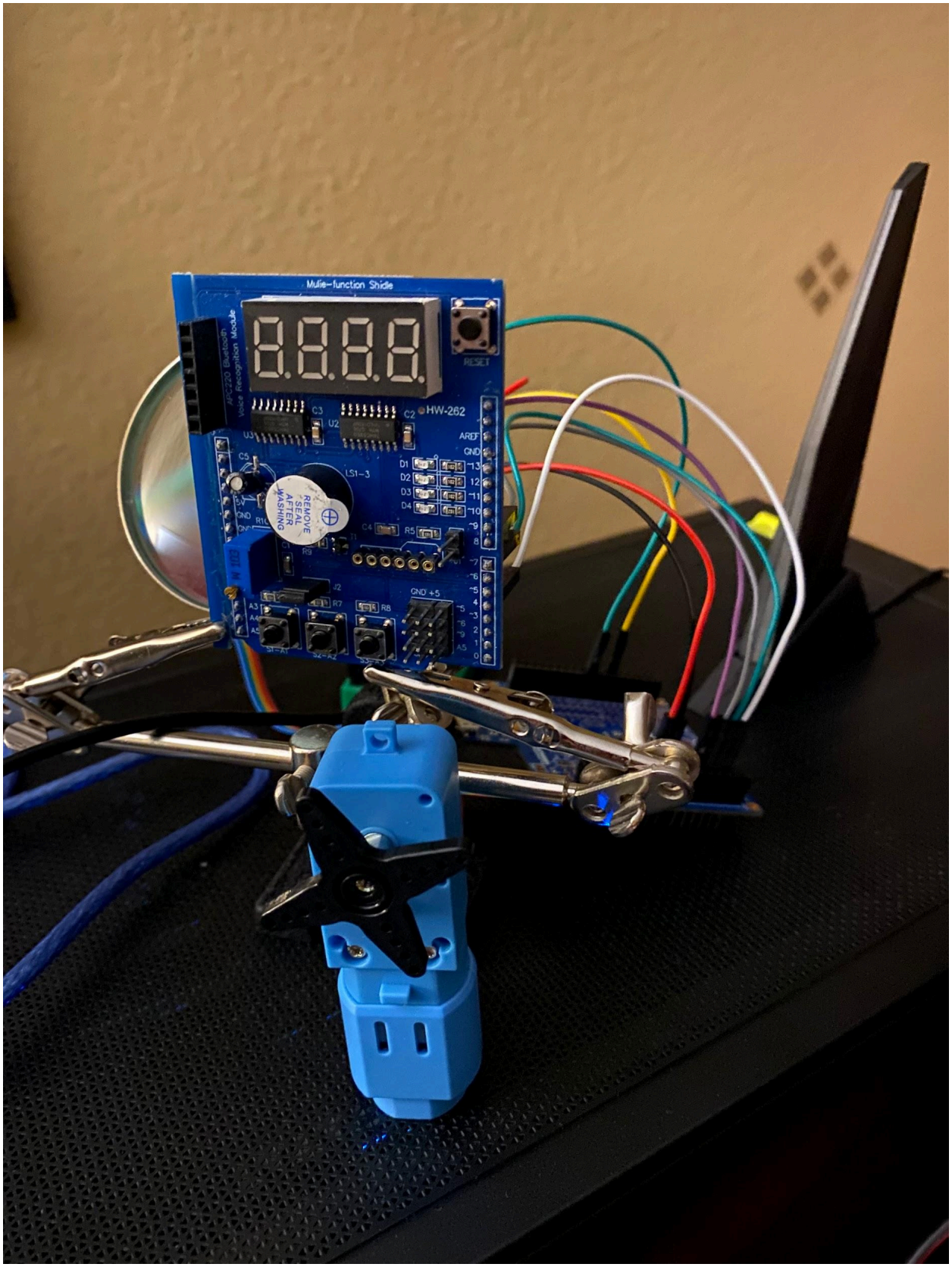

GccApplication1 - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
Hex
ATmega328PB
ISP on mEDBG (ATML2523052700012231)
GccApplication1
main.c
ATmega328PB Xplained Mini - 2231
ASF Wizard
Data Visualizer
Go
/*
* GccApplication1.c
*
* Created: 5/7/2024 5:31:10 PM
* Author : jdwid
*/
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#define BUFFERSIZE 100
#define ASCII_NUM 48
#define SAMPRATE 100
#include <avr/io.h> /* Include AVR std. library file */
#include <avr/interrupt.h>
#include <stdio.h> /* Include std. library file */
#include <util/delay.h> /* Include Delay header file */
#include <avr/delay.h>
#include <stdlib.h>
// 7seg 74HC595
#define DATA (1<<PE3) //MOSI (SI)
#define LATCH (1<<PE2) //SS (RCK)
#define CLOCK (1<<PC1) //SCK (SCK)
#define DELAY_SEG 1
/* Segment byte maps for numbers 0 to 9 */
Terminal Window
Disconnect COM3 Baud: 9600 ASCII Save to file Options
Receive
PWM Val: 223
PWM Val: 222
PWM Val: 223
PWM Val: 222
PWM Val: 223
PWM Val: 223
PWM Val: 222
PWM Val: 223
PWM Val: 223
PWM Val: 223
PWM Val: 222
Send History
Send
ASCII LF CR Send
Communications Port (COM1)
Terminal Window VA Outline Solution Explorer
Properties
Output
Show output from: Build
Done executing task "RunCompilerTask".
Task "RunOutputFileVerifyTask"
Program Memory Usage : 2888 bytes 8.8 % Full
Data Memory Usage : 763 bytes 37.3 % 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 "GccApplication1.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\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assi
Done building target "Build" in project "GccApplication1.cproj".
Done building project "GccApplication1.cproj".
Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
Output
Ready

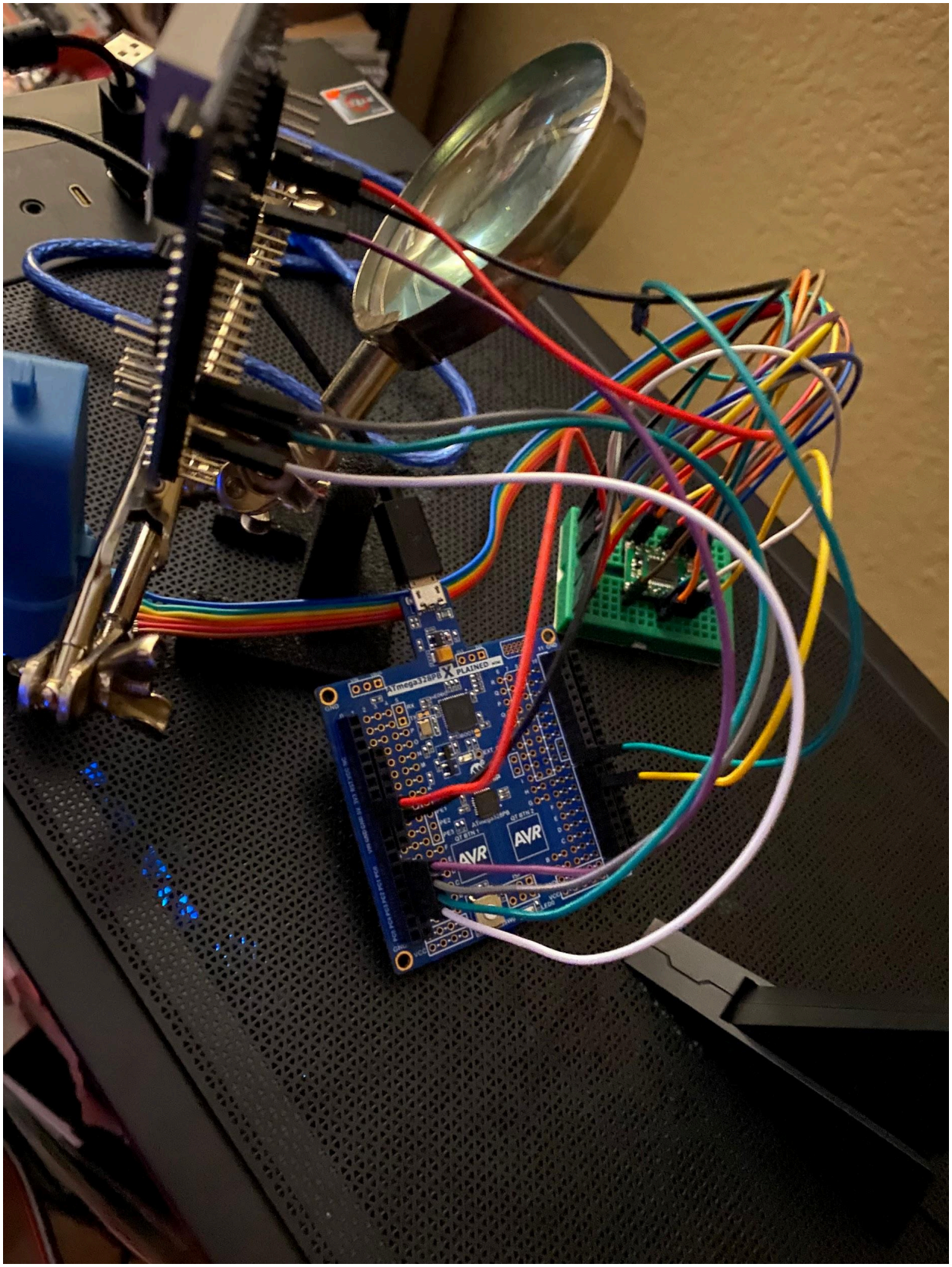
Task 4:



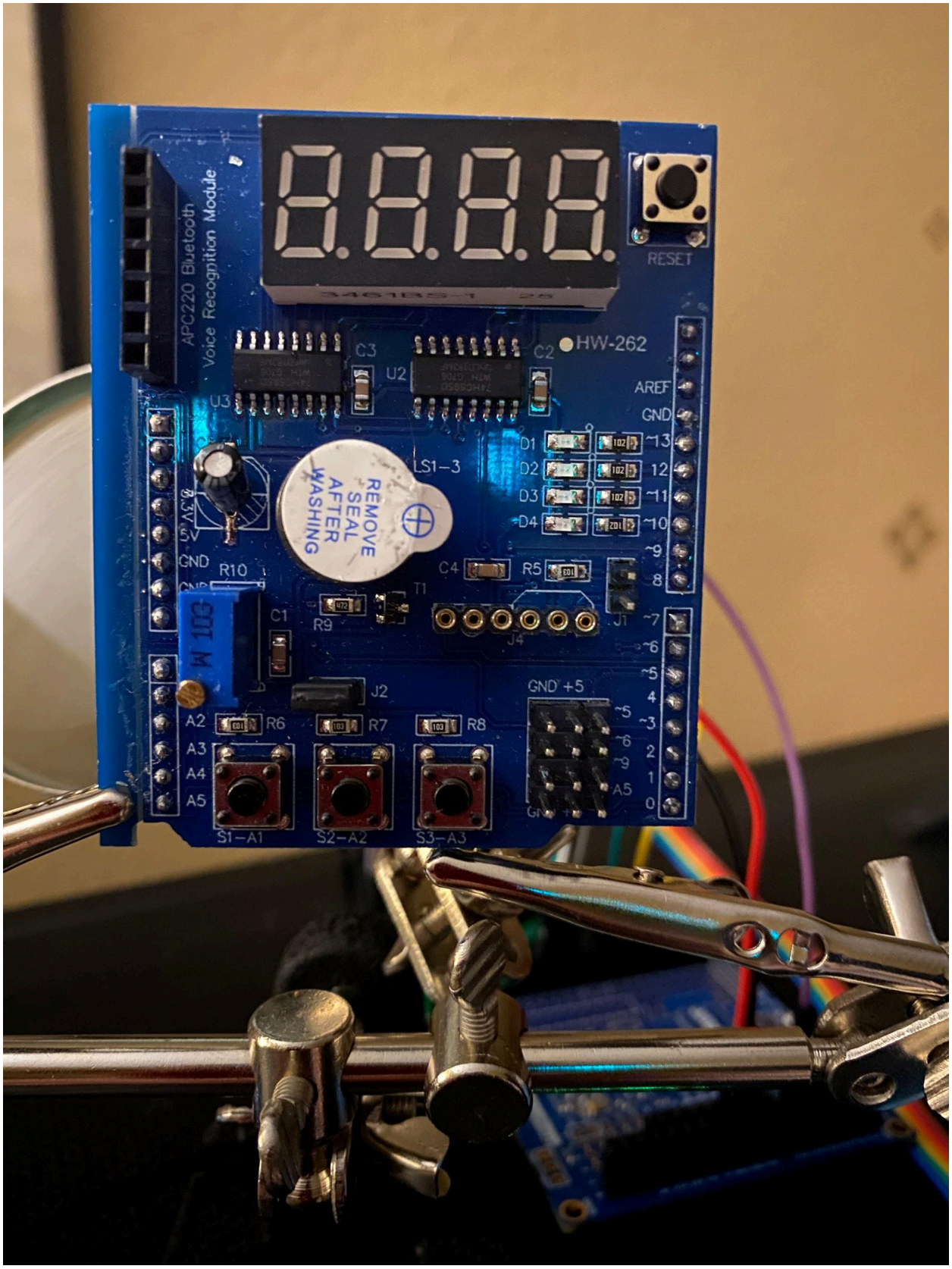
5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

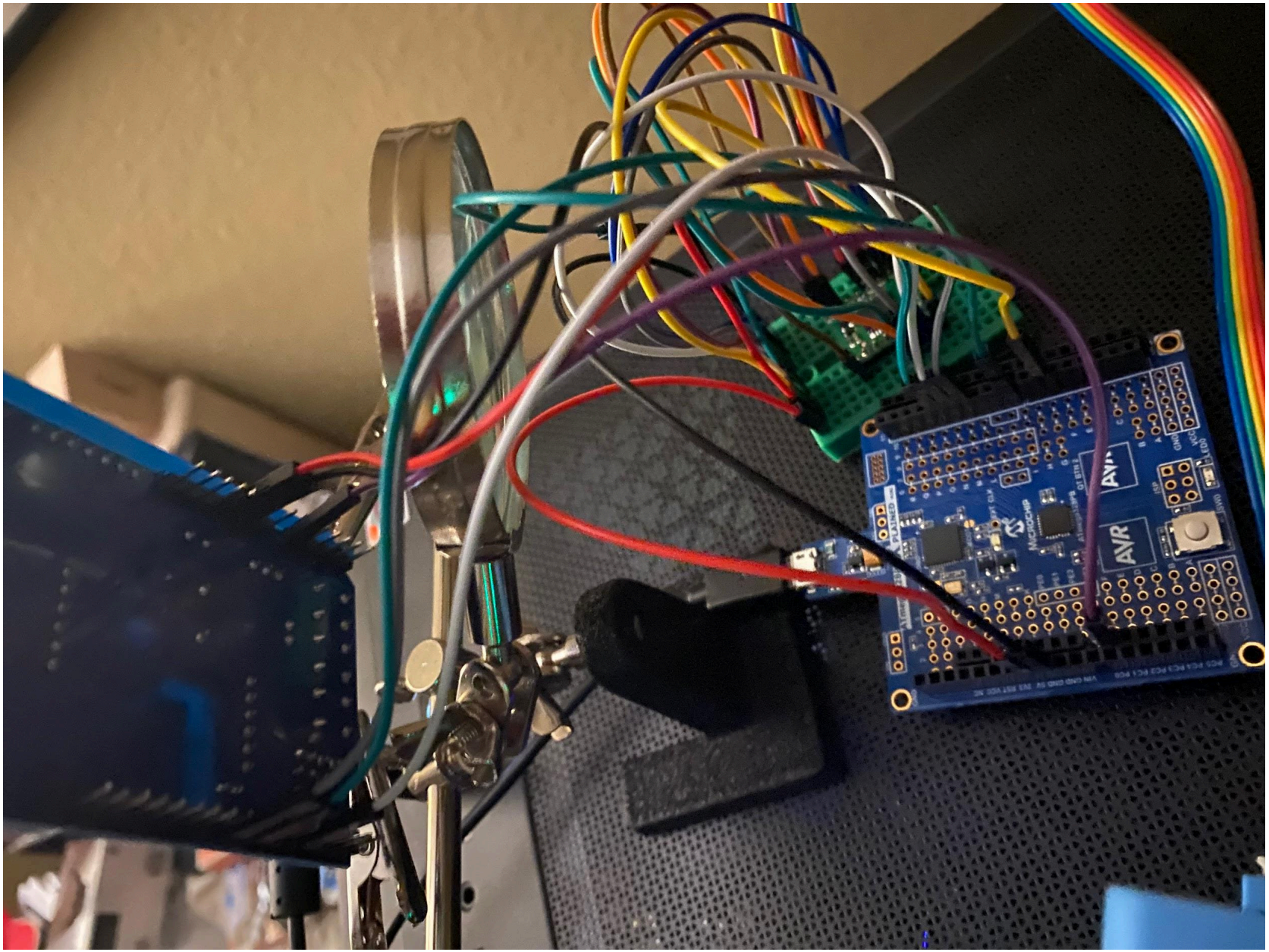
Task 1:


Mulie-function Shide
RESET
HW-262
REMOVE
SEAL
BEFORE
WASHING

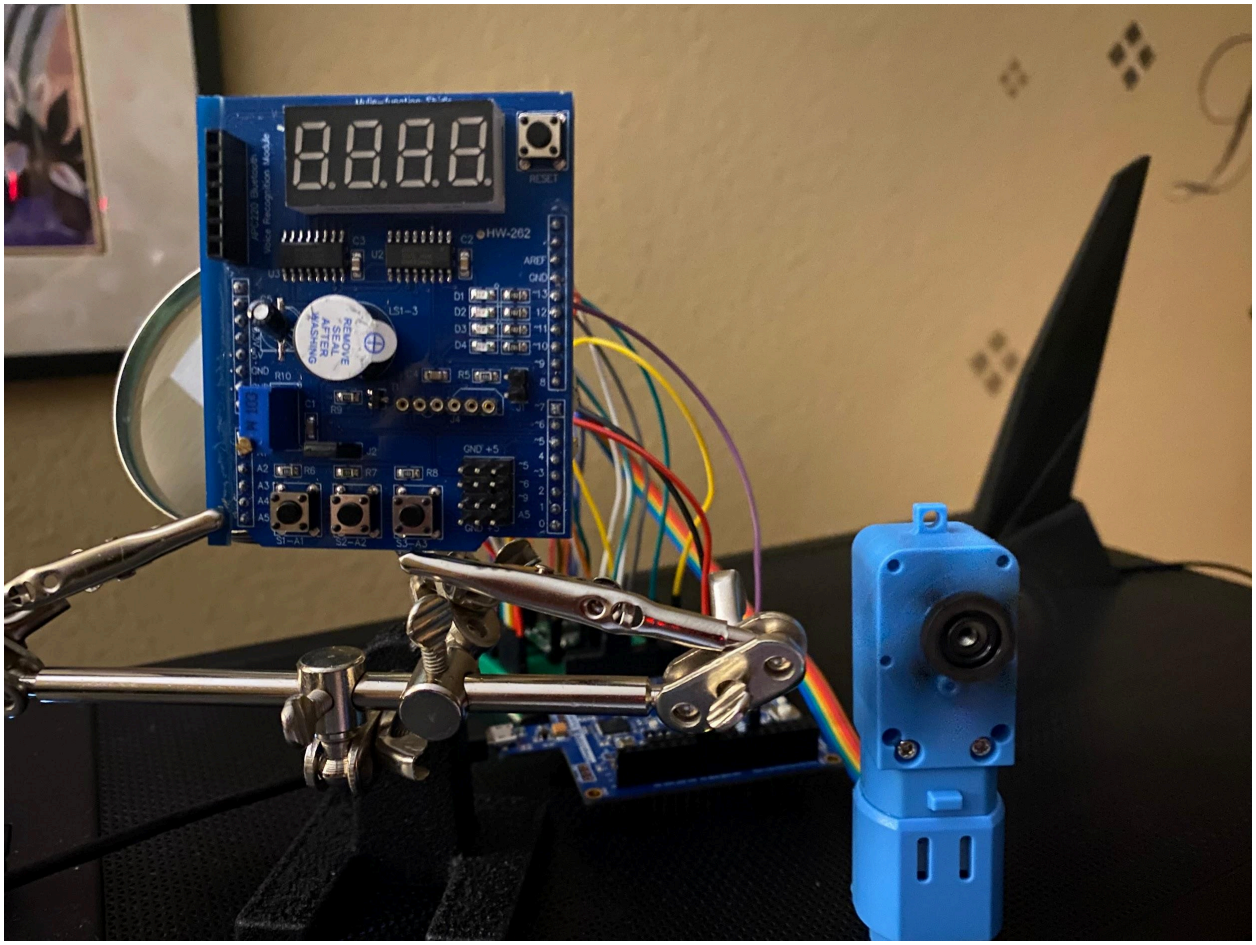

AVR
AVR

Task 2:

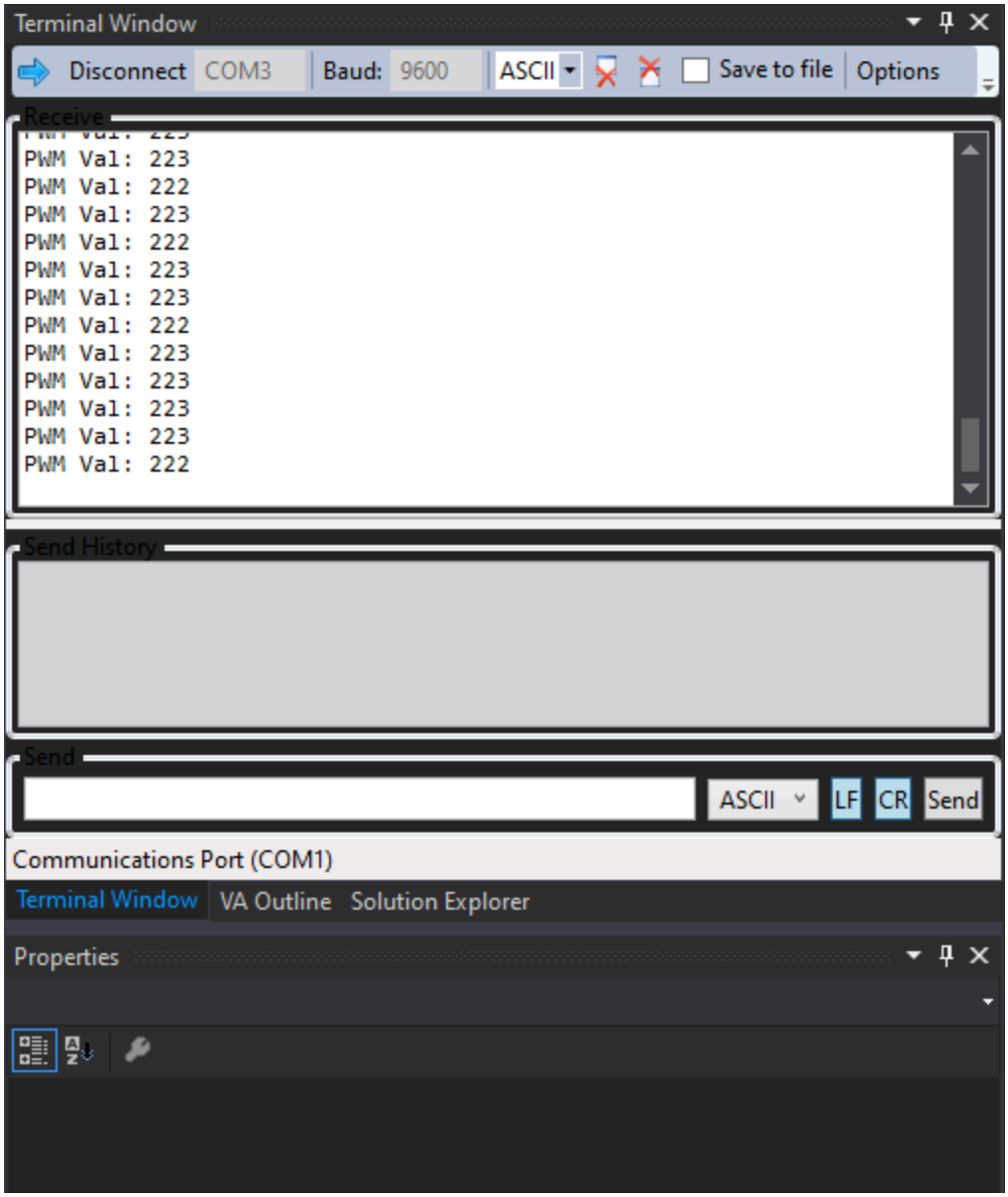

APC220 Bluetooth
Voice Recognition Module
RESET
HW-262
REMOVE SEAL AFTER WASHING
LS1-3
S1-A1
S2-A2
S3-A3
AREF
GND
GND +5


AVR
MICROCHIP

Task 3:



Task 4:



6. VIDEO LINKS OF EACH DEMO

Task 1: https://youtu.be/2AUba3bbLXQ

Task 2: https://youtu.be/2AUba3bbLXQ

Task 3: https://youtu.be/CkkoScbBtRY

Task 4: https://youtu.be/CkkoScbBtRY

7. GITHUB LINK OF THIS DA

Task 1:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%206

Task 2:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%206

Task 3:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%206

Task 4:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%206

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

"*This assignment submission is my own, original work*".
Johnathan Widney