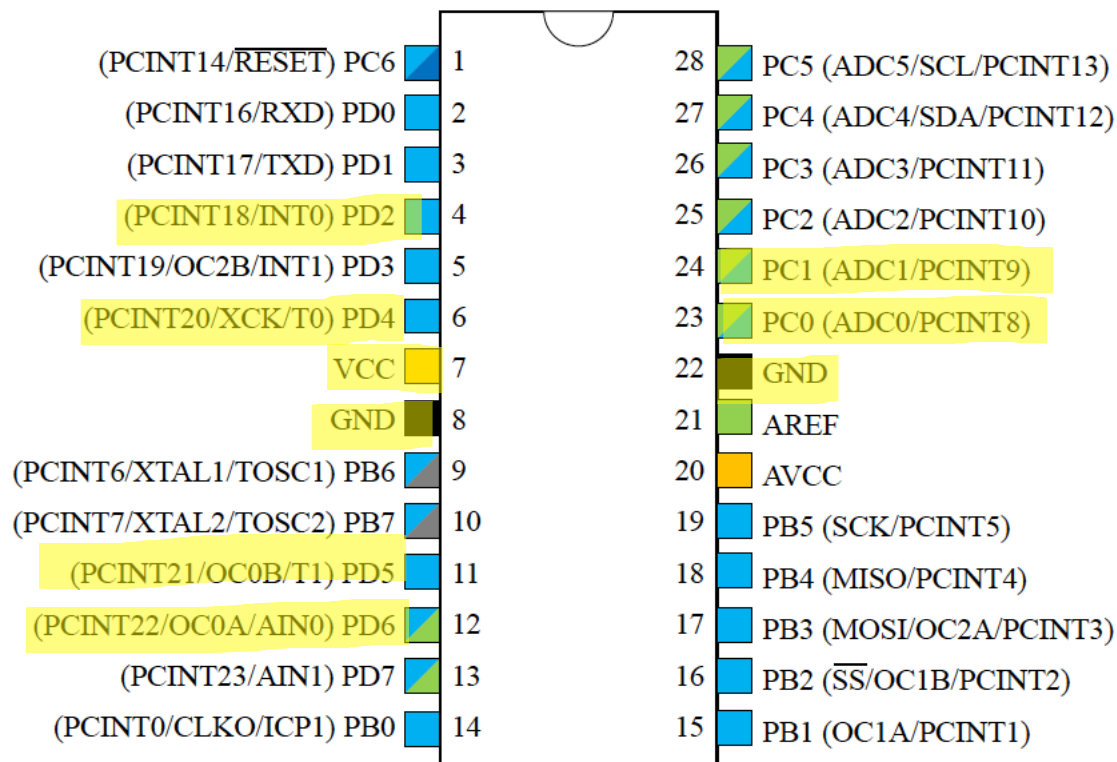


Design Assignment 4A

Student Name: Elmer Mejia
 Student #: 5003824808
 Student Email: mejiae4@unlv.nevada.edu
 Primary Github address: https://github.com/cpemejia/design_assignments.git
 Directory: design_assignments

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

-Atmega328pb - Multi-function shield -KiCad
 -Microchip Studio -Potentiometer - DC motor w/ TB6612



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

```
/*
 * DA_4A.c
 *
 * Created: 4/3/2021 7:39:15 PM
 * Author : ElmerOMejia
 */

/*
 * DA_2BC.c
 *
 * Created: 2/27/2021 11:57:51 PM
 * Author : ElmerOMejia
 */

#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#define AIN2 5
#define AIN1 4

// PD6 PWMA
// PD5 AIN2
// PD4 AIN1
// ADC0/PC0

volatile uint8_t click = 0;
volatile unsigned int newADC;

uint16_t read_adc(uint8_t channel);
void adc_init(void);

int main(void)
{
    //speed PWMA
    DDRD |= (1<<6);           // set PD6 output
    // direction AIN2/AIN1
    DDRD |= (1<<5);           // set PD5 output
    DDRD |= (1<<4);           // set PD4 output
    PORTD &= ~(1<<AIN2);
    PORTD &= ~(1<<AIN1);

    // set INT0
    PORTD = 1<<2; // pull up enable
    EICRA = 0x02;
    EIMSK = (1<<INT0); // enable interrupt 0
    sei(); // enable interrupts

    adc_init();
    // Set timer0
```

```

OCR0A = 25; // about 10% initial
// Fast PWM, non inverted
TCCR0A |= ((1<<COM0A1) | (1<<WGM01) | (1<<WGM00));
TCCR0B |= (1<<CS01); // n = 8;

while (1)
{
    if (click != 0){
        //_delay_ms(20);
        PORTD |= (1<<AIN1); // H
        PORTD &= ~(1<<AIN2); // L

    }
    else {
        //_delay_ms(20);
        PORTD &= ~(1<<AIN1); // L
        PORTD &= ~(1<<AIN2); // L
    }

    newADC = (read_adc(0)/4); // 10 bit ADC to 8 bit

    if (newADC < 25){ // MIN OCRA
        OCR0A = 25;
    }
    else if (newADC > 243){ // MAX OCRA
        OCR0A = 243;
    }
    else {
        OCR0A = newADC; // OCRA is equal adc val
    }

}

}

ISR (INT0_vect) // ISR for INT0
{
    click = ~(click);
    _delay_ms(50);
}

void adc_init(void)
{
    /** Setup and enable ADC */
    ADMUX = (0<<REFS1) | // Reference Selection Bits
    (1<<REFS0) | // AVcc - external cap at AREF
    (0<<ADLAR) | // ADC Left Adjust Result
    (0<<MUX2) | // Analog Channel Selection Bits
    (0<<MUX1) | // ADC0 (PC0 PIN23)
    (0<<MUX0);
    ADCSRA = (1<<ADEN) | // ADC ENable
    (0<<ADSC) | // ADC Start Conversion
    (0<<ADATE) | // ADC Auto Trigger Enable
    (0<<ADIF) | // ADC Interrupt Flag
    (0<<ADIE) | // ADC Interrupt Enable
    (1<<ADPS2) | // ADC Prescaler Select Bits
    (0<<ADPS1) |

```

```

    (1<<ADPS0);
}

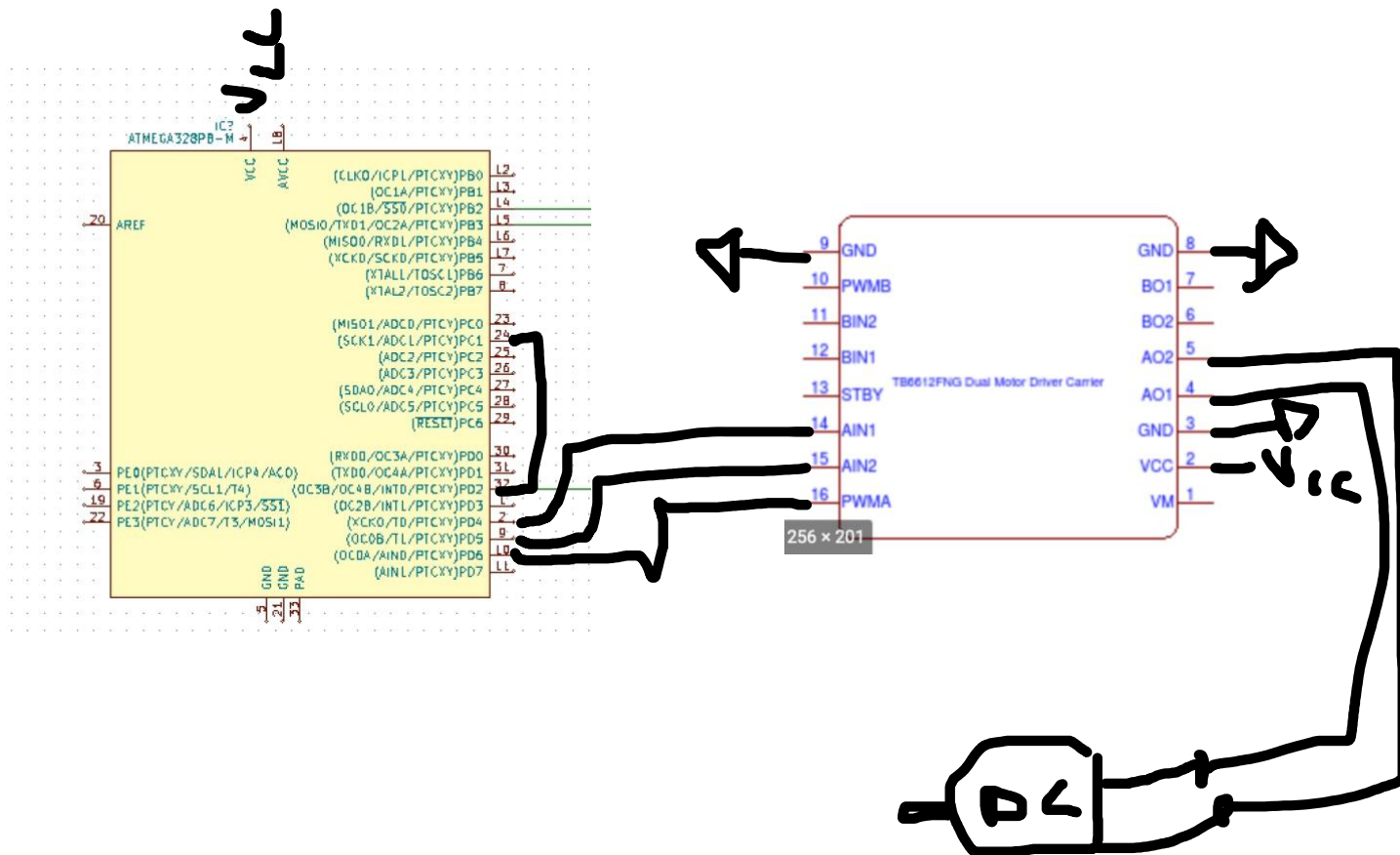
```

```

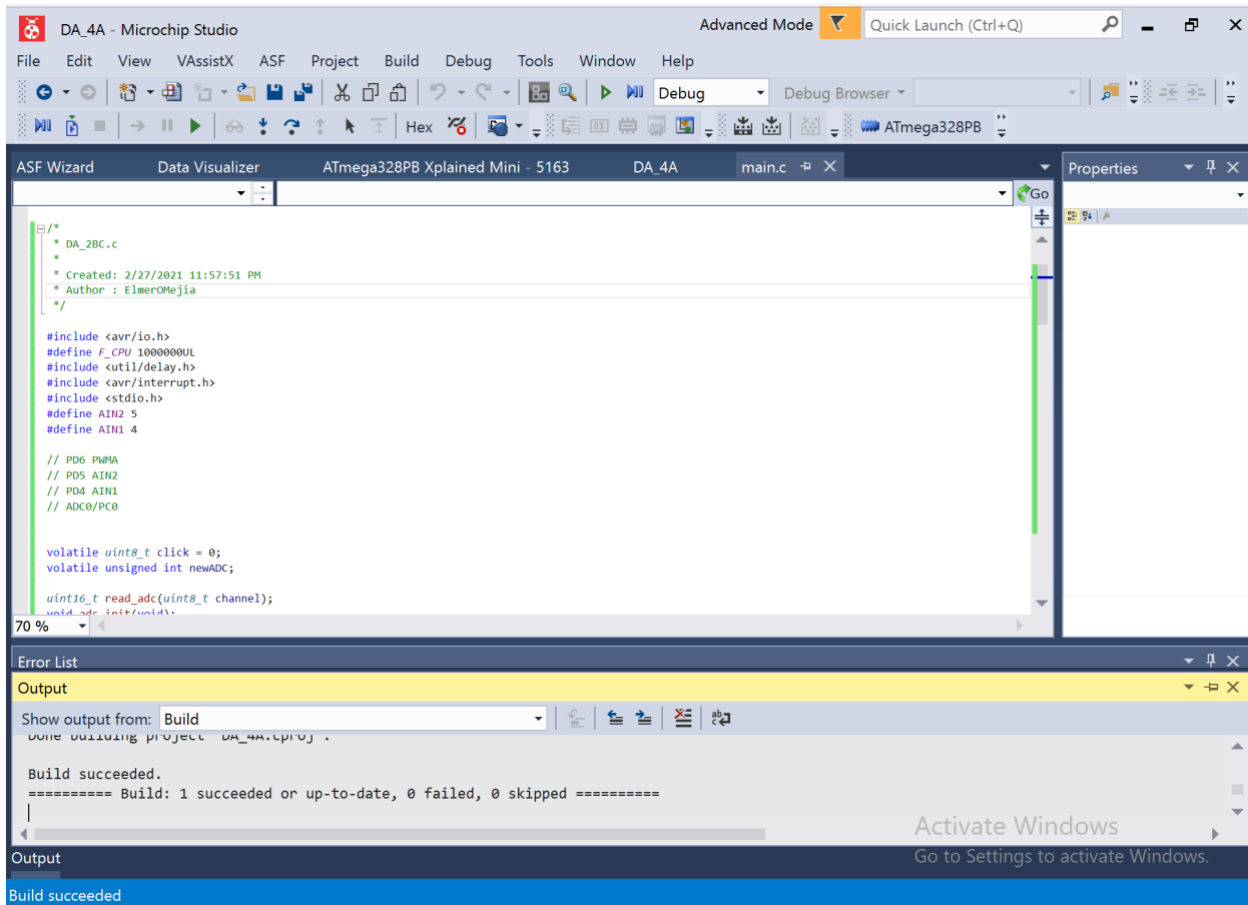
uint16_t read_adc(uint8_t channel){
    ADMUX &= 0xF0;           //Clear the older channel that was read
    ADMUX |= channel;        //Defines the new ADC channel to be read
    ADCSRA |= (1<<ADSC);     //Starts a new conversion
    while(ADCSRA & (1<<ADSC)); //Wait until the conversion is done
    return ADCW;              //Returns the ADC value of the chosen channel
}

```

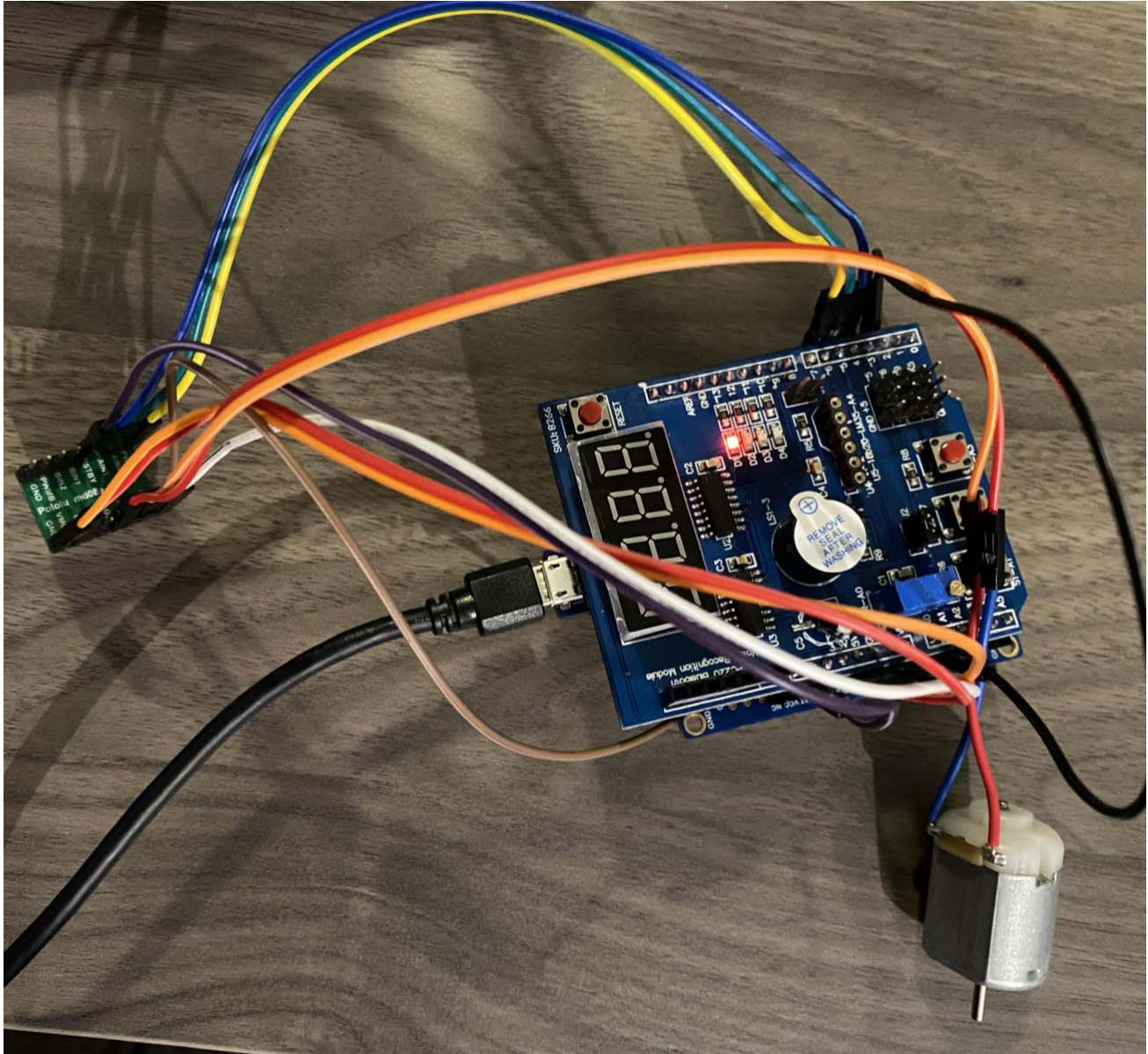
3. SCHEMATICS



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



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

[Design Assignments](#)

7. GITHUB LINK OF THIS DA

[github](#)

Student Academic Misconduct Policy

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

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

Elmer Mejia