```
/*****************************************************************************
 *
 * Project name: PWM_&_ADC.c
 * Created: 10/11/2023 6:41:30 PM
 * Author: Caiden Moreno
 * Overview: This program operates by utilizing two pushbuttons to switch back
 * and forth into PWM mode and ADC mode with an LED flashing 500ms to indicate which mode
*  the user is in.
 * Hardware:
 * Arduino ATmega2560 micro controller
 * Inputs:
 * Start PushButton  = PINA.0
 * Stop  PushButton  = PINA.1
 * Mode Switch       = PINA.4
 * Potentiometer     = PINL.0
 * Outputs:
 * Mode LED          = PINB.7
 * DC MOTOR          = PINB.5

*****************************************************************************/

#include <avr/io.h>
#include "Debugger.h"
#define startButton (PINA & 0x01) //PINA.0
#define stopButton  (PINA & 0x02) //PINA.1

#define modeSwitch (PINL & 0x01) //PINL.0

//function prototypes
void io_init(void);
void delay_ms(uint8_t ms);
void init_timer0(void);
void PWM_start(void);
void PWM_end(void);
void setupPWM(void);
void PWM_init(void);
void ramp_up_delay_n_steps(uint16_t start, uint16_t end, uint16_t ms_time, uint8_t
num_steps);
uint16_t ADCtenbitvalue(uint16_t channel);
void init_ADC(void);


////////////////////////////////////////////////////////////////////////////////
//checkoff 1
//objective: Demonstrate pushbuttons work with debugger
////////////////////////////////////////////////////////////////////////////////
int main(void)
{
	io_init(); //call initialized io ports function
	initDebug(); //call debug function

    while (1)
    {
		while (PINA & (1 << PA0))
		{
			// If PINA.0 is high, turn on LED13 (PORTB.5)
			PORTB = (0x80); //turn on LED13
```

```
                        _delay_ms(500); //time delay of 500ms
                }

                while (PINA & (1 << PA1))
                {
                        // If PINA.1 is high, turn off LED13 (PORTB.5)
                        PORTB = ~(0x80);//turn off LED13
                }

    }

}

void io_init(void) //initialize io ports
{
        //inputs
        DDRA=0x00;
        PORTA=0xFF;

        //outputs
        DDRB=(0xFF); //LED 13 set as output
        PORTB=(0x00); //turn off LED at initialization

        //potentiometer output
        DDRL=0x00;
        PORTL=0xFF;
}


////////////////////////////////////////////////////////////////////////////////
//checkoff 2
//objective: LED flashing on and off every 500ms
////////////////////////////////////////////////////////////////////////////////
int main(void)
{
        //call functions
        io_init();
        initDebug();
        init_timer0();

        while (1)
        {
                PORTB ^=(0x80); //toggle pin 7 LED13 on and off
                delay_ms(500); //time delay of 500ms
        }

}

void io_init(void) //initialize io ports
{
        //inputs
        DDRA=0x00;
        PORTA=0xFF;

        //outputs
        DDRB=(0xFF); //LED 13 set as output
        PORTB=(0x00); //turn off LED at initialization
```

```
	//potentiometer output
	DDRL=0x00;
	PORTL=0xFF;
}


void init_timer0(void)
{
    TCNT0 =  240;  //Set TCNT0 to 240 for a 1ms delay
	TCCR0A = 0; //normal mode
	TCCR0B = 0;
}
void delay_ms(uint8_t ms)
{
	for (uint8_t i = ms; i > 0; i--)// Loop to create a delay of 'ms'
	{
		TCCR0B = (1<<CS02) | (1<<CS00);  // Set prescaler to 1024
		while((TIFR0 & (1<<TOV0))==0); // Wait for Timer 0 overflow flag
		TCCR0B =0; // Stop Timer 0
		TIFR0 |= (1<<TOV0); // Clear the Timer 0 overflow flag
		TCNT0 =  240;  //set TCNT0 to 240 for a 1ms delay
	}
}



////////////////////////////////////////////////////////////////////////////////
//checkoff 3
//Objective: PWM speed profile
////////////////////////////////////////////////////////////////////////////////
int main(void)
{
	//call functions
	io_init();
	initDebug();
	init_timer0();


	while (1)
	{
		while((startButton == 0x01)==0)
		{
			// do nothing
		}

		PWM_start(); //enable PWM
		setupPWM(); //initialize PWM
		uint16_t start = .1*511;//10% for 9-bit Fast PWM mode
		uint16_t end = .9*511;//90% for 9-bit Fast PWM mode
		ramp_up_delay_n_steps(start,end,8000,8);

		while((stopButton == 0x02)==0)
		{
			// do nothing
		}
		PWM_end(); //disable PWM

	}
```

```
}

void io_init(void) //initialize io ports
{
	//inputs
	DDRA=0x00;
	PORTA=0xFF;

	//potentiometer output
	DDRL=0x00;
	PORTL=0xFF;
}
void init_timer0(void)
{
	TCNT0 =  240;  //Set TCNT0 to 240 for a 1ms delay
	TCCR0A = 0; //normal mode
	TCCR0B = 0; //reset timer
}
void delay_ms(uint8_t ms)
{
	for (uint8_t i = ms; i > 0; i--)// Loop to create a delay of 'ms'
	{
		TCCR0B = (1<<CS02) | (1<<CS00);  // Set prescaler to 1024
		while((TIFR0 & (1<<TOV0))==0); // Wait for Timer 0 overflow flag
		TCCR0B =0; // stop the timer
		TIFR0 |= (1<<TOV0); // Clear the Timer 0 overflow flag
		TCNT0 =  240;  //set TCNT0 to 240 for a 1ms delay
	}
}
void ramp_up_delay_n_steps(uint16_t start, uint16_t end, uint16_t ms_time, uint8_t
num_steps)
{
	// Calculate the change in duty cycle per step
	uint16_t duty_cycle_change = (end - start)/ num_steps;
	// Calculate the time for each step
	uint16_t step_time = ms_time / num_steps;

	for (uint16_t i = 0; i <= num_steps; i++)
	{
		OCR1A=0;
		OCR1A =((i * duty_cycle_change)+start);

		// Delay for the specified time
		delay_ms(step_time);
		delay_ms(step_time);
		delay_ms(step_time);
		delay_ms(step_time);
	}

}


//Enable PWM on OC1A
void PWM_start(void)
{
	TCCR1A = (1 << COM1A1); // enable compare output mode for OC1A
}
```

```
//Disable PWM on OC1A
void PWM_end(void)
{
	OCR1A=0; //turn off PWM signal
	TCCR1A = (TCCR1A & ~0x80);//disable compare output mode for OC1A
}

void setupPWM(void)
{
	//portb.5 set as output
	DDRB |= 0x20;

	// Set Timer 1 in fast PWM mode, 9-bit
	TCCR1A |= (0 << WGM10) | (1 << WGM11);
	TCCR1B = (0 << WGM13) | (1 << WGM12) | (1 << CS11); // Prescaler set to 8

	//set frequency to 3900Hz
	ICR1 = 511;
}

////////////////////////////////////////////////////////////////////////////////
//checkoff 4
//Objective: ADC adjsut PWM mode
////////////////////////////////////////////////////////////////////////////////
int main(void)
{
	//call functions
	io_init();
	initDebug();
	init_timer0();


	while (1)
	{

		if (modeSwitch==0x01){

		PWM_start();//enable PWM
		setupPWM(); //intialize PWM

		init_ADC();
		uint16_t rawadc = ADCtenbitvalue(0); //channel 0
		OCR1A = rawadc*4;//calculated conversion
		PORTB &= ~(1 << PB7);//turn off LED
	    }

		else if(modeSwitch!=0x01){

		PWM_end();
		while((startButton == 0x01)==0){
			PORTB ^= (1 << 7);  //toggle pin 7 LED13 on and off
			delay_ms(500); //time delay of 500ms
		}

		PWM_start();
		setupPWM();
		uint16_t start = .1*511;//10%
```

```
                uint16_t end = .9*511;//90%
                ramp_up_delay_n_steps(start,end,8000,8);

                while((stopButton == 0x02)==0)
                {
                        // do nothing
                }
                PWM_end();
                }

        }
}
void io_init(void) //initialize io ports
{
        // set PortB as an output
        DDRB=(0xFF); //LED 13 set as output
        PORTB=(0x00); //turn off LED at initialization

        // set PORTA as an input for pushbuttons
        DDRA=0x00;
        PORTA=0xFF;
        // set PORTL as an input for switch
        DDRL = 0x00;
        PORTL=0xFF;

        // set PORTF.0 as an input for potentiometer
        DDRF  = 0x00;
        PORTF = 0x00;
}

void init_ADC(void){
        DDRF &= (1<<PF0); //set portf to input
        ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);//enable with a prescaler of
128

        ADMUX = (1<<REFS0)|(1<<ADLAR);//5v reference, left adjusted

        ADCSRB = 0x00;
}

uint16_t ADCtenbitvalue(uint16_t channel) {


        uint16_t adc_read = 0;
        ADMUX=(ADMUX & 0xe0) | channel;//select channel
        ADCSRA |= (1<<ADSC); //start conversion
        while((ADCSRA&(1<<ADIF))==0){ //wait until adif turns to 1 when conversion is
completed

        }

        //adc_read =ADCH;// ADCH number
         adc_read = ADCL; // Read the low byte first
         adc_read |= (ADCH << 8); // Read the high byte and combine
        ADCSRA |= (1<<ADIF); //reset adif flag
        adc_read = (adc_read * 1023)/511; //calculated conversion for 9-bit resolution

        return adc_read; //returns adc_read value
```

```
}

void init_timer0(void)
{
	TCNT0 =  240;  //Set TCNT0 to 240 for a 1ms delay
	TCCR0A = 0; //normal mode
	TCCR0B = 0; //reset timer
}
void delay_ms(uint8_t ms)
{
	for (uint8_t i = ms; i > 0; i--)// Loop to create a delay of 'ms'
	{
		TCCR0B = (1<<CS02) | (1<<CS00);  // Set prescaler to 1024
		while((TIFR0 & (1<<TOV0))==0); // Wait for Timer 0 overflow flag
		TCCR0B =0; // stop the timer
		TIFR0 |= (1<<TOV0); // Clear the Timer 0 overflow flag
		TCNT0 =  240;  //set TCNT0 to 240 for a 1ms delay
	}
}
void ramp_up_delay_n_steps(uint16_t start, uint16_t end, uint16_t ms_time, uint8_t
num_steps)
{
	// Calculate the change in duty cycle per step
	uint16_t duty_cycle_change = (end - start)/ num_steps;
	// Calculate the time for each step
	uint16_t step_time = ms_time / num_steps;

	for (uint16_t i = 0; i <= num_steps; i++)
	{
		OCR1A=0;
		OCR1A =((i * duty_cycle_change)+start);

		// Delay for the specified time
		delay_ms(step_time);
		delay_ms(step_time);
		delay_ms(step_time);
		delay_ms(step_time);
	}

}

//Enable PWM on OC1A
void PWM_start(void)
{
	TCCR1A = (1 << COM1A1); // enable compare output mode for OC1A
}

//Disable PWM on OC1A
void PWM_end(void)
{
	OCR1A=0; //turn off PWM signal
	TCCR1A = (TCCR1A & ~0x80);//disable compare output mode for OC1A
}

void setupPWM(void)
{
	//portb.5 set as output
	DDRB |= 0x20;
```

```
        // Set Timer 1 in fast PWM mode, 9-bit
        TCCR1A |= (0 << WGM10) | (1 << WGM11);
        TCCR1B = (1 << WGM12) | (1 << CS11); // Prescaler set to 8

        //set frequency to 3900Hz
        ICR1 = 511;
}
```