```
/*********************************************************************************
*
* Project name: Interrupts.c
* Created: 10/31/2023 6:41:30 PM
* Author: Caiden Moreno
* Overview: This program uses interrupt service routines to execute a 500ms delay with
timer0 on one LED, display the ADC MSB values on 8 LEDS,
* and use an encoder to display the LEDS on and off one at a time.
*
* Hardware:
* Arduino ATmega2560 micro controller
* Inputs:
* LED outputs      = PORTA
* 500ms LED output = PINB.7
* ADC              = ADC0
* Encoder clk      = INT0
* Encoder DT       = PINC.0
*********************************************************************************/

#include <avr/io.h>
#include <avr/interrupt.h>
#include "Debugger.h"
#include "interrupt_timer0.h"
#define Mode_switcher (PINL & 0x01) //PINL.0

//function prototypes
void io_init(void);
void init_timer0(void);

//global variables
volatile uint16_t tick;
volatile uint16_t adc_read;
volatile uint8_t CW_flag;
volatile uint8_t CCW_flag;
volatile uint8_t volume;



int main(void)
{
       //call functions
       io_init();
       init_timer0();
       initDebug();
       init_ADC();
    init_encoder();
    sei();

    while (1)
    {

              //500ms timer
              if(tick>500){ //check if tick is larger than 500
                     PORTB ^=(1<<PB7);//toggle LED
                     tick=0; //reset tick to 0
              }
```

```
                //adc
                ADCSRA |= (1<<ADSC); //start conversion
                PORTA = adc_read; //reads the adc value and outputs to PORTA

                //encoder
                if(CW_flag==1){
                        volume |= (volume<<1); //shift the output by one
                        PORTA=volume; //output to LEDS
                        CW_flag=0; //reset the flag
                }

                if(CCW_flag==1){
                        volume= (volume>>1); ///shit the output by one
                        PORTA=volume; //output to LEDS
                        CCW_flag=0; //reset the flag
                }

                //check if volume is 0
                if(volume==0)
                {
                        volume=0x01; //set it to one
                }

        }
          }
}

void io_init(void){
        //outputs for LEDs
        DDRA = 0xFF;
        PORTA = 0x00;

    //500ms delay LED
        DDRB=(0xFF); //LED 13 set as output
        PORTB=(0x00); //turn off LED at initialization
}

/***********************************
 * interrupt_timer0.c
 *
 * Created: 10/31/2023 8:29:37 PM
 * Author : caiden Moreno
 ************************/

#include <avr/io.h>
#include <avr/interrupt.h>
#include "interrupt_timer0.h"

//500ms delay function
void init_timer0(void)
{

        TCCR0A = 0;
        TCCR0B = (1<<CS02) | (0<<CS01) | (0<<CS00);//prescaler 256
        TCNT0 = 200; // Set the initial value of Timer0 to 200
        TIMSK0=(1<<TOIE0); // Enable Timer0 overflow interrupt

}
```

```
//interrupt service routine for timer0 500ms LED
ISR(TIMER0_OVF_vect)
{
	TCNT0=200;  // Reset Timer0 to 200
	tick++;//tick increases by 1
}


//ADC
void init_ADC(){
	ADCSRA |= (1<<ADEN)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADPS2);//prescaler 128, enable adc
	ADMUX |= (1<<REFS0) | (1<<ADLAR);//reference voltage 5v left adjusted
	ADCSRB = 0x00;
	ADCSRA |= (1<<ADIE); //enable interrupt bit
	ADMUX = (ADMUX & (0xE0))|0;//selects channel 0
}

//interrupt service routine for ADC
ISR(ADC_vect) {

	adc_read = ADCH; //read potentiometer input

}

//encoder
void init_encoder(){

	EIMSK = (1<<INT0); //uses PD0 for external input pin
	EICRA = (1<<ISC01); // Configure external interrupts for falling edge on INT0
}

// Interrupt Service Routine for INT0
ISR(INT0_vect){
if(PINC & 0x01)//check if PINC is high
{
 CCW_flag=1; //if PINC is high set counter clockwise flag to 1
}
else
{
 CW_flag=1; //if PINC is low set clockwise flag to 1
}
}

/**********************************
 * interrupt_timer0.h
 *
 * Created: 10/31/2023 8:30:13 PM
 *  Author: Caiden Moreno
 **********************************/


#ifndef interrupt_timer0_
#define interrupt_timer0_

//include files
#include <avr/io.h>
#include <avr/interrupt.h>
```

```
//defines
#define CLK_PIN PD0
#define DT_PIN  PD1

//Define function Prototypes
void delay_ms(uint8_t ms);
void init_timer0(void);
void init_interrupt(void);
void tick_value(void);
void tick_set(uint16_t t);
void init_ADC(void);
void init_encoder(void);

//Global variables
volatile uint16_t tick;   //Global variable for tick value
volatile uint16_t adc_read; // Global variable to store the ADC value
volatile uint8_t CW_flag; //Global variable for clockwise flag
volatile uint8_t CCW_flag; //Global variable for counter clockwise flag
volatile uint8_t volume;  //Global variable for volume


#endif //interrupt_timer0_//
```