

```

/* -----
Universidad del Valle de Guatemala
IE2023: Programacion de Microcontroladores
PWMPotenciometro.c
Autor: Ian Anleu Rivera
Proyecto: Prelab 5
Hardware: ATMEGA328P
Creado: 14/04/2024
Ultima modificacion: 14/04/2024
-----
*/

//HEADER FILES
#define F_CPU 16000000 //Frecuencia es 16Mhz
#include <avr/io.h> //IO registros
#include <util/delay.h> // Lib para delays
#include <stdint.h> //Lib para enteros
#include <avr/interrupt.h> //Lib para interrupciones
#include "PWM1/PWM1.h" //PWM1

//VARIABLES GLOBALES
//Otras variables
uint8_t servo = 0;
uint8_t valoradc = 0;

//PROTOTIPOS DE FUNCIÓN
void setup(void);
void setupADC(void);

//INTERRUPCIONES

//ADC
ISR(ADC_vect){
    valoradc = ADCH; // Almacenar valor
    ADCSRA |= (1<<ADIF); // Apagar bandera
    ADCSRA |= (1<<ADSC); //Volver a iniciar
}

//PRINCIPAL
int main(void)
{
    cli();
    setup();
    setupADC();
    setupPWM1();
    sei();

    actualizar_servo(0); //Primer inicio de Servo
    ADCSRA |= (1<<ADSC); //Primer inicio de ADC
    uint8_t i = 0;

    while (1)
    {
        actualizar_servo(valoradc);
    } // Fin Main Loop
}

// OTRAS FUNCIONES
void setup(void){
    //ENTRADAS Y SALIDAS -----
    //Entradas

```

```

    // No hay digitales, solo ADC

    //Salidas
    //PB1 va al Servo 1 (OC1A)
    DDRB = 0;
    DDRB |= (1<<DDB1);
    //Otros servos
}

void setupADC(void){
    ADMUX = 0;
    ADMUX |= (1<<REFS0)|(1<<ADLAR)|(1<<MUX2)|(1<<MUX1)|(1<<MUX0); //Vcc ref | Just. Izq. |
Mux ADC7
    ADCSRA = 0;
    ADCSRA |= (1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); //Int. En | Presc. 128 (125kHz)
    ADCSRA |= (1<<ADEN); // Enable
}

```

IE2023: Programacion de Microcontroladores  
PWM1.c  
Autor: Ian Anleu Rivera  
Proyecto: Prelab 5  
Hardware: ATMEGA328P  
Creado: 14/04/2024  
Ultima modificacion: 14/04/2024

```
-----  
*/  
  
#include "PWM1.h"  
  
uint8_t listo_actualizar_pwm = 0;  
  
ISR(TIMER1_OVF_vect){  
    //Regresa a los 20ms  
    listo_actualizar_pwm = 0;  
}  
  
ISR(TIMER1_COMPA_vect){  
    //Probando si hace los ms necesarios  
}  
  
void setupPWM1(void){  
    //Interrupciones  
    TIMSK1 = 0;  
    TIMSK1 = (1<<OCIE1A) | (1<<TOIE1); //Compare A, Overflow  
  
    //Top en ICR1  
    ICR1H = (PWMTOP & 0xFF00) >> 8;  
    ICR1L = (PWMTOP & 0x00FF);  
  
    //Inicializar servo al minimo  
    OCR1AH = (SERVOMIN & 0xFF00) >> 8;  
    OCR1AL = (SERVOMIN & 0x00FF);  
  
    //Forma, modo, prescaler  
    TCCR1A = 0;  
    TCCR1B = 0;  
    TCCR1A |= (1<<COM1A1) | (1<<WGM11); //A1 No inverted | Fast con ICR1 Top  
    TCCR1B |= (1<<WGM12) | (1<<WGM13) | (1<<CS11); //Fast con ICR1 Top | Prescaler 8  
}  
  
static void actualizar_pwm(uint16_t i){  
    listo_actualizar_pwm = 1;  
    while (listo_actualizar_pwm != 0){  
        //El valor del pulso depende del leído en ADCH  
        OCR1AH = (i & 0xFF00) >> 8;  
        OCR1AL = (i & 0x00FF);  
    }  
}  
  
void actualizar_servo(uint16_t adchleido){  
    //Conversión de lectura de ADC a pulso en PWM  
    float razon = (float)adchleido/(float)256;  
    razon = (((float)SERVOMAX - (float)SERVOMIN)*razon) + (float)SERVOMIN;  
    uint16_t valor servo = (uint16_t)razon;  
    actualizar_pwm(valor servo);  
}
```

```

/* -----
Universidad del Valle de Guatemala
IE2023: Programacion de Microcontroladores
PWM1.h
Autor: Ian Anleu Rivera
Proyecto: Prelab 5
Hardware: ATMEGA328P
Creado: 14/04/2024
Ultima modificacion: 14/04/2024
-----
*/

#ifndef PWM1_H_
#define PWM1_H_

#include <avr/io.h> //IO registros
#include <stdint.h> //Lib para enteros
#include <avr/interrupt.h> //Lib para interrupciones

//Constantes
#define PWMTOP 39999
#define SERVOMAX 3999
#define SERVOMIN 1999

void setupPWM1(void);
void actualizar_servo(uint16_t adchleido);

#endif /* PWM1_H_ */

```