/\* -----------------------------------------------

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

}

/\* -----------------------------------------------

Universidad del Valle de Guatemala

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* valorservo = (*uint16\_t*)razon;

actualizar\_pwm(valorservo);

}

/\* -----------------------------------------------

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\_ \*/