```c
/*
 * Resolucao2aProvaSDM2020_1ControleCarga.c
 *
 * Created: 01/09/2020 22:46:02
 * Author : Ana Watanabe
 */



#define F_CPU 16000000UL //frequência do microcontrolador

#include <avr/io.h> //biblioteca para o microcontrolador
#include <avr/interrupt.h> //biblioteca pra interrupção

#define set_bit(Y,bit_x) (Y|=(1<<bit_x))
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x))
#define tst_bit(Y,bit_x) (Y&(1<<bit_x))
#define cpl_bit(Y,bit_x) (Y^=(1<<bit_x))

#define CHAVE       PB1    /*CHAVE é o substituto de PB1 */
#define BOTAO_EMERG PD7    /*BOTAO_EMER é o substituto de PD7 */
#define CARGA       PB0    /*CARGA é o substituto de PB0 */
#define TNORMAL     PB5    /*TNORMAL é o substituto de PB5 */
#define TELEVADA    PB6    /*TELEVADA é o substituto de PB6 */
#define TCRITICA    PB7    /*TCRITICA é o substituto de PB7 */


ISR(TIMER1_OVF_vect); // Protótipo da Interrupção TIMER1 do TC1
ISR(PCINT2_vect);     // Protótipo da Interrupção externa PCINT23

int flag_seg = 0; //variavel para contagem de tempo
int main(void)
{
	// Configuração de E/S digitais e inicialização
	DDRB  =  0b11100001; // Saídas: PB0, PB5, PB6 e PB7; Entrada: PB1
	PORTB =  0b00000010; // Pull Up: PB1 e saídas desligados: PB0, PB5, PB6 e PB7
	DDRD  =  0b01000000;  // Saída: PD6 e Entrada: PD7
	PORTD =  0b10000000; // Pull Up: PD7

	//  Configuração de ADC
	DIDR0 = 0x20; //desabilitando pino digital do ADC5
	ADMUX = 0x45; //habilitando ADC5
	ADCSRA = 0x87; //habilitando ADC com conversão simples
	// Cálculo do ADC
	// t_35 = 0,053*35 = 1,855V
	// ADC_35 = 1,855*1023/5 = 379
	// t_85 = 0,053*85 = 4,5V
	// ADC_85 = 4,5*1023/5 = 921

	// Configuração de Interrupção Externa
	UCSR0B = 0x00; // desativar Rx e Tx por usar PORTD
	PCMSK2 = 0x80; // PCINT23 (PD7)
	PCICR = 0x04;  // habilita PORTD (PCIE2)

	// Configuração do timer
	TCCR1A = 0; // timer oper.Normal OC1A e OC1B desconect. pg.217
	TCCR1B = 0; // TOP=0xffff, limpa registrador
	TCCR1B |= (1<<CS10)|(1 << CS12); // configura prescaler 1024
```

```
        TCNT1 = 0xC2F7; // valor para que estouro ocorra em 1 segundo
        TIMSK1 |= (1 << TOIE1); // habilita interrupção do TC1

        //Configuração do  PWM  (PD6 => OC0A)
        TCCR0A = 0b11000001; // PWM com fase corrigida, saida invertida OC0A
        TCCR0B = 0b00000100; // prescaler 256

        // Cálculo do prescaler do  PWM
        // f_pwm = f_osc / (2*prescaler*(1+TOP))
        // 122 = 16M / (2*prescaler*(255+1))
        // prescaler = 256

        OCR0A = 0; // duty cycle = 0
        sei(); // Liga a chave geral de interrupções.
        while(1)
        {
               if(!tst_bit(PINB,CHAVE))/*Se a chave ativa, liga tudo*/
                  {
                      if (flag_seg ==1)
                         {
                             set_bit(ADCSRA, ADSC); //ativando leitura
                             while(tst_bit(ADCSRA, ADSC) == 1) //esperando ler o ADC
                                 ;
                             flag_seg = 0;  //reseta o flag
                         }

                      if (ADC <= 379) //temperatura <= 35°, status: normal
                             {
                             OCR0A = 140; //pwm = 55% -> pwm = 255*0,55 = 140
                             set_bit(PORTB,CARGA);/*Liga a carga*/
                             PORTB|=(1<<TNORMAL);/*Liga o led  de temperatura normal*/
                             PORTB&=~((1<<TELEVADA)|(1<<TCRITICA));/*desliga os outros
leds*/
                             }
                      else if (ADC >= 921) //temperatura >= 85°, status: crítica
                             {
                             OCR0A = 255; //pwm = 100% -> pwm = 255*1 = 255
                             clr_bit(PORTB,CARGA);/*Desliga a carga*/
                             PORTB|=(1<<TCRITICA);/*Liga o led de crítica*/
                             PORTB&=~((1<<TNORMAL)|(1<<TELEVADA));/*desliga os outros
leds*/
                             }
                      else //temperatura entre 35° e 85°, status: elevada
                             {
                             OCR0A = 216; //pwm = 85% -> pwm = 255*0,85 = 216
                             set_bit(PORTB,CARGA);/*Liga a carga*/
                             PORTB|=(1<<TELEVADA);/*Liga o led de elevada*/
                             PORTB&=~((1<<TNORMAL)|(1<<TCRITICA));/*desliga os outros
leds*/
                             }
                  }   // fim do if da CHAVE
                 else    // se a chave desativada, desliga tudo
                     {
                      OCR0A = 0; //desliga o ventilador */
                      clr_bit(PORTB,CARGA);/*Desliga a carga*/
                      PORTB&=~((1<<TNORMAL)|(1<<TELEVADA)|(1<<TCRITICA));/*Desliga
todos os leds*/
                      }
```

```
	}    // fim do while
}
//Rotina de tratamento da interrupção externa
ISR(PCINT2_vect)
{
	unsigned char portb;
	portb = PORTB;
	while(tst_bit(PIND, BOTAO_EMERG) == 0) //se o BOTAO_EMERG foi acionado
		{
		OCR0A = 255; //pwm = 100% -> pwm = 255*1 = 255
		clr_bit(PORTB,CARGA);/*Desliga a carga*/
		PORTB|=(1<<TCRITICA);/*Liga o led de crítica*/
		PORTB&=~((1<<TNORMAL)|(1<<TELEVADA));/*desliga os outros leds*/
	    }
	PORTB = portb;  //salva o valor da porta B
}

//Rotina de tratamento da interrupção do timer
ISR(TIMER1_OVF_vect) //configurando o timer
{
		TCNT1 = 0xC2F7; //para resetar
		flag_seg = 1;
}
```