

```

/*
 * usina_C.c
 *
 * Created: 02/10/2019 15:39:19
 * Author : Ana Watanabe
 */

#define F_CPU 16000000UL /*define a frequência do microcontrolador 16MHz (necessário
para usar as rotinas de atraso)*/

#include <avr/io.h> //definições do componente especificado

//#define LED1 PB5 //LED1 é o substituto de PB5 na programação - teste

#define GERA1 PB1 //GERA1 é o substituto de PB1 na programação
#define GERA2 PB2 //GERA2 é o substituto de PB2 na programação
#define GERA3 PB3 //GERA3 é o substituto de PB3 na programação

#define COMP1 PD6 //COMP1 é o substituto de PD6 na programação
#define COMP2 PD7 //COMP2 é o substituto de PD7 na programação

#define set_bit(Y,bit_x) (Y|=(1<<bit_x)) /*ativa o bit x da variável Y (coloca em
1)*/
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x)) /*limpa o bit x da variável Y (coloca em
0)*/
#define tst_bit(Y,bit_x) (Y&(1<<bit_x)) /*testa o bit x da variável Y (retorna 0 ou
1)*/

main(void)
{

    DDRB = 0x07; //configura os pinos do PORTB1, PORTB2 e PORTB3 como saídas
    DDRD = 0xC0; //configura os pinos do PORTD6 e PORTD7 como saídas

    //Desliga os geradores e fecha as comportas

    clr_bit(PORTB,GERA1); //desliga gerador 1
    clr_bit(PORTB,GERA2); //desliga gerador 2
    clr_bit(PORTB,GERA3); //desliga gerador 3

    clr_bit(PORTD,COMP1); //fecha comporta 1
    clr_bit(PORTD,COMP2); //fecha comporta 2

    // Configuração do ADC
    // DIDR0 => ADC0D =1 e ADC5D = 1 entrada analógica no PC0 e PC5
    DIDR0 = 0x21; //
    // ADMUX => Tensão AVCC, alinhado a direita
    ADMUX = 0x40; //
    // ADCSRA => ADC habilitado, prescaler = 128
    ADCSRA = 0x87;

    while(1) //laço infinito
    {

        //seleciona o canal para nivel ADC5

```

```

ADMUX = 0x45;
set_bit(ADCSRA, ADSC);
while (tst_bit(ADCSRA, ADSC))
;
if (ADC <= 184) // menor ou igual que 18m
{ // desliga os geradores
  clr_bit(PORTB,GERA1); //desliga gerador 1
  clr_bit(PORTB,GERA2); //desliga gerador 2
  clr_bit(PORTB,GERA3); //desliga gerador 3
  // fecha as comportas
  clr_bit(PORTD,COMP1); //fecha comporta 1
  clr_bit(PORTD,COMP2); //fecha comporta 2
}
else
{
  if (ADC <= 512 ) // maior que 18m e menor igual a 50m
  {
    set_bit(PORTD,COMP1); //abre comporta 1
    clr_bit(PORTD,COMP2); //fecha comporta 2
  }
  else // maior que 50m
  {
    set_bit(PORTD,COMP1); //abre comporta 1
    set_bit(PORTD,COMP2); //abre comporta 2
  }
  //seleciona o canal para consumo ADC0

ADMUX = 0x40;
set_bit(ADCSRA, ADSC);
while (tst_bit(ADCSRA, ADSC))
;
if (ADC <= 306) // menor ou igual a 50MW
{
  set_bit(PORTB,GERA1); //liga gerador 1
  clr_bit(PORTB,GERA2); //desliga gerador 2
  clr_bit(PORTB,GERA3); //desliga gerador 3
}
if ((ADC > 306) && (ADC <= 613)) // maior que 50MW e menor que igual 100MW
{
  set_bit(PORTB,GERA1); //liga gerador 1
  set_bit(PORTB,GERA2); //liga gerador 2
  clr_bit(PORTB,GERA3); //desliga gerador 3
}
if (ADC > 613) // maior que 100MW
{
  set_bit(PORTB,GERA1); //liga gerador 1
  set_bit(PORTB,GERA2); //liga gerador 2
  set_bit(PORTB,GERA3); //liga gerador 3
}
} //fecha else
} // fecha while
}

```