

```

/*****
* maquina de lavar_C_ADC_INT_PWM.c
*
* Created: 06/11/2019 16:45:31
* Author : Ana Watanabe
*****/

#define F_CPU 16000000UL
#include <avr/io.h> //definições do componente especificado
#include <avr/interrupt.h>

#define set_bit(Y,bit_x) (Y|=(1<<bit_x)) /*ativa o bit */
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x)) /*limpa o bit */
#define tst_bit(Y,bit_x) (Y&(1<<bit_x)) /*testa o bit */

// Tabela de vetores pag. 158
ISR(INT1_vect); // Protótipo da Interrupção externa INT1.
ISR(TIMER1_OVF_vect); // Protótipo da Interrupção TIMER1. (16 bits)

char tempo_lavacao = 0; // tempo de lavação media e pesada
char flag_tempo = 0; // permite contar tempo
char minuto = 0;
char segundo = 0;

char status = 0; // estado das saidas da maquina;
char status_motor = 0; // estado do estado do motor (duty cycle)
int main(void)
{
    // configuração de E/S digitais
    DDRB = 0x00; // pinos PB3 e PB4: entradas
    DDRC = 0x00; // pino PC1: entrada
    DDRD = 0x27; // pinos: PD0, PD1, PD2 e PD5: saídas e PD3: entrada
    PORTB = 0x18; // pull up: PB3 E PB4
    PORTC = 0x02; // pull up: PC1
    set_bit(PORTD, PD3); // pull up da entrada da interrupção externa: PD3

    //Desliga as bombas de água

    clr_bit(PORTD, PD0); //DESLIGA A BOMBA DE ENTRADA ÁGUA
    clr_bit(PORTD, PD1); //DESLIGA A BOMBA DE SAÍDA DE ÁGUA

    // Configuração do ADC
    DIDR0 = 0x0C; // entrada analógica no PC2 e PC3
    ADCSRA = 0x87; // ADC habilitado, prescaler = 128

    // configuração da interrupção INT1 da tampa da máquina
    UCSRB = 0x00; // desabilitar Rx e Tx usar PORTD pg.354
    EICRA = 0x00; // nivel baixo em INT1 pg. 163
    EIMSK = 0x02; // Ativa INT1. pg. 164

    // configuração da interrupção de timer no TC1(16 bits)
    TCCR1A = 0; // timer para oper.normal OC1A e OC1B desconect.
    TCCR1B = 0; //limpa registrador
    TCCR1B |= (1<<CS10)|(1<<CS12); // configura prescaler 1024
    TCNT1 = 0xC2F7; //valor para que estouro ocorra em 1 segundo
    // 65536-(16MHz/1024/1Hz) = 65536 - 15.625 = 49911(0xC2F7)
    TIMSK1 |= (1<<TOIE1);

```

```

// configuração do PWM no PD5 => TC0 => OCR0B   pg.32
TCCR0A = 0b00110001; //PWM com fase corrigida, saída OC0B não invertida,
modo 1 e TOP = 0xff   pg.196
TCCR0B = 0x02; //PWM fase corrigida, modo 1 e prescaler = 8
                // fPWM = 4000Hz

// desliga motor
OCR0B = 0;    // duty cycle = 0

sei(); // Liga a chave geral de interrupções.

while(1) //laço infinito
{
    while (tst_bit(PORTB, PB3))    //aguarda o botão iniciar
        ;
    set_bit(PORTD, PD0); //LIGA A BOMBA DE ENTRADA DE ÁGUA
    do{
        // verifica nivel de água
        ADMUX = 0x43; // Tensão AVCC, alinhado a direita, canal 3
        set_bit(ADCSRA, ADSC);
        while (tst_bit(ADCSRA, ADSC) ==1) // aguarda leitura
            ;
    } while (ADC < 818); //aguarda completar 10 litros
    clr_bit(PORTD, PD0); //DESLIGA A BOMBA DE ENTRADA ÁGUA
    if(!tst_bit(PINB, PB4)) // Se foi optado com aquecimento
    {
        set_bit(PORTD, PD2); //LIGA AQUECIMENTO
        do{
            // verifica aquecimento
            ADMUX = 0x42; //Tensão AVCC, alinhado a direita, canal 2
            set_bit(ADCSRA, ADSC);
            while (tst_bit(ADCSRA, ADSC)) // aguarda leitura
                ;
        } while (ADC < 460); //aguarda completar 45 graus
        clr_bit(PORTD, PD2); //DESLIGA AQUECIMENTO
    }
    if(!tst_bit(PINC, PC1)) // se lavagem pesada
    {
        tempo_lavacao = 15;
        OCR0B = 229; // 90%
    }
    else // se lavagem leve
    {
        tempo_lavacao = 10;
        OCR0B = 102; // 40%
    }
    flag_tempo = 1;
    TCNT1 = 0xC2F7; // reseta o contador para contar 1s
    while (minuto < tempo_lavacao)
        ;
    OCR0B = 0 // desliga motor
    flag_tempo = 0; // pára de contar tempo
    set_bit(PORTD, PD1); // LIGA BOMBA DE SAIDA
    do
    {
        set_bit(ADCSRA, ADSC); //leitura do ADC

```

```

        while (tst_bit(ADCSRA, ADSC))
            ;
    } while (ADC); //aguarda ACABAR A ÁGUA
    clr_bit(PORTD, PD1); //DESLIGA A BOMBA DE SAÍDA DE ÁGUA
}
}

//*****
//  Rotina de tratamento de Interrupção
//*****

ISR(INT1_vect){
status = PORTD; // salva conteúdo das saídas na PORTD
status_motor = OCR0B;
OCR0B = 0; //desliga motor
PORTD = 0x00; // desliga tudo
// verifica se a tampa foi fechada
while(!tst_bit(PIND, PD3))
    ;
PORTD = status; // retorna condição das saídas
OCR0B = status_motor; // ...inclusive do motor
}

//*****
//  Rotina de tratamento de Timer (1s)- clock interno
//*****

ISR(TIMER1_OVF_vect){
// verifica flag_tempo, se igual a zero, sai da rotina
if(flag_tempo !=0)
{
    TCNT1 = 0xC2F7; // reseta o contador para contar 1s
    segundo++;
    if (segundo == 60)
    {
        minuto ++;
        segundo = 0;
    }
}
}
}

```