

```

/*****
 * maquina de lavar_C_ADC.c
 *
 * Created: 22/10/2019 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 INT1.
ISR(TIMER1_OVF_vect); // Protótipo da Interrupção TIMER1.

char tempo_lavacao = 0; // tempo de lavação media e pesada
char flag_tempo = 0; // flag para contagem de tempo.
// tempo =0 (não conta) e tempo =1 (conta tempo)

char minuto = 0;

int main(void)
{
    // configuração de E/S digitais
    DDRB = 0x00; // pino PB3: entrada
    DDRC = 0x00; // pino PC1: entrada
    DDRD = 0x93; // pinos: PD0, PD1 e PD4 e PD7: saídas, PD3: entrada
    PORTB = 0x08; // pull up: PB3
    PORTC = 0x02; // pull up: PC1
    set_bit(PORTD, PD3); // pull up: PD3

    //Desliga as bombas de água e o motor

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

    clr_bit(PORTD, PD4); // desliga motor
    clr_bit(PORTD, PD7);

    // Configuração do ADC
    DIDR0 = 0x08; // ADC3D =1 entrada analógica no PC3
    ADMUX = 0x43; // Tensão AVCC, alinhado a direita, canal 3
    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      pg. 217
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);
```

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

```
while(1) //laço infinito
{
    while (tst_bit(PORTB, PB3))      //aguarda o botão iniciar
    ;
    do{
        set_bit(PORTD, PD1); //LIGA A BOMBA DE ENTRADA DE ÁGUA
        // verifica nivel de água
        set_bit(ADCSRA, ADSC);
        while (tst_bit(ADCSRA, ADSC)) // aguarda leitura
        ;
        } while (ADC < 818); //aguarda completar 10 litros
        clr_bit(PORTC, PC1); //DESLIGA A BOMBA DE ENTRADA ÁGUA
        if(tst_bit(PIND, PD0))      // se lavagem pesada
        {
            tempo_lavacao = 15;
            set_bit(PORTD, PD4);
            set_bit(PORTD, PD7);
        }
        else                          // se lavagem media
        {
            tempo_lavacao = 10;
            clr_bit(PORTD, PD4);
            set_bit(PORTD, PD7);
        }
        flag_tempo = 1;      // começa a contar tempo
        TCNT1 = 0xC2F7;      // reseta o contador para contar 1s
        while (minuto < tempo_lavacao)
        ;
        clr_bit(PORTD, PD4); // desliga motor
        clr_bit(PORTD, PD7);

        flag_tempo = 0;      // pára de contar tempo
        set_bit(PORTD, PD1); // LIGA BOMBA DE SAÍDA
        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
// Obs: Quando entra numa rotina de interrupção nenhuma outra interrupção
// vai ocorrer porque o microcontrolador dá um cli(), inclusive
// o de timer.
//*****

ISR(INT1_vect){
char status;    // estado das saidas da maquina;
char count;

status = PORTD; // salva conteúdo das saidas na PORTD
count = TCNT1;  // salva o valor do contador
PORTD = 0x00;   // desliga tudo
// verifica se a tampa foi fechada
while(!tst_bit(PIND,PD3))
    ;
PORTD = status; // recupera estado das saidas na PORTD
TCNT1 = count;  // recupera o valor do contador
}

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

ISR(TIMER1_OVF_vect){
char segundo;

// 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;
    }
}
}
}

```