

```

F:\SYLVIALLY_MICROCONTROLADORES\TRABALHOS FINALIZADOS\sylvielly_projeto_final\sylvielly_co
/*MICROCONTROLADORES - ENGENHARIA DE CONTROLE E AUTOMACAO*/
/*PROJETO FINAL - CONTROLE DE AQUECIMENTO DA ÁGUA POR PWM E MONITORACAO DE TEM
//ALUNA: SYLVIALLY S. SOUSA
//MATRICULA: 20162045070410                                PROF.: FÁBIO TIMBÓ

#include<18f4550.h>                                //PIC18F4550
#fuses hs, nowdt, CPUDIV1,brownout,nolvp                                //habilita
                                                                    //hs = hig
                                                                    //CPUDIV1

#device adc = 10                                //resolucao conversor A/

#use delay(clock=8000000)                                //frequencia de operacao

#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)                                //habilitacao comunicaca
                                                                    //baud = banda
                                                                    //xmit = pino de transmi
                                                                    //rcv = pino de recepcao

//trocar AN0 por AN3
//inicio configuracao LCD
//#use standart_io(B)                                //comentar linha caso biblioteca do lcd seja d

//#define use_portD_lcd TRUE                                //comentar linha caso biblioteca do lcd seja
#define LCD_TYPE 2                                //tipo lcd
#include "LCD8B.c"                                //hiperlink programa LCD
#include "12Keys.c"                                //hiperlink programa teclado matricial
//#include<LCD.c>
//fim configuracao LCD

//declaracao de variaveis
long int LM_35;                                //variavel sensor de temperatura
long int AD3;                                //variavel porta analogica (variavel longa de
int vetor[2],valor_config=0;
int auxiliar = 0, n=0, n2=0;
int conta=0;
#int_timer0
void tempo(){
set_timer0(131+get_timer0());
n++;
n2++;
}

//inicio programa
void main(){
setup_timer_0(RTCC_DIV_64|RTCC_INTERNAL|RTCC_8_BIT);
set_timer0(131);
enable_interrupts(GLOBAL);
enable_interrupts(INT_TIMER0);

                                port_b_pullups(True);                                //habilitacao de pullups (para o
                                lcd_ini();                                //inicializacao LCD para a biblio
                                setup_timer_2(T2_DIV_BY_4,249,1);

setup_adc_ports(AN0_TO_AN3);                                //pino AN3 sera o pino ond
setup_adc(adc_clock_internal);
set_adc_channel(3);                                //seta canal analógico 3

                                tecla_pres=0;                                //verifica condicao de tecla pres
                                int valor=100;                                //valor para ajuste duty cycle PW

```

```

F:\SYLVIELLY_MICROCONTROLADORES\TRABALHOS FINALIZADOS\sylvIELly_projeto_final\sylvIELly_co
    lcd_pos_xy(1,1); //seleciona posicao no LCD
    printf(lcd_escreve, "\fTEMP. DESEJADA"); //solicita ao usua
    printf("\r\nDIGITE TEMPERATURA DESEJADA NO TECLADO MATRICIAL\r\n
//inicio while
while(TRUE){
    //para o projeto, usuario digita valor de temperatura no teclad
    //inicio teclado matricial
    //analise do teclado matricial de forma que ele capture o valor
    if(conta<=1){ //testa t
        tecla = trata_tecclas();

        if (tecla_pres)
        {
            lcd_pos_xy((conta+1),2); //coloca
            printf(lcd_escreve,"%c", tecla);
            printf("%c",tecla); //captura

            vetor[conta] = tecla-48; //convert

            conta++; //increme
        }
        tecla_pres = 0; //tecla pressionada volta para zero
    }
    //fim teclado matricial

    //inicio PWM
    else if(conta>=2){ //quando o segu

        valor_config = vetor[0]*10+vetor[1]; //valor configu
        LM_35 = read_adc(); //LM35 lê valor
        delay_ms(10);
        AD3 = LM_35*0.488758; //converte valo

        if((valor_config <= AD3)) //compara se va
        {
            set_pwm2_duty(0); //seta PWM para
            setup_ccp2(CCP_PWM); //envia para o
            if(n2==50){ //tempo para es
                lcd_pos_xy(1,1);
                printf(lcd_escreve, "\fAJUSTE OK!!!"); //tem
                lcd_pos_xy(1,2);
                printf(lcd_escreve, "TEMP CONF = %d", valor_config
                n2=0;
            }

            if(n==2500) //tempo para es
            {
                printf("\r\nAJUSTE OK!!!");
                printf("\r\nTEMPERATURA CONFIGURADA = %d", val
                n=0;
                delay_ms(1000);
            }

        }
        else{ //mostra temperatura em tempo real no LCD e se
            auxiliar = AD3;
            if(n2==50){
                lcd_pos_xy(1,1);
                printf(lcd_escreve, "\fTEMP REAL = %lu", AD3);
                lcd_pos_xy(1,2);
                printf(lcd_escreve, "TEMP CONF = %d", valor_config
                n2=0;}

```

```

F:\SYLVIELLY_MICROCONTROLADORES\TRABALHOS_FINALIZADOS\sylvielly_projeto_final\sylvielly_co
    if(n==2500)                                     //tempo para es
    {

        printf("\r\nTEMPERATURA REAL = %lu",AD3);
        printf("\r\nTEMPERATURA CONFIGURADA = %d",val
        n=0;
    }

    set_pwm2_duty(valor);                           //seta PWM pa
    setup_ccp2(CCP_PWM);                             //envia para
}

}

} //fim while
} //fim programa

```