

```

/* ***** */
/*
/*
/* Nome do arquivo:      main.c
/*
/*
/* Descricao:           Projeto final do controlador de temperatura com
/*                        interface
/*
/*
/* Autores:             Gustavo Lino e Giacomo A. Dollevedo
/*
/* Criado em:           27/07/2020
/*
/* Ultima revisao em:    28/07/2020
/* ***** */

/* Incluindo bibliotecas */
#include "board.h"
#include "util.h"
#include "lut_adc_3v3.h"
#include "lptmr.h"
#include "lcdTemp.h"
#include "ledSwi.h"
#include "pid.h"
#include "display7Temp.h"
#include "uart.h"
#include "sensTemp.h"
#include "variaveis_globais.h"
#include "stdio.h"
#include "aquecedorECooler.h"

/******
/*****      DECLARANDO CONSTANTES      *****/

/*Temperatura*/
#define TEMP_DEFAULT 30

/*Tick base do timer (em microsegundos [us])*/
#define TICK_4MS    4000

/*Estados principais do sistema*/
#define CONFIG      0
#define CONTROLE    1

/*Subestados CONFIG*/
#define UNIDADE      0
#define DEZENA       1

/*Subestados UNIDADE e DEZENA*/
#define STATE_0      0
#define STATE_N      1
#define STATE_9      2

/*Constante de ganho do controlador*/
#define FKP          0.001f
#define FKI          0.002f
#define FKD          0.003f

/******      FIM DAS CONSTANTES      *****/
/***** */

/*Variaveis para operacao do sistema*/
unsigned char ucEstado      = 0;
unsigned char ucSubestado1  = 0;
unsigned char ucSubestado2  = 0;

/*Variaveis para manter controle do tempo durante execucao*/
unsigned char ucContador1   = 0;
unsigned char ucContador2   = 0;
unsigned char ucContadorCtrl = 0;

```

```
unsigned char ucSegundos    = 0;
unsigned char ucMinutos     = 0;
unsigned char ucIdleTime    = 0;
```

```
/*Variaveis relacionadas aos displays*/
```

```
unsigned char ucD7Flag      = 0;
unsigned char ucLCDFrame    = 1;
unsigned char ucDisableD7   = 0;
```

```
/*Variavel para disparar o controle*/
```

```
unsigned char ucAttCtrl     = 0;
```

```
/* ***** */
/* Nome do metodo:      timerAtt */
/* Descricao:           Callback da interrupcao gerada pelo timer Itpmr0 */
/*                       para controlar os displays e demais elementos */
/*                       sensíveis ao tempo */
/*
/* Parametros de entrada:  n/a */
/*
/* Parametros de saida:   n/a */
/* ***** */
```

```
void timerAtt(){
```

```
    ucContador1++;
    ucContador2++;
    ucD7Flag = 1;
}
```

```
/* ***** */
/* Nome do metodo:      checkTime */
/* Descricao:           Atualiza os contadores de tempo de execucao do */
/*                       programa */
/*
/* Parametros de entrada:  n/a */
/*
/* Parametros de saida:   n/a */
/* ***** */
```

```
void checkTime(){
```

```
    /*+1 a cada 100ms*/
    if(25 <= ucContador2){
        ucContador2 = 0;
        ucContadorCtrl++;
    }
```

```
    /*+1 a cada segundo*/
    if(250 <= ucContador1){
        ucContador1 = 0;
        ucSegundos++;
    }
```

```
    /*+1 a cada minuto*/
    if(60 <= ucSegundos){
        ucSegundos = 0;
        ucMinutos++;

        if(CONFIG == ucEstado){
            ucIdleTime++;
        }
    }
}
```

```
/* ***** */
/* Nome do metodo:      boardInit */
```

```

/* Descricao:      Inicializa os perifericos necessarios para o */
/*                projeto */
/*                */
/* Parametros de entrada:  n/a */
/*                */
/* Parametros de saida:    n/a */
/*                */
/* ***** */
void boardInit()
{

    /*Inicializa o primeiro LED e os 3 ultimos botoes*/
    /*Botao 2 => "-"; Botao 3 => "+"; Botao 4 => "OK/Reset"*/
    ledSwi_init(1, 0, 0, 0);

    /*Inicializa o ADC para leitura do sensor de temperatura*/
    sensTemp_init();

    /*Inicializa o Display de 7 segmentos*/
    /*O D7S sera usado para exibir a temperatura atual*/
    display7Temp_init();

    /*Inicializa o display LCD*/
    lcdTemp_init();

    /*Inicializa um timer com tick de 4ms para atualizacao dos displays*/
    /*e controle de tempo interno do sistema*/
    tc_installLptmr0(TICK_4MS, timerAtt);

    /*Inicializa o controlador PID para atuar sobre o aquecedor*/
    pid_init();
    pid_setKi(fKi);
    pid_setKp(fKp);
    pid_setKd(fKp);

    /*Inicializa a comunica~o UART*/
    UART0_init();

    /*Inicializa o PWM, o ventilador e o aquecedor*/
    PWM_init();
    coolerfan_init();
    heater_init();
}

/* ***** */
/* Nome do metodo:      main */
/* Descricao:      Executa o loop principal do programa */
/*                */
/*                */
/* Parametros de entrada:  n/a */
/*                */
/* Parametros de saida:    n/a */
/*                */
/* ***** */
int main(void){

    /*Inicializa todos os perifericos do Kit*/
    boardInit();

    /*Definicao dos estados iniciais*/
    ucEstado      = CONFIG;
    ucSubestado1   = DEZENA;
    ucSubestado2   = STATE_0;

    /*Loop infinito de operacao do sistema*/
    while(1){

```

```

/*Faz a contagem de tempo de acordo com a interrupcao programada*/
checkTime();

/*Atualiza o LCD com o frame correto*/
/*1 -> CONFIG; 2 -> CONTROLE*/
if(0 != ucLCDFrame){
    ucDisableD7 = 1;
    showLCDdisp(ucLCDFrame);
    attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
    ucLCDFrame = 0;
    ucDisableD7 = 0;
}

/*Atualiza o D7S quando h  interrupcao && quando o LCD nao esta sendo operado*/
if(0 != ucD7Flag && 0 == ucDisableD7){
    attDisp7Temp();
}

/*Se o sistema fica inoperado por 2 minutos (no estado CONFIG), uma temperatura padrao eh*/
/*setada, e o sistema passa para o estado de CONTROLE*/
if(2 == ucIdleTime){
    ucIdleTime = 0;
    ucTempAlvo = TEMP_DEFAULT;
    ucDezTempAlvo = ucTempAlvo/10;
    ucUnTempAlvo = ucTempAlvo%10;
    ucEstado = CONTROLE;
    ucSubestado1 = DEZENA;
    ucSubestado2 = STATE_0;
}

if(1 <= ucContadorCtrl){
    readTemp();
    ucContadorCtrl = 0;
    ucAttCtrl++;
}

/*INICIA A MAQUINA DE ESTADOS DO SISTEMA*/
switch (ucEstado){

/*ESTADO DE CONFIGURACAO DA TEMPERATURA ALVO*/
case CONFIG:
    /*LED 1 indica que o estado atual eh de configuracao*/
    turnOnLED(1);
    ucLCDFrame = 0;

    if(0 == ucDezTempAlvo && DEZENA == ucSubestado1)
        ucSubestado2 = STATE_0;

    else if(9 == ucDezTempAlvo && DEZENA == ucSubestado1)
        ucSubestado2 = STATE_9;

    else
        if(DEZENA == ucSubestado1)
            ucSubestado2 = STATE_N;

    switch(ucSubestado1){

/*ESTADO PARA CONFIGURACAO DO DIGITO DA DEZENA*/
case DEZENA:
    switch(ucSubestado2){
        /*ESTADO PARA DIGITO == 0*/
        case STATE_0:
            /*Caso "OK" pressionado*/
            if(1 == readSwitch(4)){

```

```

    util_genDelay100ms();
    ucSubestado1 = UNIDADE;

    if(0 == ucUnTempAlvo)
        ucSubestado2 = STATE_0;

    else if(9 == ucUnTempAlvo)
        ucSubestado2 = STATE_9;

    else
        ucSubestado2 = STATE_N;

    break;
}

/*Caso "+" pressionado*/
else if(1 == readSwitch(3)){
    util_genDelay100ms();
    ucDezTempAlvo++;
    ucDisableD7 = 1;
    attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
    ucDisableD7 = 0;

    ucSubestado2 = STATE_N;
    break;
}

/*Caso "-" pressionado*/
else if(1 == readSwitch(2)){
    util_genDelay100ms();
    ucDisableD7 = 1;
    attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
    ucDisableD7 = 0;
    ucSubestado2 = STATE_0;
    break;
}

/*ESTADO PARA 0 < DIGITO < 9*/
case STATE_N:
    /*Caso "OK" pressionado*/
    if(1 == readSwitch(4)){
        util_genDelay100ms();
        ucSubestado1 = UNIDADE;

        if(0 == ucUnTempAlvo)
            ucSubestado2 = STATE_0;

        else if(9 == ucUnTempAlvo)
            ucSubestado2 = STATE_9;

        else
            ucSubestado2 = STATE_N;
        break;
    }

    /*Caso "+" pressionado*/
    else if(1 == readSwitch(3)){
        util_genDelay100ms();
        if(8 == ucDezTempAlvo){
            ucDezTempAlvo++;

            ucDisableD7 = 1;
            attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
            ucDisableD7 = 0;
            ucSubestado2 = STATE_9;
        }
    }
}

```

```

else{
    ucDezTempAlvo++;

    ucDisableD7 = 1;
    attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
    ucDisableD7 = 0;
    ucSubestado2 = STATE_N;
}
break;
}

/*Caso "-" pressionado*/
else if(1 == readSwitch(2)){
    util_genDelay100ms();

    if(1 == ucUnTempAlvo){
        ucDezTempAlvo--;
        ucDisableD7 = 1;
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_0;
    }

    else{
        ucDezTempAlvo--;
        ucDisableD7 = 1;
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_N;
    }
    break;
}

/*ESTADO PARA DIGITO == 9*/
case STATE_9:
    /*Caso "OK" pressionado*/
    if(1 == readSwitch(4)){
        util_genDelay100ms();
        ucSubestado1 = UNIDADE;

        if(0 == ucUnTempAlvo)
            ucSubestado2 = STATE_0;

        else if(9 == ucUnTempAlvo)
            ucSubestado2 = STATE_9;

        else
            ucSubestado2 = STATE_N;

        break;
    }

    /*Caso "+" pressionado*/
    else if(1 == readSwitch(3)){
        /*Nada acontece*/
        util_genDelay100ms();
        ucDisableD7 = 1;
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_9;
        break;
    }

    /*Caso "-" pressionado*/
    else if(1 == readSwitch(2)){

```

```

        util_genDelay100ms();
        ucDezTempAlvo--;
        ucDisableD7 = 1;
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_N;
        break;
    }
}

```

```

break; /*Break do Subestado1*/

```

```

/*ESTADO PARA CONFIGURACAO DO DIGITO DA UNIDADE*/

```

```

case UNIDADE:
    switch(ucSubestado2){
        /*ESTADO PARA DIGITO == 0*/
        case STATE_0:
            /*Caso "OK" pressionado*/
            if(1 == readSwitch(4)){
                util_genDelay100ms();
                ucSubestado1 = DEZENA;
                ucSubestado2 = STATE_0;
                ucEstado = CONTROLE;
                ucLCDFrame = 2;
                ucTempAlvo = ((10*ucDezTempAlvo)+ucUnTempAlvo);
                break;
            }

            /*Caso "+" pressionado*/
            else if(1 == readSwitch(3)){
                util_genDelay100ms();
                ucUnTempAlvo++;
                ucDisableD7 = 1;
                attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
                ucDisableD7 = 0;
                ucSubestado2 = STATE_N;
                break;
            }

            /*Caso "-" pressionado*/
            else if(1 == readSwitch(2)){
                util_genDelay100ms();
                ucDisableD7 = 1;
                attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
                ucDisableD7 = 0;
                ucSubestado2 = STATE_0;
                break;
            }
        }

        /*ESTADO PARA 0 < DIGITO < 9*/
        case STATE_N:
            /*Caso "OK" pressionado*/
            if(1 == readSwitch(4)){
                util_genDelay100ms();
                ucSubestado1 = DEZENA;
                ucSubestado2 = STATE_0;
                ucEstado = CONTROLE;
                ucLCDFrame = 2;

                if(9 == ucDezTempAlvo)
                    ucUnTempAlvo = 0;

                ucTempAlvo = ((10*ucDezTempAlvo)+ucUnTempAlvo);

                break;
            }
        }
    }
}

```

```
}
```

```
/*Caso "+" pressionado*/
```

```
else if(1 == readSwitch(3)){  
    util_genDelay100ms();  
    if(8 == ucUnTempAlvo){  
        ucUnTempAlvo++;  
        ucDisableD7 = 1;  
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);  
        ucDisableD7 = 0;  
        ucSubestado2 = STATE_9;  
    }  
}
```

```
    else{  
        ucUnTempAlvo++;  
        ucDisableD7 = 1;  
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);  
        ucDisableD7 = 0;  
        ucSubestado2 = STATE_N;  
    }  
    break;  
}
```

```
/*Caso "-" pressionado*/
```

```
else if(1 == readSwitch(2)){  
    util_genDelay100ms();  
  
    if(1 == ucUnTempAlvo){  
        ucUnTempAlvo--;  
        ucDisableD7 = 1;  
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);  
        ucDisableD7 = 0;  
        ucSubestado2 = STATE_0;  
    }  
}
```

```
    else{  
        ucUnTempAlvo--;  
        ucDisableD7 = 1;  
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);  
        ucDisableD7 = 0;  
        ucSubestado2 = STATE_N;  
    }  
    break;  
}
```

```
/*ESTADO PARA DIGITO == 9*/
```

```
case STATE_9:
```

```
/*Caso "OK" pressionado*/
```

```
if(1 == readSwitch(4)){  
    util_genDelay100ms();  
    ucSubestado1 = DEZENA;  
    ucSubestado2 = STATE_0;  
    ucEstado = CONTROLE;  
    ucLCDFrame = 2;  
  
    if(9 == ucDezTempAlvo)  
        ucUnTempAlvo = 0;  
  
    ucTempAlvo = ((10*ucDezTempAlvo)+ucUnTempAlvo);  
    break;  
}
```

```
/*Caso "+" pressionado*/
```

```
else if(1 == readSwitch(3)){  
    util_genDelay100ms();  
    ucDisableD7 = 1;
```



```

        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_9;
        break;
    }

    /*Caso "-" pressionado*/
    else if(1 == readSwitch(2)){
        util_genDelay100ms();
        ucUnTempAlvo--;
        ucDisableD7 = 1;
        attTempAlvo(ucDezTempAlvo, ucUnTempAlvo);
        ucDisableD7 = 0;
        ucSubestado2 = STATE_N;
        break;
    }
}

break; /*Break do Subestado1*/

}

break; /*Break switch Estado*/

/*ESTADO DE CONTROLE DE TEMPERATURA*/
case CONTROLE:
    /*Habilita as interrupcoes da porta Serial*/
    UART0_enableIRQ();

    /*Apaga o led que indica o estado de configuracao*/
    turnOffLED(1);

    /*NOVA ATUALIZAÃÃO DE CONTROLE NECESSÃRIA*/
    if(0 != ucAttCtrl) {

        fDutyCycle_Heater = pidUpdateData(ucTempAtual, ucTempAlvo, fDutyCycle_Heater);
        heater_PWMDuty(fDutyCycle_Heater);
    }

    /*INDICACAO VISUAL SE A TEMPERATURA ESTA ACIMA DO SETPOINT*/
    if(ucTempAtual > ucTempAlvo){
        turnOffLED(2);
        turnOffLED(3);
        turnOnLED(3);
        coolerfan_PWMDuty(0.5);
    }

    /*INDICACAO VISUAL SE A TEMPERATURA ESTA ABAIXO DO SETPOINT*/
    else if(ucTempAtual < ucTempAlvo){
        turnOffLED(2);
        turnOffLED(3);
        turnOnLED(2);
        coolerfan_PWMDuty(0);
    }

    /*INDICACAO VISUAL SE A TEMPERATURA ESTA NO SETPOINT*/
    else if(ucTempAtual == ucTempAlvo){
        turnOnLED(2);
        turnOnLED(3);
        coolerfan_PWMDuty(0);
    }

    /*DEVE RETORNAR PARA O MENU DE CONFIGURACAO CASO O BOTAO OK SEJA PRESSIOANDO*/
    if(1 == readSwitch(4)){

```

```
    util_genDelay100ms();  
    ucSubestado1  = DEZENA;  
    ucSubestado2  = STATE_0;  
    ucEstado      = CONFIG;  
    ucLCDFrame    = 1;  
    ledSwi_init(1, 0, 0, 0);  
    break;  
}
```

```
break; /*Break switch Estado*/
```

```
    }  
}  
}
```