Temperature_Control

Giuliano Bohn Benedetti
Keli Tauana Prass Ruppenthal
Victor Dallagnol Bento

Version Final
Tu Jun 26 2018

Table of Contents

Temperature Control

Trabalho final para disciplina de Projeto de Sistemas Embarcados

`https://github.com/da3mons/Embedded_Systems_Design`

Author:

Giuliano Bohn Benedetti
Keli Tauana Prass Ruppenthal
Victor Dallagnol Vento

Date:

Tu Jun 26 BRT 2018

License

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Data Structure Documentation

pt_receiver Struct Reference

Detailed Description

Protothread para comparar valor de temperatura e invocar funções.

The documentation for this struct was generated from the following file:

- **temperature_control.c**

pt_sender Struct Reference

Detailed Description

Protothread para ler temperatura da memória.

The documentation for this struct was generated from the following file:

- **temperature_control.c**

usart_config Struct Reference

The documentation for this struct was generated from the following file:

- **temperature_control.c**

usart_module Struct Reference

The documentation for this struct was generated from the following file:

- **temperature_control.c**

File Documentation

asf.h File Reference

Autogenerated API include file for the Atmel Software Framework (ASF)

```
#include <adc.h>
#include <at30tse75x.h>
#include <compiler.h>
#include <status_codes.h>
#include <delay.h>
#include <eeprom.h>
#include <gfx_mono.h>
#include <sysfont.h>
#include <board.h>
#include <interrupt.h>
#include <nvm.h>
#include <port.h>
#include <parts.h>
#include <sercom.h>
#include <sercom_interrupt.h>
#include <i2c_common.h>
#include <i2c_master.h>
#include <i2c_master_interrupt.h>
#include <spi.h>
#include <spi_interrupt.h>
#include <usart.h>
#include <ssd1306.h>
#include <clock.h>
#include <gclk.h>
#include <system.h>
#include <pinmux.h>
#include <system_interrupt.h>
#include <power.h>
#include <reset.h>
#include <stdio_serial.h>
#include <serial.h>
```

Detailed Description

Autogenerated API include file for the Atmel Software Framework (ASF)

temperature_control.c File Reference

```
#include <asf.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include "pt.h"
```

Macros

- #define **N**   5

Functions

- void **configure_eeprom** (void)
- void **temp_low** ()
- void **temp_high** ()
- void **temp_default** ()
- **PT_WAIT_UNTIL** (pt, **ack**==1)
- **eeprom_emulator_read_page** (0, **aux**)
- **PT_END** (pt)
- **PT_WAIT_UNTIL** (pt, **set**==1)
- **if** (**final_temp**<=15)
- **else if** (**final_temp** >=26)

Variables

- struct **usart_module usart_instance**
- struct **usart_config usart_conf**
- volatile char **temp_lida** [50]
- volatile char **aux** [50]
- float **temp_res**
- float **vec_temp** [**N**]
- int **final_temp**
- int **ack** = 0
- int **set** = 0
- **else**

Macro Definition Documentation

#define N   5

N: Número de amostras

Function Documentation

void configure_eeprom (void)

Configura a memória (paginação).

enum status_code error_code = eeprom_emulator_init():

Inicia serviço de emulação EEPROM

while(true)

Nenhuma seção de EEPROM foi setada para o dispositivo

else if (error_code != STATUS_OK)

Se der erro na memória

eeprom_emulator_read_page (0 , aux)

Lê temperatura da memória e salva em AUX

PT_WAIT_UNTIL (pt , ack = =1)

ProtoThread para ler temperatura da memória. Espera flag que informa que temperatura já foi escrita na memória. Atualiza flag para outra ProtoThread.

PT_WAIT_UNTIL (pt , set = =1)

ProtoThread para comparar a temperatura. Espera flag que informa que temperatura já foi lida da memória.

void temp_default ()

Temperatura padrão (entre 15ºC e 26ºC)

Função para temperatura default do condicionador de ar.

void temp_high ()

Temperatura maior ou igual a 26ºC

Função para setar temperatura do condicionador de ar acima ou igual a 26°.

void temp_low ()

Temperatura menor ou igual a 15°C

Função para setar temperatura do condicionador de ar abaixa ou igual a 15°.

Variable Documentation

ack = 0

Controle para a primeira thread.

aux[50]

Vetor auxilar para ler da temperatura (na ProtoThread).

else

```
Initial value:{
            temp_default()
```

final_temp

Variável para armazenar o valor médio de todas as amostras.

set = 0

Controle para a segunda thread.

temp_lida[50]

Vetor que recebe a temperatura final convertida.

temp_res

Recebe amostra de temperatura.

stdio_serial_init(&) usart_instance EDBG_CDC_MODULE & usart_conf

configura usart, comunicação com sensor externo.

Inicia comunicação com o periferico de Temperatura

usart_enable(& usart_instance

Módulo usart.

Inicia periférico, já configurado para medição de temperatura

vec_temp[N]

Vetor para salvar amostras de temperaturas.

Index

INDEX