

Datalogger de Temperatura programável via porta serial

AUTHOR: Iury Cleveston, Lucas Cielo Borin, Miguel Pfitscher

Table of Contents

Table of contents

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

HEADER (Estrutura que sera salva na memoria)	4
header	5
unionAddress (Separa um endereco de 2 bytes em partes H/L de 1 byte)	6

File Index

File List

Here is a list of all files with brief descriptions:

amostragem.c (Contem as funcoes que manipulam como e realizada a amostragem)	7
amostragem.h (Possui os prototipos das funcoes utilizadas em amostragem.c)	10
botaoIRQ.c (Contem as funcoes que operam ao uso do botao)	13
botaoIRQ.h (Possui os prototipos das funcoes utilizadas em botaoIRQ.c)	15
eeeprom.c (Contem as manipulacoes necessarias para usar a memoria)	17
eeeprom.h (Possui os prototipos das funcoes utilizadas em eeeprom.c)	21
ProcessorExpert.c	25
serial.c (Contem as manipulacoes necessarias para usar corretamente a porta serial)	29
serial.h (Possui os prototipos das funcoes utilizadas em serial.c)	32
timer.c (Este arquivo contem as configuracoes dos tres timers disponiveis ao programa)	35
timer.h (Possui os prototipos das funcoes utilizadas em timer.c)	37
Vectors.c (Arquivo que contem o vetor de interrupcoes modificado)	38

Data Structure Documentation

HEADER Struct Reference

Estrutura que sera salva na memoria.

```
#include <eeprom.h>
```

Detailed Description

Estrutura que sera salva na memoria.

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

- `eeprom.h`

header Struct Reference

```
#include <eeprom.h>
```

Data Fields

- `uint8_t header`
- `char_t tipo`
- `int16_t dado`
- `DATEREC date`
- `TIMEREC time`

Detailed Description

Definition at line 93 of file `eeprom.h`.

Field Documentation

`int16_t dado`

Definition at line 96 of file `eeprom.h`.

`DATEREC date`

Definition at line 97 of file `eeprom.h`.

`uint8_t header`

Definition at line 94 of file `eeprom.h`.

`TIMEREC time`

Definition at line 98 of file `eeprom.h`.

`char_t tipo`

Definition at line 95 of file `eeprom.h`.

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

- `eeprom.h`

unionAddress Union Reference

Separa um endereço de 2 bytes em partes H/L de 1 byte.

```
#include <eeprom.h>
```

Data Fields

- `uint16_t EE`
 - `struct {`
 - `uint8_t H`
 - `uint8_t L`
 - `} Bytes`
-

Detailed Description

Separa um endereço de 2 bytes em partes H/L de 1 byte.

Definition at line 81 of file `eeprom.h`.

Field Documentation

struct { ... } Bytes

uint16_t EE

Definition at line 82 of file `eeprom.h`.

uint8_t H

Definition at line 84 of file `eeprom.h`.

uint8_t L

Definition at line 85 of file `eeprom.h`.

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

- `eeprom.h`

File Documentation

amostragem.c File Reference

Contem as funcoes que manipulam como e realizada a amostragem.

```
#include "amostragem.h"
```

Functions

- void **configuraAmostragem** ()
Configura o conversor A/D.
- void **calibrarAmostragem** (int16_t tempCalibra)
Funcao que calibra o sensor.
- void **paraAmostragem** ()
Interrompe a realizacao da amostragem.
- void **iniciaAmostragem** ()
Inicia amostragem.
- uint16_t **amostra** ()
Realiza a amostragem.
- void * **getAmostra** (void)
Interrupcao que obtem a amostra.

Variables

- uint16_t **TAXA_AMOSTRAGEM** = 5000
Taxa de amostragem padrao de 10ms.
- uint8_t **CANAL_AMOSTRAGEM** = 1
Canal de conversao AD padrao 1.
- bool **amostrando** = FALSE
Flag que informa se a amostragem esta ocorrendo.

Detailed Description

Contem as funcoes que manipulam como e realizada a amostragem.

Possui as funcoes de como e configurada e realizada a amostragem e como a mesma e salva na memoria. Tambem possui as funcoes de calibragem para realizar a amostragem.

See also:

amostragem.h

Function Documentation

unsigned int amostra (void)

Realiza a amostragem.

Configura o conversor A/D para encontrar uma tensao de referencia e uma temperatura amostrada a fim de calcular o valor de temperatura real. Esse calculo varia se a temperatura amostrada for maior ou menor do que 25 graus celsius.

Returns:

Valor real de temperatura obtido

Definition at line 173 of file amostragem.c.

void calibrarAmostragem (int16_t tempCalibra)

Funcao que calibra o sensor.

Definition at line 60 of file amostragem.c.

void configuraAmostragem (void)

Configura o conversor A/D.

Faz a configuracao do conversor A/D usando 10 bits, clock assincrono de 1MHz e habilitando o bandgap

Definition at line 35 of file amostragem.c.

void * getAmostra (void)

Interrupcao que obtem a amostra.

Interrupcao que chama a funcao de realizacao de amostragem e em seguida a funcao de salvar a amostra na memoria.

Definition at line 248 of file amostragem.c.

void iniciaAmostragem (void)

Inicia amostragem.

Inicia a amostragem obtendo a data e hora em que e iniciada e depois de um periodo de tempo, comeca a realizar a amostragem.

Definition at line 137 of file amostragem.c.

void paraAmostragem (void)

Interrompe a realizacao da amostragem.

Para a realizacao da amostragem, mudando seu status para FALSE e desligando o timer de amostragem.

Definition at line 115 of file amostragem.c.

Variable Documentation

bool amostrando = FALSE

Flag que informa se a amostragem esta ocorrendo.

Definition at line 21 of file amostragem.c.

uint8_t CANAL_AMOSTRAGEM = 1

Canal de conversao AD padrao 1.

Definition at line 18 of file amostragem.c.

uint16_t TAXA_AMOSTRAGEM = 5000

Taxa de amostragem padrao de 10ms.

Definition at line 15 of file amostragem.c.

amostragem.h File Reference

Possui os prototipos das funcoes utilizadas em **amostragem.c**.

```
#include "Cpu.h"
#include "Events.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include <stdio.h>
#include "eeprom.h"
#include "serial.h"
#include "timer.h"
#include "relogio.h"
```

Functions

- void **configuraAmostragem** (void)
Configura o conversor A/D.
- void **iniciaAmostragem** (void)
Inicia amostragem.
- void **paraAmostragem** (void)
Interrompe a realizacao da amostragem.
- void **calibrarAmostragem** (uint16_t tempCalibra)
Funcao que calibra o sensor.
- uint16_t **amostra** (void)
Realiza a amostragem.
- void * **getAmostra** (void)
Interrupcao que obtem a amostra.

Variables

- uint16_t **TAXA_AMOSTRAGEM**
Taxa de amostragem padrao de 10ms.
- uint8_t **CANAL_AMOSTRAGEM**
Canal de conversao AD padrao 1.
- bool **amostrando**
Flag que informa se a amostragem esta ocorrendo.

Detailed Description

Possui os prototipos das funcoes utilizadas em **amostragem.c**.

Define os prototipos das funcoes utilizadas pelo arquivo **amostragem.c** bem como as suas variaveis externas.

Function Documentation

uint16_t amostra (void)

Realiza a amostragem.

Configura o conversor A/D para encontrar uma tensao de referencia e uma temperatura amostrada a fim de calcular o valor de temperatura real. Esse calculo varia se a temperatura amostrada for maior ou menor do que 25 graus celsius.

Returns:

Valor real de temperatura obtido

Definition at line 173 of file amostragem.c.

void calibrarAmostragem (int16_t tempCalibra)

Funcao que calibra o sensor.

Definition at line 60 of file amostragem.c.

void configuraAmostragem (void)

Configura o conversor A/D.

Faz a configuracao do conversor A/D usando 10 bits, clock assincrono de 1MHz e habilitando o bandgap

Definition at line 35 of file amostragem.c.

void* getAmostra (void)

Interrupcao que obtem a amostra.

Interrupcao que chama a funcao de realizacao de amostragem e em seguida a funcao de salvar a amostra na memoria.

Definition at line 248 of file amostragem.c.

void iniciaAmostragem (void)

Inicia amostragem.

Inicia a amostragem obtendo a data e hora em que e iniciada e depois de um periodo de tempo, comeca a realizar a amostragem.

Definition at line 137 of file amostragem.c.

void paraAmostragem (void)

Interrompe a realizacao da amostragem.

Para a realizacao da amostragem, mudando seu status para FALSE e desligando o timer de amostragem.

Definition at line 115 of file amostragem.c.

Variable Documentation

bool amostrando

Flag que informa se a amostragem esta ocorrendo.

Definition at line 21 of file amostragem.c.

uint8_t CANAL_AMOSTRAGEM

Canal de conversao AD padrao 1.

Definition at line 18 of file amostragem.c.

uint16_t TAXA_AMOSTRAGEM

Taxa de amostragem padrao de 10ms.

Definition at line 15 of file amostragem.c.

botaoIRQ.c File Reference

Contem as funcoes que operam ao uso do botao.

```
#include "botaoIRQ.h"
```

Functions

- void **configuraIRQ** (void)
Configura IRQ.
- void **resetIRQ** (void)
Reseta valores do IRQ para novo clique.
- **ISR** (interruptIRQ)

Variables

- uint8_t **segundos** = 0
 - uint8_t **mSegundos** = 0
 - uint8_t **botaoApertado** = 0
 - bool **apertado** = FALSE
-

Detailed Description

Contem as funcoes que operam ao uso do botao.

Contem as manipulacoes que foram necessarias para usar o botaoIRQ contido na placa da Freescale.

See also:

botaoIRQ.h

Function Documentation

void configuraIRQ (void)

Configura IRQ.

Funcao que inicializa e manipula registradores para utilizar o botao corretamente.

Definition at line 38 of file botaoIRQ.c.

ISR (interruptIRQ)

Definition at line 146 of file botaoIRQ.c.

void resetIRQ (void)

Reseta valores do IRQ para novo clique.

Zera a variavel que conta o numero de cliques no botao, reseta a flag de pressionado, desabilita contador de cliques e zera a variavel que conta o tempo de aperto.

Definition at line 119 of file botaoIRQ.c.

Variable Documentation

bool apertado = FALSE

Definition at line 29 of file botaoIRQ.c.

uint8_t botaoApertado = 0

Definition at line 23 of file botaoIRQ.c.

uint8_t mSegundos = 0

Definition at line 20 of file botaoIRQ.c.

uint8_t segundos = 0

Definition at line 17 of file botaoIRQ.c.

botaoIRQ.h File Reference

Possui os prototipos das funcoes utilizadas em **botaoIRQ.c**.

```
#include "Cpu.h"
#include "Events.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "WDog.h"
#include "timer.h"
#include "relogio.h"
```

Functions

- void **configuraIRQ** (void)
Configura IRQ.
- void **resetIRQ** (void)
Reseta valores do IRQ para novo clique.

Variables

- uint8_t **segundos**
- uint8_t **botaoApertado**
- bool **apertado**

Detailed Description

Possui os prototipos das funcoes utilizadas em **botaoIRQ.c**.

Define os prototipos das funcoes utilizadas pelo arquivo **botaoIRQ.c** bem como as suas variaveis externas.

Function Documentation

void configuraIRQ (void)

Configura IRQ.

Funcao que inicializa e manipula registradores para utilizar o botao corretamente.

Definition at line 38 of file botaoIRQ.c.

void resetIRQ (void)

Reseta valores do IRQ para novo clique.

Zera a variavel que conta o numero de cliques no botao, reseta a flag de pressionado, desabilita contador de cliques e zera a variavel que conta o tempo de aperto.

Definition at line 119 of file botaoIRQ.c.

Variable Documentation

bool apertado

Definition at line 29 of file botaoIRQ.c.

uint8_t botaoApertado

Definition at line 23 of file botaoIRQ.c.

uint8_t segundos

Definition at line 17 of file botaoIRQ.c.

eeeprom.c File Reference

Contem as manipulacoes necessarias para usar a memoria.

```
#include "eeeprom.h"
```

Functions

- void **configuraMem** ()
Configura a memoria eeeprom.
- void **delay** ()
Funcao de espera.
- void **salvaAmostra** (int8_t dadoAmostra)
Salva o dado amostrado na memoria eeeprom.
- void **salvaHeader** (HEADER *c)
Salva struct na memoria.
- void **limparMem** (bool mensagemSerial)
Limpa os dados da memoria eeeprom.
- byte **auxMem** (bool read, byte dado)
funcao auxiliar usada para ler ou escrever na memoria
- void **escreveMem** (uint16_t endereco, byte dado)
Escreve um dado desejado na memoria.
- byte **leMem** (uint16_t endereco)
- bool **leHeader** (uint16_t endereco, HEADER *dadoTemporario)
Le header do inicio do bloco dos dados.
- bool **leDado** (uint16_t endereco, int8_t *dadoTemporario)
Le dado do endereco informado.

Variables

- unionAddress **uAddr**
- unionAddress **uDados**
- uint16_t **address** = ENDERECO
- bool **travado** = TRUE
- bool **memCheia** = FALSE

Detailed Description

Contem as manipulacoes necessarias para usar a memoria.

Possui as funcoes que controlam a memoria eeeprom, fazendo sua configuracao, escrita e leitura, alem de outras funcoes implementadas necessarias para seu correto funcionamento.

See also:

eeeprom.h

Function Documentation

byte auxMem (bool *read*, byte *dado*)

funcao auxiliar usada para ler ou escrever na memoria

Funcao que auxilia as funcoes de leitura e escrita na memoria, sendo chamada quando conveniente.

Parameters:

<i>read</i>	indica se necessita-se de leitura ou escrita
<i>dado</i>	e o valor do dado ou endereco que ira ser lido ou escrito

Definition at line 271 of file eeprom.c.

void configuraMem (void)

Configura a memoria eeprom.

Faz as modificacoes necessarias no IIC para fazer a configuracao necessaria da memoria eeprom

Definition at line 35 of file eeprom.c.

void delay (void)

Funcao de espera.

Manipula os registradores TPM para fazer uma espera de tempo.

Definition at line 124 of file eeprom.c.

void escreveMem (uint16_t *endereco*, byte *dado*)

Escreve um dado desejado na memoria.

Definition at line 328 of file eeprom.c.

bool leDado (uint16_t *endereco*, int8_t * *dadoTemporario*)

Le dado do endereco informado.

Definition at line 476 of file eeprom.c.

bool leHeader (uint16_t *endereco*, HEADER * *dadoTemporario*)

Le header do inicio do bloco dos dados.

Definition at line 432 of file eeprom.c.

byte leMem (uint16_t *endereco*)

Definition at line 383 of file eeprom.c.

void limparMem (bool *mensagemSerial*)

Limpa os dados da memoria eeprom.

Limpa os dados da memoria eeprom, substituindo os dados pelo valor padrao 127.

Parameters:

<i>TRUE/FALSE</i>	para mostrar a limpeza
-------------------	------------------------

Definition at line 234 of file eeprom.c.

void salvaAmostra (int8_t *dadoAmostra*)

Salva o dado amostrado na memoria eeprom.

Definition at line 167 of file eeprom.c.

void salvaHeader (HEADER * *c*)

Salva struct na memoria.

void salvaHeader(HEADER *c) Salva a struct na memoria contendo o cabecalho do dado, junto com seu tipo, data e hora usando a funcao **escreveMem()**.

Parameters:

<i>Struct</i>	com valores a ser salvos
---------------	--------------------------

Definition at line 191 of file eeprom.c.

Variable Documentation

uint16_t address = ENDERECO

Definition at line 19 of file eeprom.c.

bool memCheia = FALSE

Definition at line 24 of file eeprom.c.

bool travado = TRUE

Definition at line 21 of file eeprom.c.

unionAddress uAddr

Definition at line 14 of file eeprom.c.

unionAddress uDados

Definition at line 17 of file eeprom.c.

eeeprom.h File Reference

Possui os prototipos das funcoes utilizadas em **eeeprom.c**.

```
#include "Cpu.h"
#include "Events.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "WDog.h"
#include "relogio.h"
#include "timer.h"
#include "serial.h"
#include "amostragem.h"
```

Data Structures

- union **unionAddress**
- *Separa um endereco de 2 bytes em partes H/L de 1 byte.* struct **header**

Macros

- #define **AMOSTRA** 1
- #define **EVENTO** 2
- #define **FALTA_ENERGIA** 1
- #define **MEMORIA_CHEIA** 2
- #define **enderecoMax** 0x07FF
- #define **ENDERECO** 6
- #define **escravoIIC** 0xA0
- #define **MEM_DELAY** 800
- #define **INIT_HEADER** 127
- #define **INIT_FORMAT** 126

Typedefs

- typedef struct **header** **HEADER**

Functions

- void **configuraMem** (void)
Configura a memoria eeeprom.
- void **delay** (void)
Funcao de espera.
- void **limparMem** (bool mensagemSerial)
Limpa os dados da memoria eeeprom.
- bool **leDado** (uint16_t endereco, int8_t *dadoTemporario)
Le dado do endereco informado.
- bool **leHeader** (uint16_t endereco, **HEADER** *dadoTemporario)
Le header do inicio do bloco dos dados.
- void **escreveMem** (uint16_t endereco, byte dado)
Escreve um dado desejado na memoria.
- void **salvaHeader** (**HEADER** *cabecalho)

Salva struct na memoria.

- void **salvaAmostra** (int8_t dadoAmostra)
Salva o dado amostrado na memoria eeprom.

Variables

- uint16_t **address**
 - bool **memCheia**
-

Detailed Description

Possui os prototipos das funcoes utilizadas em **eeprom.c**.

Define os prototipos das funcoes utilizadas pelo arquivo **eeprom.c** bem como as suas macros e variaveis externas.

Macro Definition Documentation

#define AMOSTRA 1

Definition at line 29 of file eeprom.h.

#define ENDERECO 6

Definition at line 55 of file eeprom.h.

#define enderecoMax 0x07FF

Definition at line 50 of file eeprom.h.

#define escravoIIC 0xA0

Definition at line 60 of file eeprom.h.

#define EVENTO 2

Definition at line 35 of file eeprom.h.

#define FALTA_ENERGIA 1

Definition at line 40 of file eeprom.h.

#define INIT_FORMAT 126

Definition at line 75 of file eeprom.h.

#define INIT_HEADER 127

Definition at line 70 of file eeprom.h.

#define MEM_DELAY 800

Definition at line 65 of file eeprom.h.

#define MEMORIA_CHEIA 2

Definition at line 45 of file eeprom.h.

Typedef Documentation

typedef struct header HEADER

Function Documentation

void configuraMem (void)

Configura a memoria eeprom.

Faz as modificacoes necessarias no IIC para fazer a configuracao necessaria da memoria eeprom

Definition at line 35 of file eeprom.c.

void delay (void)

Funcao de espera.

Manipula os registradores TPM para fazer uma espera de tempo.

Definition at line 124 of file eeprom.c.

void escreveMem (uint16_t endereco, byte dado)

Escreve um dado desejado na memoria.

Definition at line 328 of file eeprom.c.

bool leDado (uint16_t endereco, int8_t * dadoTemporario)

Le dado do endereco informado.

Definition at line 476 of file eeprom.c.

bool leHeader (uint16_t *endereco*, HEADER * *dadoTemporario*)

Le header do inicio do bloco dos dados.

Definition at line 432 of file eeprom.c.

void limparMem (bool *mensagemSerial*)

Limpa os dados da memoria eeprom.

Limpa os dados da memoria eeprom, substituindo os dados pelo valor padrao 127.

Parameters:

<i>TRUE/FALSE</i>	para mostrar a limpeza
-------------------	------------------------

Definition at line 234 of file eeprom.c.

void salvaAmostra (int8_t *dadoAmostra*)

Salva o dado amostrado na memoria eeprom.

Definition at line 167 of file eeprom.c.

void salvaHeader (HEADER * *c*)

Salva struct na memoria.

void **salvaHeader(HEADER **c*)** Salva a struct na memoria contendo o cabecalho do dado, junto com seu tipo, data e hora usando a funcao **escreveMem()**.

Parameters:

<i>Struct</i>	com valores a ser salvos
---------------	--------------------------

Definition at line 191 of file eeprom.c.

Variable Documentation

uint16_t address

Definition at line 19 of file eeprom.c.

bool memCheia

Definition at line 24 of file eeprom.c.

ProcessorExpert.c File Reference

```
#include "Cpu.h"
#include "Events.h"
#include "Vtpm1ovf.h"
#include "Vtpm2ovf.h"
#include "Vtpm3ovf.h"
#include "relogio.h"
#include "ASerial.h"
#include "Vlvd.h"
#include "Virq.h"
#include "WDog.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "stdlib.h"
#include "stdio.h"
#include "timer.h"
#include "eeprom.h"
#include "amostragem.h"
#include "botaoIRQ.h"
#include "serial.h"
```

Enumerations

- enum { **eMEDINDO** = 0, **ePARADO** }

Functions

- void **calibrar** (void)
Funcao para calibrar o sensor.
- void **sair** (void)
Funcao que desloga o usuario.
- void **setDate** (void)
Funcao que configura a data.
- void **setTime** (void)
Funcao que configura o horario.
- void **setSenha** (void)
Funcao que altera a senha.
- void **setAmostragem** (void)
Funcao que altera a frequencia da taxa de amostragem.
- void **mostrar** (void)
Funcao que mostra os dados na tela.
- void **status** (void)
Funcao que mostra o status do sistema.
- void **ajuda** (void)
Funcao que mostra menu de ajuda.
- void **configuraLVD** (void)
Configura a funcao de tensao baixa.

- **ISR** (tensaoBaixa)
Interrupcao gerada ao detectar queda de tensao.
- void **main** (void)
Funcao principal do programa.

Variables

- enum { ... } **estado**

Enumeration Type Documentation

anonymous enum

Enumerator

eMEDINDO

ePARADO

Definition at line 42 of file ProcessorExpert.c.

Function Documentation

void ajuda (void)

Funcao que mostra menu de ajuda.

Definition at line 150 of file ProcessorExpert.c.

void calibrar (void)

Funcao para calibrar o sensor.

Definition at line 114 of file ProcessorExpert.c.

void configuraLVD (void)

Configura a funcao de tensao baixa.

Definition at line 61 of file ProcessorExpert.c.

ISR (tensaoBaixa)

Interrupcao gerada ao detectar queda de tensao.

Definition at line 73 of file ProcessorExpert.c.

void main (void)

Funcao principal do programa.

Aqui esta contido a FSM que foi especificada para o trabalho, ela possui dois estados, PARADO e MEDINDO. Estas ficam em um laco infinito esperando que o usuario envie mensagens que ativem as demais funcoes descritas nessa documentacao.

Definition at line 498 of file ProcessorExpert.c.

void mostrar (void)

Funcao que mostra os dados na tela.

Definition at line 386 of file ProcessorExpert.c.

void sair (void)

Funcao que desloga o usuario.

Definition at line 101 of file ProcessorExpert.c.

void setAmostragem (void)

Funcao que altera a frequencia da taxa de amostragem.

Definition at line 351 of file ProcessorExpert.c.

void setDate (void)

Funcao que configura a data.

Definition at line 229 of file ProcessorExpert.c.

void setSenha (void)

Funcao que altera a senha.

Definition at line 318 of file ProcessorExpert.c.

void setTime (void)

Funcao que configura o horario.

Definition at line 272 of file ProcessorExpert.c.

void status (void)

Funcao que mostra o status do sistema.

Definition at line 173 of file ProcessorExpert.c.

Variable Documentation

enum { ... } estado

serial.c File Reference

Contem as manipulacoes necessarias para usar corretamente a porta serial.

```
#include "serial.h"
```

Functions

- void * **incSerial** ()
Marca o tempo de espera para a serial.
- void * **incConf** ()
Marca o tempo que o usuario possui para entrar no modo de configuracao.
- void **verificaSerial** ()
Funcao que verifica se o buffer recebeu algum caracter.
- bool **leSerial** ()
Funcao que le string da serial e retorna se houve exito.
- void **escreveSerial** (char *mensagem)
Funcao que escreve mensagem da serial.

Variables

- char_t **buffer** [TAMANHO_BUFFER]
Buffer que guardara os caracteres.
- char_t **SENHA** [TAMANHO_BUFFER] = "barrichelo"
Senha para configuracao.
- bool **autenticado** = FALSE
Flag que indica que o usuario esta no modo de configuracao.
- uint8_t **segundoSerial** = 0
Variavel que armazena tempo de espera da serial.
- uint8_t **segundoConf** = 0
Variavel que armazena tempo para entrar no modo de configuracao.

Detailed Description

Contem as manipulacoes necessarias para usar corretamente a porta serial.

Aqui esta contido como e feita a leitura e escrita no buffer de dados atraves da porta serial. Tambem possui as funcoes que verificam a autenticidade do usuario para obter acesso ao programa e a de verificacao do buffer no qual o programa interpreta os comandos enviados pelo usuario.

See also:

serial.h

Function Documentation

void escreveSerial (char * mensagem)

Funcao que escreve mensagem da serial.

Escreve os caracteres da mensagem na serial ate identificar um caracter de fim de string, entao encerra a escrita e faz a limpeza do buffer.

Parameters:

<i>caracteres</i>	da mensagem a ser escrita
-------------------	---------------------------

Definition at line 286 of file serial.c.

void * incConf (void)

Marca o tempo que o usuario possui para entrar no modo de configuracao.

Definition at line 84 of file serial.c.

void * incSerial (void)

Marca o tempo de espera para a serial.

Definition at line 74 of file serial.c.

bool leSerial (void)

Funcao que le string da serial e retorna se houve exito.

Essa funcao le os caracteres mandados pelo usuario na porta serial, a leitura acaba quando o caracter de enter for lido e a funcao retorna TRUE. Retorna FALSE caso um ctrl-c seja detectado ou se houve estouro do timer por falta de comandos do usuario.

Returns:

TRUE se durante o tempo do timer receber o caracter de ENTER, senao FALSE

Definition at line 195 of file serial.c.

void verificaSerial (void)

Funcao que verifica se o buffer recebeu algum caracter.

Faz a verificacao do buffer a procura de algum caracter esperado, por exemplo, o '+' quando repetido tres vezes dentro de uma faixa de tempo da acesso ao usuario poder fazer sua autenticacao.

Definition at line 100 of file serial.c.

Variable Documentation

bool autenticado = FALSE

Flag que indica que o usuario esta no modo de configuracao.

Definition at line 21 of file serial.c.

char_t buffer[TAMANHO_BUFFER]

Buffer que guardara os caracteres.

Definition at line 15 of file serial.c.

uint8_t segundoConf = 0

Variavel que armazena tempo para entrar no modo de configuracao.

Definition at line 27 of file serial.c.

uint8_t segundoSerial = 0

Variavel que armazena tempo de espera da serial.

Definition at line 24 of file serial.c.

char_t SENHA[TAMANHO_BUFFER] = "barrichelo"

Senha para configuracao.

Definition at line 18 of file serial.c.

serial.h File Reference

Possui os prototipos das funcoes utilizadas em **serial.c**.

```
#include "Cpu.h"
#include "Events.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "WDog.h"
#include "ASerial.h"
#include "timer.h"
#include "ctype.h"
```

Macros

- **#define TAMANHO_BUFFER 512**
Define tamanho do buffer de informacoes.
- **#define TEMPO_ESPERA 10**
Tempo de espera da serial.

Functions

- void **verificaSerial** (void)
Funcao que verifica se o buffer recebeu algum caracter.
- bool **leSerial** (void)
Funcao que le string da serial e retorna se houve sucesso.
- void **escreveSerial** (char *mensagem)
Funcao que escreve mensagem da serial.

Variables

- char_t **SENHA** [TAMANHO_BUFFER]
Senha para configuracao.
- bool **autenticado**
Flag que indica que o usuario esta no modo de configuracao.
- char_t **buffer** [TAMANHO_BUFFER]
Buffer que guardara os caracteres.

Detailed Description

Possui os prototipos das funcoes utilizadas em **serial.c**.

Define os prototipos das funcoes utilizadas pelo arquivo **serial.c** bem como as suas macros e variaveis externas.

Macro Definition Documentation

#define TAMANHO_BUFFER 512

Define tamanho do buffer de informacoes.

Definition at line 30 of file serial.h.

#define TEMPO_ESPERA 10

Tempo de espera da serial.

Definition at line 36 of file serial.h.

Function Documentation

void escreveSerial (char * *mensagem*)

Funcao que escreve mensagem da serial.

Escreve os caracteres da mensagem na serial ate identificar um caracter de fim de string, entao encerra a escrita e faz a limpeza do buffer.

Parameters:

<i>caracteres</i>	da mensagem a ser escrita
-------------------	---------------------------

Definition at line 286 of file serial.c.

bool leSerial (void)

Funcao que le string da serial e retorna se houve sucesso.

Essa funcao le os caracteres mandados pelo usuario na porta serial, a leitura acaba quando o caracter de enter for lido e a funcao retorna TRUE. Retorna FALSE caso um ctrl-c seja detectado ou se houve estouro do timer por falta de comandos do usuario.

Returns:

TRUE se durante o tempo do timer receber o caracter de ENTER, senao FALSE

Definition at line 195 of file serial.c.

void verificaSerial (void)

Funcao que verifica se o buffer recebeu algum caracter.

Faz a verificacao do buffer a procura de algum caracter esperado, por exemplo, o '+' quando repetido tres vezes dentro de uma faixa de tempo da acesso ao usuario poder fazer sua autenticacao.

Definition at line 100 of file serial.c.

Variable Documentation

bool autenticado

Flag que indica que o usuario esta no modo de configuracao.

Definition at line 21 of file serial.c.

char_t buffer[TAMANHO_BUFFER]

Buffer que guardara os caracteres.

Definition at line 15 of file serial.c.

char_t SENHA[TAMANHO_BUFFER]

Senha para configuracao.

Definition at line 18 of file serial.c.

timer.c File Reference

Este arquivo contem as configuracoes dos tres timers disponiveis ao programa.

```
#include "timer.h"
```

Functions

- void **conta** (uint8_t timer, uint16_t time, void *funcao)
Funcao que inicia contagem do timer escolhido. Ao final da contagem, executa a funcao passada como parametro.
- void **desligaTimer** (uint8_t timer)
Funcao que desligar o timer.
- **ISR** (timer1)
Interrupcao do Timer 1.
- **ISR** (timer2)
Interrupcao do Timer 2.
- **ISR** (timer3)
Interrupcao do Timer 3.

Detailed Description

Este arquivo contem as configuracoes dos tres timers disponiveis ao programa.

Aqui esta contido como e implementado os tres timers que sao utilizados durante a execucao do programa.

See also:

timer.h

Function Documentation

void conta (uint8_t *timer*, uint16_t *time*, void * *funcao*)

Funcao que inicia contagem do timer escolhido. Ao final da contagem, executa a funcao passada como parametro.

Definition at line 40 of file timer.c.

void desligaTimer (uint8_t *timer*)

Funcao que desligar o timer.

Definition at line 92 of file timer.c.

ISR (timer1)

Interrupcao do Timer 1.

Definition at line 114 of file timer.c.

ISR (timer2)

Interrupcao do Timer 2.

Definition at line 139 of file timer.c.

ISR (timer3)

Interrupcao do Timer 3.

Definition at line 164 of file timer.c.

timer.h File Reference

Possui os prototipos das funcoes utilizadas em **timer.c**.

```
#include "Cpu.h"
#include "Events.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
```

Functions

- void **conta** (uint8_t timer, uint16_t time, void *funcao)
Funcao que inicia contagem do timer escolhido. Ao final da contagem, executa a funcao passada como parametro.
- void **desligaTimer** (uint8_t timer)
Funcao que desligar o timer.

Detailed Description

Possui os prototipos das funcoes utilizadas em **timer.c**.

Define os prototipos das funcoes utilizadas pelo arquivo **timer.c**

Function Documentation

void conta (uint8_t *timer*, uint16_t *time*, void * *funcao*)

Funcao que inicia contagem do timer escolhido. Ao final da contagem, executa a funcao passada como parametro.

Definition at line 40 of file timer.c.

void desligaTimer (uint8_t *timer*)

Funcao que desligar o timer.

Definition at line 92 of file timer.c.

Vectors.c File Reference

Arquivo que contem o vetor de interrupcoes modificado.

```
#include "Cpu.h"
#include "Vtpm1ovf.h"
#include "Vtpm2ovf.h"
#include "Vtpm3ovf.h"
#include "relogio.h"
#include "ASerial.h"
#include "Vlvd.h"
#include "Virq.h"
#include "WDog.h"
```

Detailed Description

Arquivo que contem o vetor de interrupcoes modificado.

Componente gerado pelo 'MC9S08QE128_80' que fornece os metodos basicos e eventos para a configuracao do nucleo da CPU.

Index

INDEX