PSE AV2 - Fernanda Padilha

210920c

Generated by Doxygen 1.8.20

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 pinRead Struct Reference

Estrutura utilizada para ler dados do sensor.

Data Fields

- int pin
  Pino lido da placa.
- float value
  Valor lido da placa.

3.1.1 Detailed Description

Estrutura utilizada para ler dados do sensor.

3.1.2 Field Documentation

3.1.2.1 pin

```
int pin
```

Pino lido da placa.

3.1.2.2 value

```
float value
```

Valor lido da placa.

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

- E:/Aulas/PSE/elc1048/Projeto_Av2/PF_Buzzer_Doc/PF_Buzzer_Doc.c

Chapter 4

File Documentation

4.1 E:/Aulas/PSE/elc1048/Projeto_Av2/PF_Buzzer_Doc/PF_Buzzer_Doc.c File Reference

```
#include <Arduino_FreeRTOS.h>
#include <semphr.h>
#include <queue.h>
#include <task.h>
```

Data Structures

- struct pinRead

  Estrutura utilizada para ler dados do sensor.

Functions

- void TaskAnalogRead (void *pvParameters)
- void TaskTempAtual (void *pvParameters)
- void TaskTempMedia (void *pvParameters)
- void TaskBuzzer (void *pvParameters)
- void setup ()

  Função que executa quando liga a placa ou aperta o botão reset.
- void loop ()
- void TaskAnalogRead (void *pvParameters __attribute__((unused)))

  Tarefa que lê dados do sensor.
- void TaskTempAtual (void *pvParameters __attribute__((unused)))

  Tarefa que consome dado da fila se disponível;.
- void TaskTempMedia (void *pvParameters __attribute__((unused)))

  Tarefa que consome o buffer para cálculo da média.
- void TaskBuzzer (void *pvParameters __attribute__((unused)))

  Tarefa que controla o buzzer.

Variables

- SemaphoreHandle_t xSerialSemaphore
- int flag

  Controla o buffer para cálculo da Média.
- int k

  Contador de preenchimento do buffer da Média.
- int i

  Variável de controle do buzzer.

- float bufferTemp [10]
- const int pinBuzzer = 11

  Indica a porta digital ligada ao buzzer.
- QueueHandle_t structQueue

4.1.1 Function Documentation

4.1.1.1 loop()

```
void loop ( )
```

Vazio. Tudo é feito nas tarefas.

4.1.1.2 setup()

```
void setup ( )
```

Função que executa quando liga a placa ou aperta o botão reset.

Inicia a comunicação serial a 9600 bits por segundo.

Espera a porta serial conectar.

Confirma que a conexão foi estabelecida.(Debug)

Checa se o semáforo da porta serial já não foi criado.

Cria a mutex que controla a porta serial.

Torna a porta serial disponível, "dando" o semáforo.

Cria a fila de dados do sensor.

Verifica se a fila foi criada.

Cria tarefas que serão executadas independentemente.

Cria a tarefa para consumir dados da fila.

Cria a tarefa para cálculo da média.

Cria a tarefa produtora de dados da fila.

Cria a tarefa que emite sons no buzzer.

Agora, o escalonador de tarefas, que assume o controle do escalonamento de tarefas individuais, é iniciado automaticamente.

4.1.1.3 TaskAnalogRead() [1/2]

```
void TaskAnalogRead (
            void *pvParameters  __attribute__(unused) )
```

Tarefa que lê dados do sensor.

Codificação dos valores lidos em tensão para temperatura. Fonte: https://portal.vidadesilicio.com.br/lm35-medindo-temperatura-com-arduino/

Posta um item na fila. https://www.freertos.org/a00117.html

Um tick de atraso (15ms) entre as leituras para estabilidade.

4.1.1.4 TaskAnalogRead() [2/2]

```
void TaskAnalogRead (
            void * pvParameters )
```

4.1.1.5 TaskBuzzer() [1/2]

```
void TaskBuzzer (
            void *pvParameters  __attribute__(unused) )
```

Tarefa que controla o buzzer.

Fonte:http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-squids/basico/137-projeto-36-controlando-frequencia-de-um-buzzer-com-potenciometro

Frequência tocada no buzzer.

Variável para guardar o valor consumido do buffer.

Consome dado do buffer.

Caso a temperatura atinja valor superior a 29, codifica os valores de temperatura

entre 30 e 80 para valores de frequencia entre 0 e 2500.

Buzzer emite som para cada leitura acima de 29.

Se não for detectada temperatura superior a 29, o som do buzzer apenas será alterado na chamada de noTone().

Se não há dados no buffer (k=0), buzzer não consome dados.

Reseta variável de controle do buzzer.

Silencia o buzzer até a próxima chamada de tone().

4.1.1.6 TaskBuzzer() [2/2]

```
void TaskBuzzer (
            void * pvParameters )
```

4.1.1.7 TaskTempAtual() [1/2]

```
void TaskTempAtual (
            void *pvParameters  __attribute__(unused) )
```

Tarefa que consome dado da fila se disponível;.

Read an item from a queue. https://www.freertos.org/a00118.html

Verifica se ainda não foram armazenados 10 dados no buffer da média.

A variável de controle do buzzer recebe contador do buffer.

Caso não, flag continua em 0.

Se o semáforo estiver disponível, a tarefa consegue o controle da porta serial.

Comunica o valor lido da fila.

Posição do buffer no momento.

Libera a porta serial.

Incrementa a variável de controle do buffer.

Caso o contador atinja 10,

reseta a variável de controle do buzzer para evitar leitura do buffer.

Altera a flag e sinaliza que a média pode ser calculada.

4.1.1.8 TaskTempAtual() [2/2]

```
void TaskTempAtual (
            void * pvParameters )
```

4.1.1.9 TaskTempMedia() [1/2]

```
void TaskTempMedia (
            void *pvParameters  __attribute__(unused) )
```

Tarefa que consome o buffer para cálculo da média.

Variável que guarda a média.

Variável que guarda o acumulado dos dados do buffer.

Verifica se a flag foi alterada para 1.

Executa laço para cálculo da média dos valores guardados no buffer.

Reseta a flag para confirmar que os dados do buffer foram consumidos e podem ser substituidos.

Reseta variável de controle do buffer.

Verifica se a porta serial está disponível. Caso obtenha o controle do semáforo,

comunica o valor da média pela porta serial.

Reseta a variável da média.

Reseta variável do acumulado.

Libera a porta serial.

Se ainda não foram feitas 10 leituras, a média não será calculada.

4.1.1.10 TaskTempMedia() [2/2]

```
void TaskTempMedia (
            void * pvParameters )
```

4.1.2 Variable Documentation

4.1.2.1 bufferTemp

```
float bufferTemp[10]
```

Vetor que guarda 10 dados lidos do sensor para ser calculada a média pela task TempMedia.

4.1.2.2 flag

```
int flag
```

Controla o buffer para cálculo da Média.

4.1.2.3 i

```
int i
```

Variável de controle do buzzer.

4.1.2.4 k

```
int k
```

Contador de preenchimento do buffer da Média.

4.1.2.5 pinBuzzer

```
const int pinBuzzer = 11
```

Indica a porta digital ligada ao buzzer.

4.1.2.6 structQueue

```
QueueHandle_t structQueue
```

Handle da fila que a task AnalogRead envia dados lidos do sensor.

4.1.2.7 xSerialSemaphore

```
SemaphoreHandle_t xSerialSemaphore
```

Declaração da mutex Semaphore Handle que vai controlar a Serial Port. Garante que apenas uma tarefa controla a serial a cada vez.

Index