

seriallibRTAI Reference Manual

Generated by Doxygen 1.4.7

Mon Aug 4 17:54:12 2008

Contents

1	seriallibRTAI Main Page	1
2	seriallibRTAI Data Structure Index	3
2.1	seriallibRTAI Data Structures	3
3	seriallibRTAI File Index	5
3.1	seriallibRTAI File List	5
4	seriallibRTAI Data Structure Documentation	7
4.1	SERIALPORTCONFIG Struct Reference	7
5	seriallibRTAI File Documentation	9
5.1	seriallibRTAI.c File Reference	9
5.2	seriallibRTAI.h File Reference	14

Chapter 1

seriallibRTAI Main Page

A biblioteca seriallibRTAI foi concebida para dar funcionalidade de comunicação serial para processos LINUX com a extensão de tempo real RTAI. Ela é disponibilizada na forma de código fonte nos arquivos **seriallibRTAI.c** (p. 9) e **seriallibRTAI.h** (p. 14). Essa biblioteca foi concebida para ser compatível com processos com múltiplos threads, e permite ainda que vários threads acessem a mesma porta serial. No caso, estão implementadas funções para COM1, COM2, COM3 e COM4. E ainda, a biblioteca implementa funções de controle de acesso por semáforo, o que permite que uma mesma porta serial possa ser acessada por um só thread por vez. Dependendo do tipo de protocolo, o uso de semáforos se faz necessário.

O projeto acompanha um exemplo no diretório test. Para compilar o exemplo, basta fazer make. O resultado é o arquivo eval_seriallibRTAI. Antes de executar esse arquivo é necessário pelo menos uma vez após ter iniciado o sistema carregar os módulos do RTAI. Para isso, basta executar o script loadmods.

Chapter 2

seriallibRTAI Data Structure Index

2.1 seriallibRTAI Data Structures

Here are the data structures with brief descriptions:

SERIALPORTCONFIG	7
-------------------------	-------	---

Chapter 3

seriallibRTAI File Index

3.1 seriallibRTAI File List

Here is a list of all files with brief descriptions:

seriallibRTAI.c (Arquivo com as funções da biblioteca seriallibRTAI)	9
seriallibRTAI.h (Arquivo cabeçalho da biblioteca seriallibRTAI)	14

Chapter 4

seriallibRTAI Data Structure Documentation

4.1 SERIALPORTCONFIG Struct Reference

```
#include <seriallibRTAI.h>
```

Data Fields

- unsigned int **ComPortNumber**
- unsigned int **ComPortBPS**
- unsigned int **ComPortAddress**
- float **FramePeriodUS**

4.1.1 Field Documentation

4.1.1.1 unsigned int SERIALPORTCONFIG::ComPortAddress

Endereço base da porta de comunicação serial

4.1.1.2 unsigned int SERIALPORTCONFIG::ComPortBPS

Taxa de comunicação em BPS, de 2 a 115200

4.1.1.3 unsigned int SERIALPORTCONFIG::ComPortNumber

Número da porta serial, de 1 a 4

4.1.1.4 float SERIALPORTCONFIG::FramePeriodUS

Período em microsegundos correspondente a um frame

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

- seriallibRTAI.h

Chapter 5

seriallibRTAI File Documentation

5.1 seriallibRTAI.c File Reference

Arquivo com as funções da biblioteca seriallibRTAI.

```
#include <stdio.h>
#include <math.h>
#include <unistd.h>
#include <sys/io.h>
#include <rtai_lxrt.h>
#include <rtai_sem.h>
#include <rtai_sched.h>
#include "seriallibRTAI.h"
```

Defines

- `#define SERIALLIBRTAI_COMPORTADDRESS_1 0x3F8`
- `#define SERIALLIBRTAI_COMPORTADDRESS_2 0x2F8`
- `#define SERIALLIBRTAI_COMPORTADDRESS_3 0x3E8`
- `#define SERIALLIBRTAI_COMPORTADDRESS_4 0x2E8`

Functions

- `void seriallibRTAI_delay (RTIME MaximaEsperaUS)`
- `int seriallibRTAI_init (P SERIALPORTCONFIG pSerialPortConfig, int ComPortNumber, unsigned long int ComPortBPS)`
- `void seriallibRTAI_semwait (P SERIALPORTCONFIG pSerialPortConfig)`
- `void seriallibRTAI_semsignal (P SERIALPORTCONFIG pSerialPortConfig)`
- `int seriallibRTAI_sendbyte (P SERIALPORTCONFIG pSerialPortConfig, unsigned char *pData)`
- `int seriallibRTAI_receivebyte (P SERIALPORTCONFIG pSerialPortConfig, unsigned char *pData, double MaximaEsperaUS)`
- `int seriallibRTAI_status (P SERIALPORTCONFIG pSerialPortConfig)`

Variables

- SEM * pComPortSemaphores [4]

5.1.1 Detailed Description

Arquivo com as funções da biblioteca seriallibRTAI.

5.1.2 Define Documentation

5.1.2.1 `#define SERIALLIBRTAI_COMPORTADDRESS_1 0x3F8`

Endereço base da porta serial COM1. Uso interno.

5.1.2.2 `#define SERIALLIBRTAI_COMPORTADDRESS_2 0x2F8`

Endereço base da porta serial COM2. Uso interno.

5.1.2.3 `#define SERIALLIBRTAI_COMPORTADDRESS_3 0x3E8`

Endereço base da porta serial COM3. Uso interno.

5.1.2.4 `#define SERIALLIBRTAI_COMPORTADDRESS_4 0x2E8`

Endereço base da porta serial COM4. Uso interno.

5.1.3 Function Documentation

5.1.3.1 `void seriallibRTAI_delay (RTIME MaximaEsperaUS)`

Função de uso interno da biblioteca, que provê um atraso em microsegundos dado por *MaximaEsperaUS*. Essa função bloqueia a thread.

Parameters:

MaximaEsperaUS atraso em unidade de microsegundos

5.1.3.2 `int seriallibRTAI_init (P SERIALPORTCONFIG pSerialPortConfig, int ComPortNumber, unsigned long int ComPortBPS)`

Função que inicia a porta serial *ComPortNumber* com a taxa dada em BPS por *ComPortBPS*. Essa função deverá ser chamada por cada thread que tenha acesso à porta serial *ComPortNumber*. Seus argumentos de chamada são:

Parameters:

pSerialPortConfig Ponteiro para estrutura SERIALPORCONFIG que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura SERIALPORCONFIG.

ComPortNumber Numero da porta serial, no intervalo de 1 a 4.

ComPortBPS Taxa de comunicação em BPS, no intervalo de 2 a 115200.

Returns:

SERIALLIBRTAI_SUCCESS : Porta iniciada com sucesso.

SERIALLIBRTAI_ERROR_INCORRECTPORTNUMBER : Erro, corresponde a um ComPortNumber inválido.

SERIALLIBRTAI_ERROR_MAXBPSPRECISION : Erro, a taxa ComPortBPS não pode ser realizada com erro inferior a. SERIALLIBRTAI_MAXBPSPRECISION.

SERIALLIBRTAI_ERROR_IOPL : Erro, corresponde a uma tentativa de executar o programa sem que se tenha acesso privilegiado de administrador a portas de E/S.

5.1.3.3 seriallibRTAI_receivebyte (P SERIALPORTCONFIG *pSerialPortConfig*, unsigned char * *pData*, double *MaximaEsperaUS*)

Função que aguarda um byte chegar pela porta serial descrita por pSerialPortConfig por um tempo máximo dado por MaximaEsperaUS, dado em microsegundos. Se um dado chegar dentro do período dado por MaximaEsperaUS, o mesmo será colocado na variável apontada por pData.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada.

pData Ponteiro para o byte recebido.

MaximaEsperaUS Tempo máximo de espera pela chegada de um byte pela porta. Se MaximaEsperaUS

Returns:

SERIALLIBRTAI_SUCCESS : Operação realizada com sucesso. Um byte foi recebido pela porta serial e se encontra disponível na variável apontada por pData.

SERIALLIBRTAI_ERROR_MAXWAITFORRECEPTION : Nenhum byte chegou dentro do tempo estipulado por MaximaEsperaUS

Warning:

Essa função fica bloqueada por até MaximaEsperaUS enquanto um byte não chegar.

5.1.3.4 void seriallibRTAI_semwait (P SERIALPORTCONFIG *pSerialPortConfig*)

Função que libera semáforo que foi previamente cedido por seriallibRTAI_semwait para acessar a porta descrita por pSerialPortConfig.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7)..

Warning:

Se uma determinada porta somente é gerenciada por um só thread, não há necessidade de se usar essas funções de semáforo. As funções de semáforo têm somente utilidade em situações em que mais de um thread pode acessar a porta serial X, com X = 1, 2, 3 ou 4. Após concluir o acesso à porta serial cedido por essa função, deve-se chamar `seriallibRTAI_semsignal` para liberar o semáforo

5.1.3.5 void seriallibRTAI_semwait (PSERIALPORTCONFIG *pSerialPortConfig*)

Função que aguarda semáforo para acessar a porta descrita por `pSerialPortConfig`. Juntamente com `seriallibRTAI_semsignal`, pode-se garantir o acesso exclusivo de um thread à porta serial.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7)..

Warning:

Se uma determinada porta somente é gerenciada por um só thread, não há necessidade de se usar essas funções de semáforo. As funções de semáforo têm somente utilidade em situações em que mais de um thread pode acessar a porta serial X, com X = 1, 2, 3 ou 4. Após concluir o acesso à porta serial cedido por essa função, deve-se chamar `seriallibRTAI_semsignal` para liberar o semáforo

5.1.3.6 int seriallibRTAI_sendbyte (PSERIALPORTCONFIG *pSerialPortConfig*, unsigned char * *pData*)

Função que envia um byte apontado por `pData` pela porta serial descrita por `pSerialPortConfig`.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7). Se `SERIALLIBRTAI_USE_RS485 = 1`, então o sinal RTS será colocado em nível lógico 1 enquanto durar o frame do byte enviado, permitindo assim ativar o driver externo de uma porta com conversor RS-485. Nessa situação, essa função somente retorna quando o byte tiver sido enviado. Caso contrário, a função somente escreverá no buffer de saída o byte apontado por `pData`, retornando em * seguida.

pData Ponteiro para o byte que será enviado.

Returns:

SERIALLIBRTAI_SUCCESS : Dado escrito no registro de saída com sucesso. Entretanto, isso significa apenas que uma transmissão está em curso. Para se certificar de que o dado foi efetivamente transmitido, deve-se fazer uso da função **seriallibRTAI_status()** (p. 13)

SERIALLIBRTAI_ERROR_MAXWAITENDOFTRANSMISSION : Situação de erro em que a função ficou aguardando por um período de até 5 frames para disponibilização do registro de saída da porta

Warning:

Essa função fica bloqueada enquanto o último byte escrito no buffer de saída ainda não tiver sido enviado.

5.1.3.7 int seriallibRTAI_status (P SERIALPORTCONFIG *pSerialPortConfig*)

Função que lê o registro de status da porta serial descrita por *pSerialPortConfig*.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada.

Returns:

O valor de retorno tem os bits setados conforme que o dado foi efetivamente enviadoos eventos que ocorreram com a porta serial, que podem ser testados usando um teste lógico E bit a bit com as seguintes máscaras:

```
SERIALLIBRTAI_STATUSMASK_ERROR_RX_FIFO
SERIALLIBRTAI_STATUSMASK_EMPTY_DH_REGISTERS
SERIALLIBRTAI_STATUSMASK_EMPTY_TX_REGISTER
SERIALLIBRTAI_STATUSMASK_BREAK_INTERRUPT
SERIALLIBRTAI_STATUSMASK_FRAMING_ERROR
SERIALLIBRTAI_STATUSMASK_PARITY_ERROR
SERIALLIBRTAI_STATUSMASK_OVERRUN_ERROR
SERIALLIBRTAI_STATUSMASK_RX_DATA_READY
```

As máscaras acima correspondem a eventos que são detalhados em <http://www.beyondlogic.org/serial/serial.htm>

5.1.4 Variable Documentation**5.1.4.1 SEM * pComPortSemaphores[4]**

Vetor de ponteiros para semáforos. Cada elemento desse vetor é um ponteiro para o semáforo associado à porta X, com X = 1, 2, 3 ou 4. Os semáforos de cada porta são iniciados na chamada à função **seriallibRTAI_init()** (p. 10). Para uso interno pelas funções **seriallibRTAI_semwait()** (p. 12) e **seriallibRTAI_semsignal()** (p. 11).

5.2 seriallibRTAI.h File Reference

Arquivo cabeçalho da biblioteca seriallibRTAI.

Data Structures

- struct **SERIALPORTCONFIG**

Defines

- #define **SERIALLIBRTAI_USE_RS485** 1
- #define **SERIALLIBRTAI_MAXBPSPRECISION** 0.02
- #define **SERIALLIBRTAI_SUCCESS** 0
- #define **SERIALLIBRTAI_ERROR_IOPL** 1
- #define **SERIALLIBRTAI_ERROR_MAXWAITENDOFTRANSMISSION** 2
- #define **SERIALLIBRTAI_ERROR_MAXWAITFORRECEPTION** 3
- #define **SERIALLIBRTAI_ERROR_MAXBPSPRECISION** 4
- #define **SERIALLIBRTAI_ERROR_INCORRECTPORTNUMBER** 5
- #define **SERIALLIBRTAI_STATUSMASK_ERROR_RX_FIFO** 0x80
- #define **SERIALLIBRTAI_STATUSMASK_EMPTY_DH_REGISTERS** 0x40
- #define **SERIALLIBRTAI_STATUSMASK_EMPTY_TX_REGISTER** 0x20
- #define **SERIALLIBRTAI_STATUSMASK_BREAK_INTERRUPT** 0x10
- #define **SERIALLIBRTAI_STATUSMASK_FRAMING_ERROR** 0x08
- #define **SERIALLIBRTAI_STATUSMASK_PARITY_ERROR** 0x04
- #define **SERIALLIBRTAI_STATUSMASK_OVERRUN_ERROR** 0x02
- #define **SERIALLIBRTAI_STATUSMASK_RX_DATA_READY** 0x01

Typedefs

- typedef struct **SERIALPORTCONFIG** * **PSERIALPORTCONFIG**

Functions

- int seriallibRTAI_init (PSERIALPORTCONFIG pSerialPortConfig, int ComPortNumber, unsigned long int ComPortBPS)
- int seriallibRTAI_sendbyte (PSERIALPORTCONFIG pSerialPortConfig, unsigned char *pData)
- int seriallibRTAI_receivebyte (PSERIALPORTCONFIG pSerialPortConfig, unsigned char *pData, double MaximaEsperaUS)
- int seriallibRTAI_status (PSERIALPORTCONFIG pSerialPortConfig)
- void seriallibRTAI_semwait (PSERIALPORTCONFIG pSerialPortConfig)
- void seriallibRTAI_semsignal (PSERIALPORTCONFIG pSerialPortConfig)

5.2.1 Detailed Description

Arquivo cabeçalho da biblioteca seriallibRTAI.

5.2.2 Define Documentation

5.2.2.1 `#define SERIALLIBRTAI_ERROR_INCORRECTPORTNUMBER 5`

5.2.2.2 `#define SERIALLIBRTAI_ERROR_IOPL 1`

5.2.2.3 `#define SERIALLIBRTAI_ERROR_MAXBPSPRECISION 4`

5.2.2.4 `#define SERIALLIBRTAI_ERROR_MAXWAITENDOFTRANSMISSION 2`

5.2.2.5 `#define SERIALLIBRTAI_ERROR_MAXWAITFORRECEPTION 3`

5.2.2.6 `#define SERIALLIBRTAI_MAXBPSPRECISION 0.02`

Erro relativo maximo aceitavel para definir taxa de transmissão. PadrSe SERIALLIBRTAI_USE_RS485 esetiver em 1, a linha RTS será colocada em nível lógico 1 durante a trnamissão de cada frame de um byte. Em geral, isso é usado quando se desejar utilizar um driver RS485 com sinal de controle pelo pino RTSão: 0.02, que corresponde a 2% de erro.

5.2.2.7 `#define SERIALLIBRTAI_STATUSMASK_BREAK_INTERRUPT 0x10`

5.2.2.8 `#define SERIALLIBRTAI_STATUSMASK_EMPTY_DH_REGISTERS 0x40`

5.2.2.9 `#define SERIALLIBRTAI_STATUSMASK_EMPTY_TX_REGISTER 0x20`

5.2.2.10 `#define SERIALLIBRTAI_STATUSMASK_ERROR_RX_FIFO 0x80`

5.2.2.11 `#define SERIALLIBRTAI_STATUSMASK_FRAMING_ERROR 0x08`

5.2.2.12 `#define SERIALLIBRTAI_STATUSMASK_OVERRUN_ERROR 0x02`

5.2.2.13 `#define SERIALLIBRTAI_STATUSMASK_PARITY_ERROR 0x04`

5.2.2.14 `#define SERIALLIBRTAI_STATUSMASK_RX_DATA_READY 0x01`

5.2.2.15 `#define SERIALLIBRTAI_SUCCESS 0`

5.2.2.16 `#define SERIALLIBRTAI_USE_RS485 1`

Se SERIALLIBRTAI_USE_RS485 esetiver em 1, a linha RTS será colocada em nível lógico 1 durante a trnamissão de cada frame de um byte. Em geral, isso é usado quando se desejar utilizar um driver RS485 com sinal de controle pelo pino RTS.

5.2.3 Typedef Documentation

5.2.3.1 typedef struct SERIALPORTCONFIG * PSerialPortConfig

5.2.4 Function Documentation

5.2.4.1 int seriallibRTAI_init (PSerialPortConfig *pSerialPortConfig*, int *ComPortNumber*, unsigned long int *ComPortBPS*)

Função que inicia a porta serial *ComPortNumber* com a taxa dada em BPS por *ComPortBPS*. Essa função deverá ser chamada por cada thread que tenha acesso à porta serial *ComPortNumber*. Seus argumentos de chamada são:

Parameters:

pSerialPortConfig Ponteiro para estrutura SERIALPORCONFIG que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura SERIALPORCONFIG.

ComPortNumber Numero da porta serial, no intervalo de 1 a 4.

ComPortBPS Taxa de comunicação em BPS, no intervalo de 2 a 115200.

Returns:

SERIALLIBRTAI_SUCCESS : Porta iniciada com sucesso.

SERIALLIBRTAI_ERROR_INCORRECTPORTNUMBER : Erro, corresponde a um ComPortNumber inválido.

SERIALLIBRTAI_ERROR_MAXBPSPRECISION : Erro, a taxa ComPortBPS não pode ser realizada com erro inferior a. SERIALLIBRTAI_MAXBPSPRECISION.

SERIALLIBRTAI_ERROR_IOPL : Erro, corresponde a uma tentativa de executar o programa sem que se tenha acesso privilegiado de administrador a portas de E/S.

5.2.4.2 int seriallibRTAI_receivebyte (PSerialPortConfig *pSerialPortConfig*, unsigned char * *pData*, double *MaximaEsperaUS*)

Função que aguarda um byte chegar pela porta serial descrita por *pSerialPortConfig* por um tempo máximo dado por *MaximaEsperaUS*, dado em microsegundos. Se um dado chegar dentro do período dado por *MaximaEsperaUS*, o mesmo será colocado na variável apontada por *pData*.

Parameters:

pSerialPortConfig Ponteiro para estrutura SERIALPORTCONFIG (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada.

pData Ponteiro para o byte recebido.

MaximaEsperaUS Tempo máximo de espera pela chegada de um byte pela porta. Se *MaximaEsperaUS*

Returns:

SERIALLIBRTAI_SUCCESS : Operação realizada com sucesso. Um byte foi recebido pela porta serial e se encontra disponível na variável apontada por *pData*.

SERIALLIBRTAI_ERROR_MAXWAITFORRECEPTION : Nenhum bayte chegou dentro do tempo estipulado por *MaximaEsperaUS*

Warning:

Essa função fica bloqueada por até `MaximaEsperaUS` enquanto um byte não chegar.

5.2.4.3 void seriallibRTAI_semsignal (PSERIALPORTCONFIG *pSerialPortConfig*)

Função que libera semáforo que foi previamente cedido por `seriallibRTAI_semwait` para acessar a porta descrita por `pSerialPortConfig`.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7)..

Warning:

Se uma determinada porta somente é gerenciada por um só thread, não há necessidade de se usar essas funções de semáforo. As funções de semáforo têm somente utilidade em situações em que mais de um thread pode acessar a porta serial X, com $X = 1, 2, 3$ ou 4. Após concluir o acesso à porta serial cedido por essa função, deve-se chamar `seriallibRTAI_semsignal` para liberar o semáforo

5.2.4.4 void seriallibRTAI_semwait (PSERIALPORTCONFIG *pSerialPortConfig*)

Função que aguarda semáforo para acessar a porta descrita por `pSerialPortConfig`. Juntamente com `seriallibRTAI_semsignal`, pode-se garantir o acesso exclusivo de um thread à porta serial.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7)..

Warning:

Se uma determinada porta somente é gerenciada por um só thread, não há necessidade de se usar essas funções de semáforo. As funções de semáforo têm somente utilidade em situações em que mais de um thread pode acessar a porta serial X, com $X = 1, 2, 3$ ou 4. Após concluir o acesso à porta serial cedido por essa função, deve-se chamar `seriallibRTAI_semsignal` para liberar o semáforo

5.2.4.5 int seriallibRTAI_sendbyte (PSERIALPORTCONFIG *pSerialPortConfig*, unsigned char * *pData*)

Função que envia um byte apontado por `pData` pela porta serial descrita por `pSerialPortConfig`.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada. Mesmo que uma dada porta serial seja utilizada por diversos threads, cada thread deverá ter a sua estrutura **SERIALPORTCONFIG** (p. 7). Se **SERIALLIBRTAI_USE_RS485** = 1, então o sinal RTS será colocado em nível lógico 1 enquanto durar o frame do byte enviado, permitindo assim ativar o driver externo de uma porta com conversor RS-485. Nessa situação, essa função somente retorna quando o byte tiver sido enviado. Caso contrário, a função somente escreverá no buffer de saída o byte apontado por **pData**, retornando em * seguida.

pData Ponteiro para o byte que será enviado.

Returns:

SERIALLIBRTAI_SUCCESS : Dado escrito no registro de saída com sucesso. Entretanto, isso significa apenas que uma transmissão está em curso. Para se certificar de que o dado foi efetivamente transmitido, deve-se fazer uso da função **seriallibRTAI_status()** (p. 13)

SERIALLIBRTAI_ERROR_MAXWAITENDOFTRANSMISSION : Situação de erro em que a função ficou aguardando por um período de até 5 frames para disponibilização do registro de saída da porta

Warning:

Essa função fica bloqueada enquanto o último byte escrito no buffer de saída ainda não tiver sido enviado.

5.2.4.6 int seriallibRTAI_status (PSERIALPORTCONFIG *pSerialPortConfig*)

Função que lê o registro de status da porta serial descrita por **pSerialPortConfig**.

Parameters:

pSerialPortConfig Ponteiro para estrutura **SERIALPORTCONFIG** (p. 7) que guarda informações de configuração da porta serial no contexto do thread de chamada.

Returns:

O valor de retorno tem os bits setados conforme que o dado foi efetivamente enviados eventos que ocorreram com a porta serial, que podem ser testados usando um teste lógico E bit a bit com as seguintes máscaras:

SERIALLIBRTAI_STATUSMASK_ERROR_RX_FIFO
SERIALLIBRTAI_STATUSMASK_EMPTY_DH_REGISTERS
SERIALLIBRTAI_STATUSMASK_EMPTY_TX_REGISTER
SERIALLIBRTAI_STATUSMASK_BREAK_INTERRUPT
SERIALLIBRTAI_STATUSMASK_FRAMING_ERROR
SERIALLIBRTAI_STATUSMASK_PARITY_ERROR
SERIALLIBRTAI_STATUSMASK_OVERRUN_ERROR
SERIALLIBRTAI_STATUSMASK_RX_DATA_READY

As máscaras acima correspondem a eventos que são detalhados em <http://www.beyondlogic.org/serial/serial.htm>

Index

ComPortAddress
 SERIALPORTCONFIG, 7
ComPortBPS
 SERIALPORTCONFIG, 7
ComPortNumber
 SERIALPORTCONFIG, 7

FramePeriodUS
 SERIALPORTCONFIG, 7

pComPortSemaphores
 seriallibRTAI.c, 13
PSERIALPORTCONFIG
 seriallibRTAI.h, 16

seriallibRTAI.c, 9
seriallibRTAI.c
 pComPortSemaphores, 13
 SERIALLIBRTAI_-
 COMPORTADDRESS_1, 10
 SERIALLIBRTAI_-
 COMPORTADDRESS_2, 10
 SERIALLIBRTAI_-
 COMPORTADDRESS_3, 10
 SERIALLIBRTAI_-
 COMPORTADDRESS_4, 10
 seriallibRTAI_delay, 10
 seriallibRTAI_init, 10
 seriallibRTAI_receivebyte, 11
 seriallibRTAI_semsignal, 11
 seriallibRTAI_semwait, 12
 seriallibRTAI_sendbyte, 12
 seriallibRTAI_status, 13
seriallibRTAI.h, 14
seriallibRTAI.h
 PSERIALPORTCONFIG, 16
 SERIALLIBRTAI_ERROR_-
 INCORRECTPORTNUMBER,
 15
 SERIALLIBRTAI_ERROR_IOPL, 15
 SERIALLIBRTAI_ERROR_-
 MAXBPSPRECISION, 15
 SERIALLIBRTAI_ERROR_-
 MAXWAITENDOFTRANSMISSION,
 15
 SERIALLIBRTAI_ERROR_-
 MAXWAITFORRECEPTION, 15
 seriallibRTAI_init, 16
 SERIALLIBRTAI_-
 MAXBPSPRECISION, 15
 seriallibRTAI_receivebyte, 16
 seriallibRTAI_semsignal, 17
 seriallibRTAI_semwait, 17
 seriallibRTAI_sendbyte, 17
 seriallibRTAI_status, 18
 SERIALLIBRTAI_STATUSMASK_-
 BREAK_INTERRUPT, 15
 SERIALLIBRTAI_STATUSMASK_-
 EMPTY_DH_REGISTERS, 15
 SERIALLIBRTAI_STATUSMASK_-
 EMPTY_TX_REGISTER, 15
 SERIALLIBRTAI_STATUSMASK_-
 ERROR_RX_FIFO, 15
 SERIALLIBRTAI_STATUSMASK_-
 FRAMING_ERROR, 15
 SERIALLIBRTAI_STATUSMASK_-
 OVERRUN_ERROR, 15
 SERIALLIBRTAI_STATUSMASK_-
 PARITY_ERROR, 15
 SERIALLIBRTAI_STATUSMASK_-
 RX_DATA_READY, 15
 SERIALLIBRTAI_SUCCESS, 15
 SERIALLIBRTAI_USE_RS485, 15
SERIALLIBRTAI_COMPORTADDRESS_1
 seriallibRTAI.c, 10
SERIALLIBRTAI_COMPORTADDRESS_2
 seriallibRTAI.c, 10
SERIALLIBRTAI_COMPORTADDRESS_3
 seriallibRTAI.c, 10
SERIALLIBRTAI_COMPORTADDRESS_4
 seriallibRTAI.c, 10
seriallibRTAI_delay
 seriallibRTAI.c, 10
SERIALLIBRTAI_ERROR_-
 INCORRECTPORTNUMBER
 seriallibRTAI.h, 15
SERIALLIBRTAI_ERROR_IOPL
 seriallibRTAI.h, 15
SERIALLIBRTAI_ERROR_-
 MAXBPSPRECISION

seriallibRTAI.h, 15	seriallibRTAI.h, 15
SERIALLIBRTAI_ERROR_-	SERIALPORTCONFIG, 7
MAXWAITENDOFTTRANSMISSION	ComPortAddress, 7
seriallibRTAI.h, 15	ComPortBPS, 7
SERIALLIBRTAI_ERROR_-	ComPortNumber, 7
MAXWAITFORRECEPTION	FramePeriodUS, 7
seriallibRTAI.h, 15	
seriallibRTAI_init	
seriallibRTAI.c, 10	
seriallibRTAI.h, 16	
SERIALLIBRTAI_MAXBPSPRECISION	
seriallibRTAI.h, 15	
seriallibRTAI_receivebyte	
seriallibRTAI.c, 11	
seriallibRTAI.h, 16	
seriallibRTAI_sensignal	
seriallibRTAI.c, 11	
seriallibRTAI.h, 17	
seriallibRTAI_semwait	
seriallibRTAI.c, 12	
seriallibRTAI.h, 17	
seriallibRTAI_sendbyte	
seriallibRTAI.c, 12	
seriallibRTAI.h, 17	
seriallibRTAI_status	
seriallibRTAI.c, 13	
seriallibRTAI.h, 18	
SERIALLIBRTAI_STATUSMASK_-	
BREAK_INTERRUPT	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
EMPTY_DH_REGISTERS	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
EMPTY_TX_REGISTER	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
ERROR_RX_FIFO	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
FRAMING_ERROR	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
OVERRUN_ERROR	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_-	
PARITY_ERROR	
seriallibRTAI.h, 15	
SERIALLIBRTAI_STATUSMASK_RX_-	
DATA_READY	
seriallibRTAI.h, 15	
SERIALLIBRTAI_SUCCESS	
seriallibRTAI.h, 15	
SERIALLIBRTAI_USE_RS485	
