Biblioteca para o TWI (I2C)

Funções

|  |  |  |
| --- | --- | --- |
| byte | twi\_er\_ok | (byte adr) |
| void | twi\_scan | (void) |
| void | twi\_config\_400k | (void) |
| void | twi\_config\_100k | (void) |
| void | void twi\_start | (byte ix) |
| void | void twi\_start\_rep | (int ix) |
| void | twi\_stop | (void) |
| byte | twi\_er\_check | (byte ee, byte ix) |
| void | twi\_er | (byte eer, byte ix) |
| void | twi\_et | (byte eet, byte ix) |
| void | twi\_dado\_er | (byte dado, byte ix) |
| byte | twi\_dado\_et\_ack | (byte ix) |
| byte | byte twi\_dado\_et\_nack | (byte ix) |
| byte | byte espera\_twint | (byte ix) |
| void | twi\_erro | (int cod, int ix) |
|  |  |  |
|  |  |  |
|  |  |  |

* byte **twi\_er\_ok** (byte adr)

Verificar se tem resposta do escravo receptor no endereço “adr”.

TRUE se escravo respondeu com ACK.

FALSE se escravo respondeu com NACK.

* void **twi\_scan** (void)

Buscar por todos os dispositivos TWI para escrita. Imprime na serial o que encontrar.

Como busca por escrita, faz adr<<1 = bbbb bbb0.

* void **twi\_config\_400k** (void)

Configurar TWI para 400k (frequência da linha SCL).

* void **twi\_config\_100k** (void)

Configurar TWI para 100k (frequência da linha SCL).

* void **twi\_start** (byte ix)

Gerar um START no TWI.

* void **twi\_start\_rep** (int ix)

Gerar um START Repetido no TWI.

* void **twi\_stop** (void)

Enviar STOP.

* byte **twi\_er\_check** (byte ee, byte ix)

Enviar endereço de Escrita do Escravo Receptor (ER) e esperar ACK.

Retorna TRUE se escravo gerou ACK.

Retorna FALSE se gerou NACK (não gera msg de erro).

ee = endereço do escravo.

* void **twi\_er** (byte eer, byte ix)

Enviar endereço de Escrita do Escravo (ER) e esperar ACK.

eer = endereço do escravo receptor.

* void **twi\_et** (byte eet, byte ix)

Enviar endereço de Leitura do Escravo (ET) e esperar ACK.

eet = endereço do escravo transmissor.

* void **twi\_dado\_er** (byte dado, byte ix)

Enviar dado para escravo previamente endereçado.

* byte **twi\_dado\_et\_ack** (byte ix)

Receber dado e gerar ACK.

* byte **twi\_dado\_et\_nack** (byte ix)

Receber dado e gerar NACK.

* byte **espera\_twint** (byte ix)

Esperar TWINT.

* void **twi\_erro** (int cod, int ix)

ERROS no TWI (I2C).

1 = Erro ao gerar START.

2 = Erro ao gerar START Repetido.

3 = Erro Escravo Receptor endereçado (ER) não enviou ACK.

4 = Erro Escravo Transmissor endereçado (ET) não enviou ACK.

5 = Erro Escravo Receptor (ER) não enviou ACK após envio do dado.

6 = Erro ao receber um dado do Escravo Transmissor (ET) e gerar um ACK.

7 = Erro ao receber um dado do Escravo Transmissor (ET) e gerar um NACK.

8 = Erro ao esperar TWINT - Timeout esperando TWINT ir para 1.

*Gabarito para configurar os registradores do TWI*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **TWCR** | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |
| **TWSR** | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | - | TWPS1 | TWPS0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **TWBR** | TWBR7 | TWBR6 | TWBR5 | TWBR4 | TWBR3 | TWBR2 | TWBR1 | TWBR0 |
| - | - | - | - | - | - | - | - |
| **TWDR** | TWD7 | TWD6 | TWD5 | TWD4 | TWD3 | TWD2 | TWD1 | TWD0 |
| - | - | - | - | - | - | - | - |
| **TWAR** | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE |
| - | - | - | - | - | - | - | - |
| **TWAMR** | TWAM6 | TWAM5 | TWAM4 | TWAM3 | TWAM2 | TWAM1 | TWAM0 | - |
| - | - | - | - | - | - | - | - |