

Driver - PIO e PMC

Rafael Corsi

Generated by Doxygen 1.8.6

Wed Mar 23 2016 14:24:22



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	<a href="#">/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/pio_maua.h File Reference</a> . .	3
2.1.1	Detailed Description . . . . .	4
2.1.2	Macro Definition Documentation . . . . .	4
2.1.2.1	PIO_DEFAULT . . . . .	4
2.1.3	Function Documentation . . . . .	4
2.1.3.1	<a href="#">_pio_clear</a> . . . . .	4
2.1.3.2	<a href="#">_pio_get_output_data_status</a> . . . . .	4
2.1.3.3	<a href="#">_pio_pull_down</a> . . . . .	4
2.1.3.4	<a href="#">_pio_pull_up</a> . . . . .	5
2.1.3.5	<a href="#">_pio_set</a> . . . . .	5
2.1.3.6	<a href="#">_pio_set_input</a> . . . . .	5
2.1.3.7	<a href="#">_pio_set_output</a> . . . . .	5
2.2	<a href="#">/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/pmc_maua.h File Reference</a> . .	5
2.2.1	Detailed Description . . . . .	6
2.2.2	Function Documentation . . . . .	6
2.2.2.1	<a href="#">_pmc_disable_clock_periferico</a> . . . . .	6
2.2.2.2	<a href="#">_pmc_enable_clock_periferico</a> . . . . .	6
	<b>Index</b>	<b>8</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all documented files with brief descriptions:

/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/ <a href="#">pio_maua.h</a>	
Funções para configurar o PIO do SAM4S . . . . .	3
/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/ <a href="#">pmc_maua.h</a>	
Funções para configurar o PMC do SAM4S . . . . .	5



## Chapter 2

# File Documentation

### 2.1 /home/corsi/Documentos/EEN251/Codigos/08-DRIVER\_PIO\_PCM/pio\_maua.h File Reference

Funções para configurar o PIO do SAM4S.

#### Macros

- #define **PIO\_DEFAULT** (0u << 0)
- #define **PIO\_PULLUP** (1u << 0)
- #define **PIO\_DEGLITCH** (1u << 1)
- #define **PIO\_OPENDRAIN** (1u << 2)
- #define **PIO\_DEBOUNCE** (1u << 3)

#### Functions

- void **\_pio\_set\_input** (Pio \*p\_pio, const uint32\_t ul\_mask, const uint32\_t ul\_attribute)  
*Configure one or more pin(s) or a PIO controller as inputs. Optionally, the corresponding internal pull-up(s) and glitch filter(s) can be enabled.*
- void **\_pio\_set\_output** (Pio \*p\_pio, const uint32\_t ul\_mask, const uint32\_t ul\_default\_level, const uint32\_t ul\_pull\_up\_enable)  
*Configure one or more pin(s) of a PIO controller as outputs, with the given default value.*
- void **\_pio\_pull\_up** (Pio \*p\_pio, const uint32\_t ul\_mask, const uint32\_t ul\_pull\_up\_enable)  
*Configure PIO internal pull-up.*
- void **\_pio\_pull\_down** (Pio \*p\_pio, const uint32\_t ul\_mask, const uint32\_t ul\_pull\_down\_enable)  
*Configure PIO pin internal pull-down.*
- void **\_pio\_set** (Pio \*p\_pio, const uint32\_t ul\_mask)  
*Set a high output level on all the PIOs defined in ul\_mask. This has no immediate effects on PIOs that are not output, but the PIO controller will save the value if they are changed to outputs.*
- void **\_pio\_clear** (Pio \*p\_pio, const uint32\_t ul\_mask)  
*Set a low output level on all the PIOs defined in ul\_mask. This has no immediate effects on PIOs that are not output, but the PIO controller will save the value if they are changed to outputs.*
- uint32\_t **\_pio\_get\_output\_data\_status** (const Pio \*p\_pio, const uint32\_t ul\_mask)  
*Return 1 if one or more PIOs of the given Pin are configured to output a high level (even if they are not output). To get the actual value of the pin, use PIO\_Get() instead.*

### 2.1.1 Detailed Description

Funções para configurar o PIO do SAM4S.

#### Author

Rafael Corsi

#### Date

22/3/2016

### 2.1.2 Macro Definition Documentation

#### 2.1.2.1 #define PIO\_DEFAULT (0u << 0)

Esses defines serão utilizados para configurar o modo de operação dos I/Os, quando em modo de entrada

### 2.1.3 Function Documentation

#### 2.1.3.1 void \_pio\_clear ( Pio \* *p\_pio*, const uint32\_t *ul\_mask* )

Set a low output level on all the PIOs defined in *ul\_mask*. This has no immediate effects on PIOs that are not output, but the PIO controller will save the value if they are changed to outputs.

#### Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask of one or more pin(s) to configure.

#### 2.1.3.2 uint32\_t \_pio\_get\_output\_data\_status ( const Pio \* *p\_pio*, const uint32\_t *ul\_mask* )

Return 1 if one or more PIOs of the given Pin are configured to output a high level (even if they are not output). To get the actual value of the pin, use `PIO_Get()` instead.

#### Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask of one or more pin(s).

#### Return values

1	At least one PIO is configured to output a high level.
0	All PIOs are configured to output a low level.

#### 2.1.3.3 void \_pio\_pull\_down ( Pio \* *p\_pio*, const uint32\_t *ul\_mask*, const uint32\_t *ul\_pull\_down\_enable* )

Configure PIO pin internal pull-down.

#### Parameters

<i>p_pio</i>	Pointer to a PIO instance.
--------------	----------------------------



<i>ul_mask</i>	Bitmask of one or more pin(s) to configure.
<i>ul_pull_down_enable</i>	Indicates if the pin(s) internal pull-down shall be configured.

2.1.3.4 void \_pio\_pull\_up ( Pio \* *p\_pio*, const uint32\_t *ul\_mask*, const uint32\_t *ul\_pull\_up\_enable* )

Configure PIO internal pull-up.

Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask of one or more pin(s) to configure.
<i>ul_pull_up_enable</i>	Indicates if the pin(s) internal pull-up shall be configured.

2.1.3.5 void \_pio\_set ( Pio \* *p\_pio*, const uint32\_t *ul\_mask* )

Set a high output level on all the PIOs defined in *ul\_mask*. This has no immediate effects on PIOs that are not output, but the PIO controller will save the value if they are changed to outputs.

Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask of one or more pin(s) to configure.

2.1.3.6 void \_pio\_set\_input ( Pio \* *p\_pio*, const uint32\_t *ul\_mask*, const uint32\_t *ul\_attribute* )

Configure one or more pin(s) or a PIO controller as inputs. Optionally, the corresponding internal pull-up(s) and glitch filter(s) can be enabled.

Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask indicating which pin(s) to configure as input(s).
<i>ul_attribute</i>	PIO attribute(s).

2.1.3.7 void \_pio\_set\_output ( Pio \* *p\_pio*, const uint32\_t *ul\_mask*, const uint32\_t *ul\_default\_level*, const uint32\_t *ul\_pull\_up\_enable* )

Configure one or more pin(s) of a PIO controller as outputs, with the given default value.

Parameters

<i>p_pio</i>	Pointer to a PIO instance.
<i>ul_mask</i>	Bitmask indicating which pin(s) to configure.
<i>ul_default_level</i>	Default level on the pin(s).
<i>ul_pull_up_enable</i>	Indicates if the pin shall have its pull-up activated.

## 2.2 /home/corsi/Documentos/EEN251/Codigos/08-DRIVER\_PIO\_PCM/pmc\_maua.h File Reference

Funções para configurar o PMC do SAM4S.

```
#include <stdint.h>
#include <sam4sd32c.h>
```

## Functions

- [uint32\\_t \\_pmc\\_enable\\_clock\\_periferico](#) (uint32\_t ID)  
*STDINT possui as definições dos tipos de variáveis e constantes.*
- [uint32\\_t \\_pmc\\_disable\\_clock\\_periferico](#) (uint32\_t ID)  
*Desativamos o clocks no periferico.*

### 2.2.1 Detailed Description

Funções para configurar o PMC do SAM4S.

#### Author

Rafael Corsi

#### Date

22/3/2016

### 2.2.2 Function Documentation

#### 2.2.2.1 uint32\_t \_pmc\_disable\_clock\_periferico ( uint32\_t ID )

Desativamos o clocks no periferico.

#### Parameters

ID	ID do periferico
----	------------------

#### Note

ID não deve ser shiftado, i.e : (1 << ID\_XXX).

#### Return values

0	se bem-sucedido
1	se malsucedido

Existem dois registradores que configuram os clocks dos perifericos, é necessário testarmos antes para saber em qual deve ser salvo.

O ID máximo para o PMC\_PCER0 é 32

qual seria a situação que retornariamos 1 ?

#### 2.2.2.2 uint32\_t \_pmc\_enable\_clock\_periferico ( uint32\_t ID )

STDINT possui as definições dos tipos de variáveis e constantes.

Inclui as definições prévias do uc em uso Ativa o clock no periferico

## Parameters

<i>ID</i>	ID do periferico
-----------	------------------

## Note

ID não deve ser shiftado, i.e :  $(1 \ll ID\_xxx)$ .

## Return values

0	se bem-sucedido
1	se malsucedido

inclui o head do pmc Existem dois registradores que configuram os clocks dos perifericos, é necessário testarmos antes para saber em qual deve ser salvo.

O ID máximo para o PMC\_PCER0 é 32

qual seria a situação que retornariamos 1 ?

# Index

- `_pio_clear`
  - `pio_maua.h`, [4](#)
- `_pio_get_output_data_status`
  - `pio_maua.h`, [4](#)
- `_pio_pull_down`
  - `pio_maua.h`, [4](#)
- `_pio_pull_up`
  - `pio_maua.h`, [5](#)
- `_pio_set`
  - `pio_maua.h`, [5](#)
- `_pio_set_input`
  - `pio_maua.h`, [5](#)
- `_pio_set_output`
  - `pio_maua.h`, [5](#)
- `_pmc_disable_clock_periferico`
  - `pmc_maua.h`, [6](#)
- `_pmc_enable_clock_periferico`
  - `pmc_maua.h`, [6](#)
- `PIO_DEFAULT`
  - `pio_maua.h`, [4](#)
- `pio_maua.h`
  - `_pio_clear`, [4](#)
  - `_pio_get_output_data_status`, [4](#)
  - `_pio_pull_down`, [4](#)
  - `_pio_pull_up`, [5](#)
  - `_pio_set`, [5](#)
  - `_pio_set_input`, [5](#)
  - `_pio_set_output`, [5](#)
  - `PIO_DEFAULT`, [4](#)
- `pmc_maua.h`
  - `_pmc_disable_clock_periferico`, [6](#)
  - `_pmc_enable_clock_periferico`, [6](#)