

PIO Driver

Rafael Corsi Ferrão
corsiferrao@gmail.com

31 de março de 2016

1 Driver

Dois drivers deverão ser projetados, um para controlar o periférico responsável pelo clock (PMC) e outro para comandar o periférico PIO.

Os drivers são normalmente compostos por arquivos ".c" e ".h". No ".c" estará a implementação das funções e nos arquivos ".h" os includes necessários, defines, variáveis e prototipagem das funções.

A estrutura de arquivos proposta é a seguinte:

```
src
├── main.c
├── Driver
│   ├── pio_maua.c
│   ├── pio_maua.h
│   ├── pmc_maua.c
│   └── pmc_maua.h
```

Devemos criar uma pasta "Driver" dentro do src do projeto : src/Driver.

Nessa pasta vamos adicionar 4 arquivos, sendo eles :

- pmc_maua.c : contém as funções que controlam o PMC
- pmc_maua.h : Header do driver que controla o PMC
- pio_maua.c : contém as funções que controlam o PIO

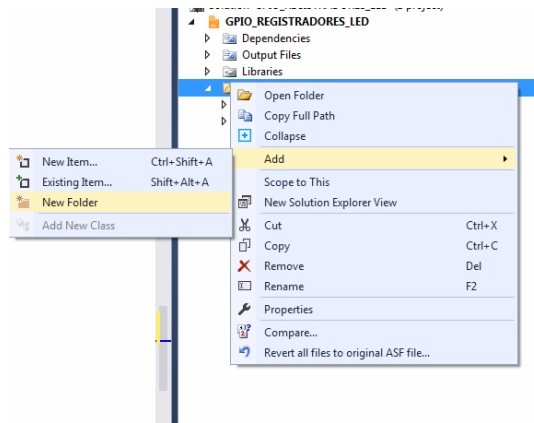


Figura 1

- pio_maua.h : Header do driver que controla o PIO

Esses arquivos estão localizados no repositório da disciplina, na pasta : *EEN251/Codigos/08-DRIVER_PIO_PCM*. Para adicionar basta clicarmos com o botão direito sobre a pasta "Driver" → Add → Existing Item.

Nessa mesma pasta há um documento que descreve cada uma das funções já definidas.

1.1 PMC

Duas funções são definidas na biblioteca pmc_maua:

```
uint32_t _pmc_enable_clock_periferico(uint32_t ID);
uint32_t _pmc_disable_clock_periferico(uint32_t ID);
```

Essas funções, uma vez incluído no cabeçalho do main.c podem ser utilizadas para configurar o PMC. Verifique como essas funções foram implementadas.

Exemplo de uso:

```
#include <asf.h>

/**
 * Inclui a biblioteca de configuracao do PMC
 */
#include "Driver/pmc_maua.h"

int main (void){

    .....

    /**
     * Ativa o clock do PMC A e C
     */
```

```
_pmc_enable_clock_periferico(ID_PIOA);  
_pmc_enable_clock_periferico(ID_PIOC);  
  
.....  
}
```

6 - Include

Qual a diferença entre usarmos o #include com " " e < > ?

Exemplo :

```
#include <asf.h>  
#include "asf.h"
```

1.2 PIO

Nessa biblioteca iremos criar uma série de funções que lidam com a configuração e controle do PIO. As funções a serem implementadas já foram definidas no pio_maua.h, porém devemos agora fazer a implementação de sua lógica.

DICA: implemente e teste por partes o projeto.

Driver - PIO e PMC

Rafael Corsi

Generated by Doxygen 1.8.6

Thu Mar 31 2016 19:01:02

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/pio_maua.h File Reference . .	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	_pio_clear	4
2.1.3.2	_pio_get_output_data_status	4
2.1.3.3	_pio_pull_down	4
2.1.3.4	_pio_pull_up	5
2.1.3.5	_pio_set	5
2.1.3.6	_pio_set_input	5
2.1.3.7	_pio_set_output	5
2.2	/home/corsi/Documentos/EEN251/Codigos/08-DRIVER_PIO_PCM/pmc_maua.h File Reference . .	5
2.2.1	Detailed Description	6
2.2.2	Function Documentation	6
2.2.2.1	_pmc_disable_clock_periferico	6
2.2.2.2	_pmc_enable_clock_periferico	6
	Index	8

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_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_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_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

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

Note

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

Return values

<i>0</i>	se bem-sucedido
<i>1</i>	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](#)