

## 07 - Pesquisa PIO Output

Rafael Corsi  
rafael.corsi@insper.edu.br

20 de fevereiro de 2017

*Entregar na 6ª em formato PDF via github.*

1. Criar softwares para microcontroladores utilizando suas especificidades (periféricos/ low power);
2. Avaliar e melhorar soluções embarcadas integrando hardware/software levando em conta adequação a uma aplicação;
3. Integrar em um protótipo hardware, software básico, sistema operacional de tempo real e módulos de interfaceamento com usuários, de comunicação e de alimentação;
4. Compreender as limitações de microcontroladores e seus periféricos;
5. Buscar e analisar documentação (datasheet) e extrair informações

### 1 Periféricos

*(Utilize o manual encontrado em : Manuais/Cortex-M7-SAM-E70 para resolução dessa secção.)*

Periféricos são hardwares auxiliares encontrados no uC que fornecem funcionalidades extras tais como : gerenciador de energia (SUPC), comunicação serial UART (UART), comunicação a dois fios (TWI), controlador de saída e entrada paralela (PIO), dentre muitos outros.

### Questão. 1.1: Periféricos

Liste a funcionalidade dos periféricos listados a seguir :

- RTC - Real time clock /
- TC - Timer/Counter

Os periféricos são configuráveis via escrita/leitura nos registradores do microcontrolador, cada periférico possui um endereço único mapeado em memória. A Fig. 1 define os endereços de memórias reservado para cada funcionalidade do uC.

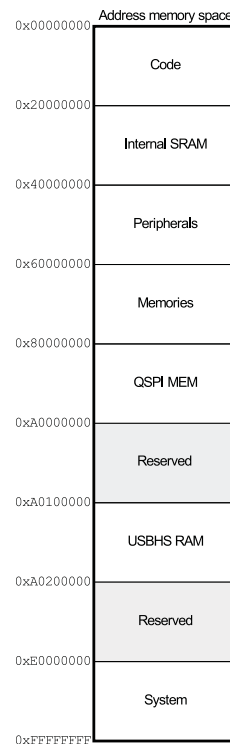


Figura 1: Mapa de memória SAME70  
( *Cortex-M7-SAM-E70- pg. 41* )

### Questão. 1.2: Mapa de memória

- Qual endereço de memória reservado para os periféricos ?
- qual o tamanho (em endereço) dessa secção ?

! O diagrama completo do mapeamento de memória pode ser encontrado na página 41.

### Questão. 1.3: Periféricos

Encontre os endereços de memória referentes aos seguintes periféricos :

1. PIOA
2. PIOB
3. ACC
4. UART1
5. UART2

## 2 PMC - Gerenciador de energia

O Power Management Controller (PMC) é um periférico responsável por gerenciar a energia e clock dos demais periféricos. Para utilizarmos um periférico é necessário primeiramente ativarmos o mesmo no PMC.

Cada periférico é referenciado no PMC via um número único (ID), esse ID também será utilizado para o gerenciamento de interrupções. Os IDs estão listados na Tabela : 13.1 do datasheet Cortex-M7-SAM-E70.pdf.

Table 13-1. Peripheral Identifiers

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Description
0	SUPC	X	–	Supply Controller
1	RSTC	X	–	Reset Controller
2	RTC	X	–	Real Time Clock
3	RTT	X	–	Real Time Timer
4	WDT	X	–	Watchdog Timer
5	PMC	X	–	Power Management Controller
6	EFC	X	–	Enhanced Embedded Flash Controller
7	UART0	X	X	Universal Asynchronous Receiver/Transmitter
8	UART1	X	X	Universal Asynchronous Receiver/Transmitter

Figura 2: Lista de periféricos e IDs

### Questão. 2.1: PIO D ID

Qual ID do PIOC ?

## 3 Parallel Input Output (PIO)

## Secção 32 datasheet

! Leitura obrigatória, Secção 32 do datasheet.

No ARM-Atmel os pinos são gerenciados por um hardware chamado de Parallel Input/Output Controller (PIO), esse dispositivo é capaz de gerenciar até 32 diferentes pinos (I/Os).

Além do controle direto do pino pelo PIO, cada I/O no ARM-Atmel pode ser associado a uma função diferente (periférico), por exemplo: o I/O *PA20* pode ser controlador pelo periférico do PWM enquanto o *PA18* pode ser controlador pela UART.

Isso fornece flexibilidade ao desenvolvimento de uma aplicação, já que os I/Os não possuem uma funcionalidade fixa. Porém não possuímos tanta flexibilidade assim, existe uma relação de quais I/Os os periféricos podem controlar.

Podemos interpretar a tabela como : o pino **102** do microcontrolador identificado como **PA0** (PIOA\_0) pode ser utilizado como **WKUP0** (wakeup) ou mapeado para um dos tres perifericos :

- Periferico A: PWM (Pulse width modulation)
- Periferico B: TIOA0 (Timer 0)
- Periferico C: I2C\_MCL (I2C master clear)

A tabela na página 16 do datasheet (Table 5-1) ilustra quais periféricos podem ser associados aos respectivos pinos, a Fig. 3 mostra as opções para o PIOA0 até PIOA9.

Table 5-1. 144-lead Package Pinout

LQFP Pin	LFBGA Ball	UFBGA Ball	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral C		PIO Peripheral D		Reset State
					Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	
102	C11	E11	VDDIO	GPIOD_AD	PA0	I/O	WKUP0 <sup>(1)</sup>	I	PWMCO_PWMH0	O	TIOA0	I/O	A17/BA1	O	I2SC0_MCK	–	PIO, I, PU, ST
99	D12	F11	VDDIO	GPIOD_AD	PA1	I/O	WKUP1 <sup>(1)</sup>	I	PWMCO_PWMH0	O	TIOB0	I/O	A18	O	I2SC0_OK	–	PIO, I, PU, ST
93	E12	G12	VDDIO	GPIOD	PA2	I/O	WKUP2 <sup>(1)</sup>	I	PWMCO_PWMH1	O	–	–	DATRG	I	–	–	PIO, I, PU, ST
91	F12	G11	VDDIO	GPIOD_AD	PA3	I/O	PIODC0 <sup>(2)</sup>	I	TWD0	I/O	LONCOL1	I	PCK2	O	–	–	PIO, I, PU, ST
77	K12	L12	VDDIO	GPIOD	PA4	I/O	WKUP3/PIODC1 <sup>(2)</sup>	I	TWCK0	O	TCLK0	I	UTXD1	O	–	–	PIO, I, PU, ST
73	M11	N13	VDDIO	GPIOD_AD	PA5	I/O	WKUP4/PIODC2 <sup>(2)</sup>	I	PWMC1_PWMH3	O	ISI_D4	I	URXD1	I	–	–	PIO, I, PU, ST
114	B9	B11	VDDIO	GPIOD_AD	PA6	I/O	–	–	–	–	PCK0	O	UTXD1	O	–	–	PIO, I, PU, ST
35	L2	N1	VDDIO	CLOCK	PA7	I/O	XIN32 <sup>(3)</sup>	I	–	–	PWMCO_PWMH3	O	–	–	–	–	PIO, HZ
36	M2	N2	VDDIO	CLOCK	PA8	I/O	XOUT32 <sup>(3)</sup>	O	PWMC1_PWMH3	O	AFE0_ADTRG	I	–	–	–	–	PIO, HZ
75	M12	L11	VDDIO	GPIOD_AD	PA9	I/O	WKUP6/PIODC3 <sup>(2)</sup>	I	URXD0	I	ISI_D3	I	PWMCO_PWMH0	I	–	–	PIO, I, PU, ST

Figura 3: Mux PIOA - periféricos vs pinos  
Cortex-M7-SAM-E70pg. 16

### Questão. 3.1: PIO periféricos

Verifique quais periféricos podem ser configuráveis nos I/Os :

1. PC1
2. PB6

O SAME70 possui internamente 5 PIOs: PIOA, PIOB, PIOC, PIOD e PIOE. Cada um é responsável por gerenciar até 32 pinos.

Os I/Os são classificados por sua vez em grandes grupos: A, B,C . . . . (exe: PA01, PB22, PC12) e cada grupo é controlado por um PIO (PIOA, PIOB, PIOC, . . . ).

### 3.1 Configurações

O PIO suporta as seguintes configurações :

- Interrupção em nível ou borda em qualquer I/O
- Filtragem de "glitch"
- Debouncing
- Open-Drain
- Pull-up/Pull-down
- Capacidade de trabalhar de forma paralela

### Questão. 3.2: Debouncing

- O que é debouncing ?
- Descreva um algoritmo que implemente o debouncing.

### 3.2 Funcionalidade

O diagrama de blocos do PIO é ilustrado na Fig. 4, onde :

1. Peripheral DMA (direct memory access) controller (PDC) : O P/IO pode receber dados via DMA.
2. Interrup Controller : Já que o PIO suporta interrupções nos I/Os o mesmo deve se comunicar com o controlador de interrupções para informar a CPU (NVIC) que uma interrupção ocorreu.

3. PMC : A energia e clock desse periférico é controlado pelo PMC (Power management controller).
4. Embedded peripheral : O acesso aos pinos dos periféricos é realizado via PIO.

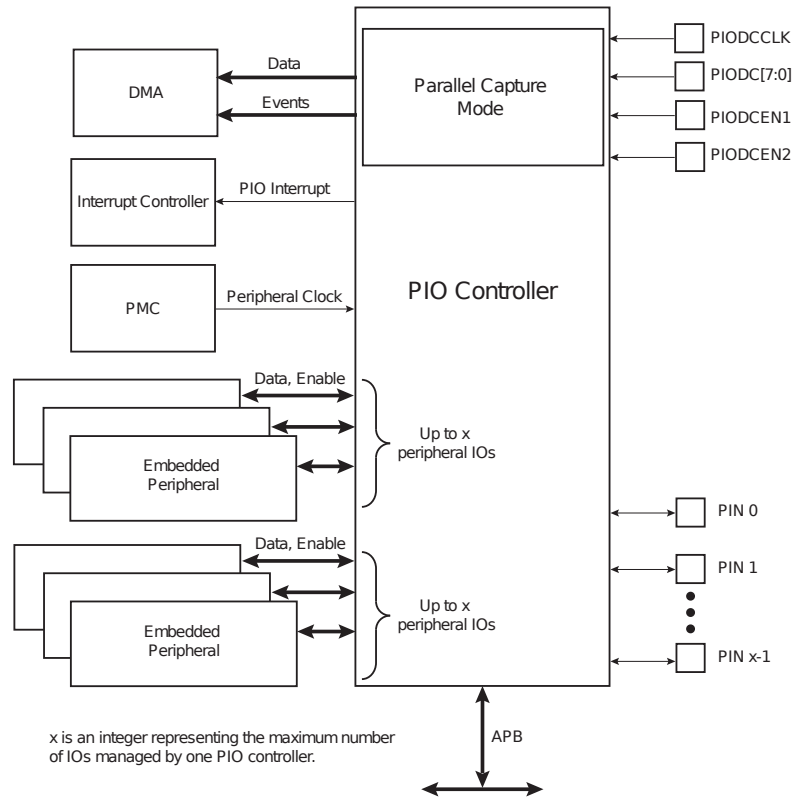


Figura 4: Block Diagram  
Cortex-M7-SAM-E70- pg. 346

Im diagrama lógico mais detalhado é encontrado no datasheet (Fig. 5), esse diagrama mostra as funções dos registradores e seu impacto no PIO.

### 3.3 SET/Clear

Algumas funcionalidades no PIO são configuráveis via dois registradores distintos (set/-clear), o primeiro apenas altera o estado do bit específico de 0 para 1, enquanto que o segundo (clear) altera o estado do registrador de 1 para 0. Isso é utilizado para evitar uma condição de corrida ("race conditions"<sup>1</sup>).

Um exemplo desse caso é o registrador Output Data Status Register (PIO\_ODSR) que configura o valor de saída de um pino, ou seja, se será "1" ou "0". Para configurarmos

<sup>1</sup><http://stackoverflow.com/questions/34510/what-is-a-race-condition/34745#34745>

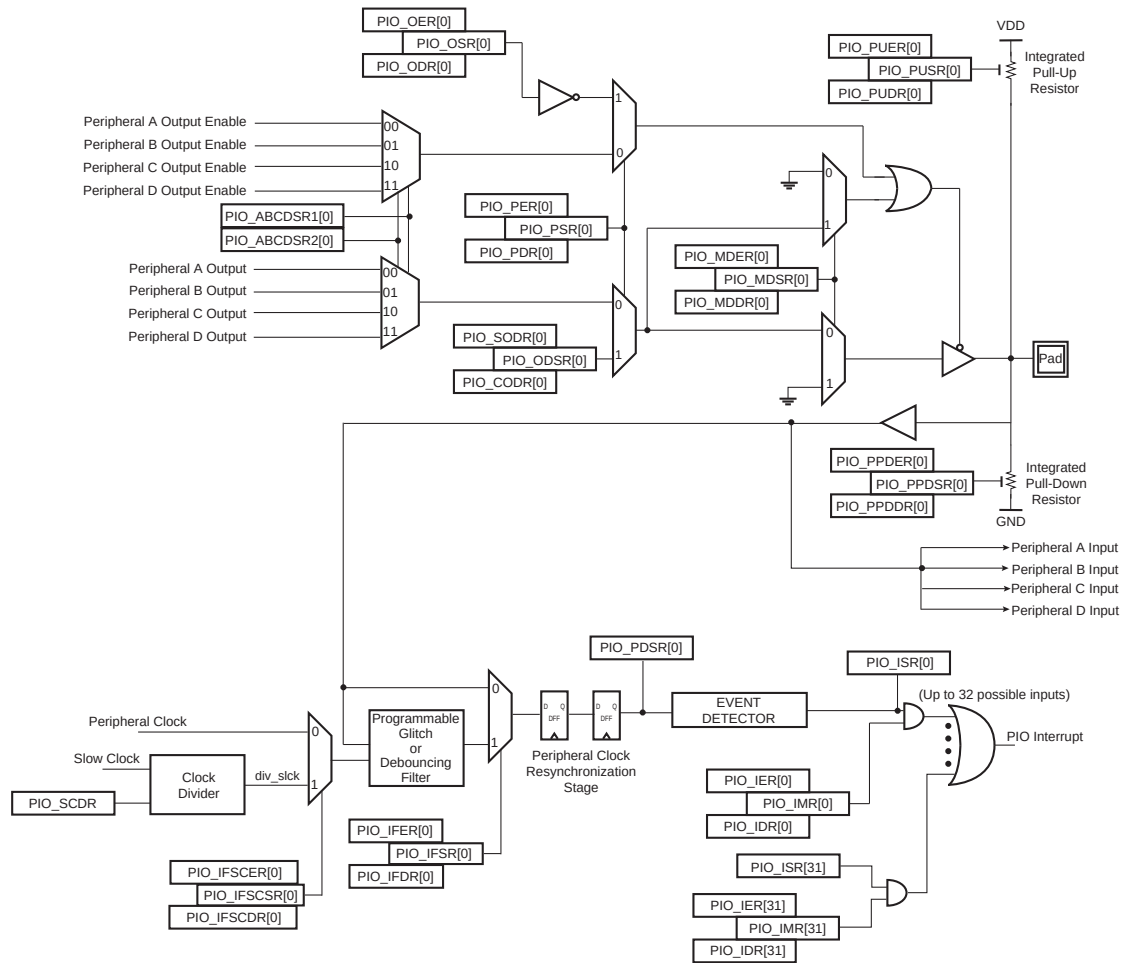


Figura 5: [Datasheet pg. 571] I/O Line Control Logic

por exemplo o bit 2 (referente ao pino PIOA02) precisamos "setar" o segundo bit via o registrador Set Output Data Register (IO\_SODR):

```
PIOA->SODR = 1 << 1;
```

Não podemos por exemplo zerar esse registrador :

```
PIOA->SODR &= ~(1 << 1);
```

Para tanto é necessário utilizarmos o registrador Clear Output Data Register (CODR)

```
PIOA->CODR = 1 << 1;
```

Esse tipo de controle evita que tenhamos que fazer uma operação de leitura no registrador antes de alterar o bit específico, aumentando assim a eficiência desse periférico.

### Questão. 3.3: Race condicions

- O que é *race condicions* ?
- Como que essa forma de configurar os registradores evita isso?

## 3.4 Configurando um pino em modo de output

### 31.5.4 - Output Control (Cortex-M7-SAM-E70. 550)

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO\_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO\_ABCDSR1 and PIO\_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO\_OER) and Output Disable Register (PIO\_ODR). The results of these write operations are detected in the Output Status Register (PIO\_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO\_SODR) and the Clear Output Data Register (PIO\_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO\_ODSR), which represents the data driven on the I/O lines. Writing in PIO\_OER and PIO\_ODR manages PIO\_OSR whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO\_SODR and PIO\_CODR affects PIO\_ODSR. This is important as it defines the first level driven on the I/O line.

### Questão. 3.4: Pino em modo output

- Explique com suas palavras o trecho anterior extraído do datasheet do uC, se possível referencie com o diagrama "I/O Line Control Logic".



## 4 Avaliação

Insatisfatório (I) :

- não apresentou o estudo
- contém apenas 40% do total pedido
- contém plágio (textos copiados sem referências)

---

Em Desenvolvimento (D) :

- apresentou o estudo com até 2 aulas de atraso
- contém apenas 70% tópicos do exigido
- texto incoerente

---

Essencial (C)

- apresentou o estudo com até 1 dia de atraso
- entregou mais de 70% da atividade com respostas corretas
- texto coerente

---

Proficiente (B)

- apresentou o estudo sem atraso
- entregou mais de 70% da atividade com respostas corretas

---

Avançado (A)

- apresentou o estudo sem atraso
  - entregou mais de 90% da atividade com respostas corretas
  - texto consistente e bem escrito
-