

## Computação Embarcada - Aula 03 - Estudo - Periféricos

Rafael Corsi - rafael.corsi@insper.edu.br

Fevereiro 2018

! Para a aula do dia 5/3

### Table of Contents

- Periféricos
- PMC - Gerenciador de energia
- Parallel Input Output (PIO)
  - Configurações
  - Funcionalidade

## Periféricos

*(Utilize o manual encontrado em : Manuais/SAME70.pdf para resolução dessa seção.)*

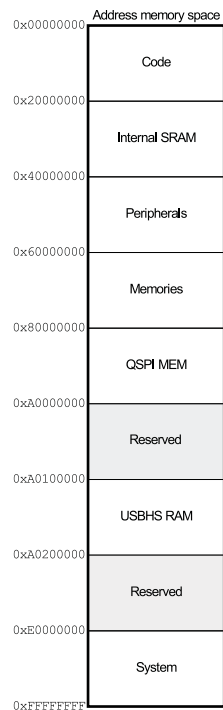
Periféricos são hardwares auxiliares encontrados no uC que fornecem funcionalidades extras tais como : gerenciador de energia (SUPC), comunicação serial UART (UART), comunicação a dois fios (TWI), controlador de saída e entrada paralela (PIO), dentre muitos outros.

### Pergunta 1

Liste as principais funcionalidade dos periféricos listados a seguir :

- RTC - Real time clock
- TC - Timer/Counter
- ISI - Image Sensor Interface

Os periféricos são configuráveis via escrita/leitura nos registradores do micro-controlador, cada periférico possui um endereço único mapeado em memória. A Fig. Mapa de memória SAME70 define os endereços de memórias reservado para cada funcionalidade do uC.



### Pergunta 2

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



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

### Pergunta 3

Encontre os endereços de memória referentes aos seguintes periféricos : 1. PIOA 2. PIOB 3. ACC 4. UART1 5. UART2

## 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

Figure 1: Lista de periféricos e IDs

#### Pergunta 4

- Qual ID do TC0 ?

## 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. Mux PIOA 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	
902	C11	E11	VDDIO	GPIO_AD	PA0	IO	WKUP0 <sup>1</sup>	I	PWM00_PWMH0	O	TIOA0	IO	A17BA1	O	I2SD0_MCK	–	PIO, I, PU, ST
99	D12	F11	VDDIO	GPIO_AD	PA1	IO	WKUP1 <sup>1</sup>	I	PWM00_PWMH0	O	TIOB0	IO	A18	O	I2SD0_OK	–	PIO, I, PU, ST
93	E12	G12	VDDIO	GPIO	PA2	IO	WKUP2 <sup>1</sup>	I	PWM00_PWMH1	O	–	–	DATRG	I	–	–	PIO, I, PU, ST
91	F12	G11	VDDIO	GPIO_AD	PA3	IO	PDOC0 <sup>2</sup>	I	TWD0	IO	LONCOL1	I	PK2	O	–	–	PIO, I, PU, ST
77	K12	L12	VDDIO	GPIO	PA4	IO	WKUP3PDOC1 <sup>3</sup>	I	TWCK0	O	TCUK0	I	UTXD1	O	–	–	PIO, I, PU, ST
73	M11	N13	VDDIO	GPIO_AD	PA5	IO	WKUP4PDOC2 <sup>3</sup>	I	PWM01_PWMH3	O	ISL_D4	I	URXD1	I	–	–	PIO, I, PU, ST
114	B9	B11	VDDIO	GPIO_AD	PA6	IO	–	–	–	–	PK0	O	UTXD1	O	–	–	PIO, I, PU, ST
35	L2	N1	VDDIO	CLOCK	PA7	IO	XIN32 <sup>4</sup>	I	–	–	PWM00_PWMH3	O	–	–	–	–	PIO, HZ
36	M2	N2	VDDIO	CLOCK	PA8	IO	XOUT32 <sup>4</sup>	O	PWM01_PWMH3	O	AFEB_ADTRG	I	–	–	–	–	PIO, HZ
75	M12	L11	VDDIO	GPIO_AD	PA9	IO	WKUP6PDOC3 <sup>3</sup>	I	URXD0	I	ISL_D3	I	PWM00_PWMH0	I	–	–	PIO, I, PU, ST

Figure 2: Mux PIOA - periféricos vs pinos pg. 16

## Pergunta 5

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, ...).

## 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

## Pergunta 6

- O que é debouncing ?
- Porque deve ser utilizado ?
- Descreva um algoritmo que implemente o debouncing.

## Funcionalidade

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

1. Peripheral DMA (direct memory access) controller (PDC) : O P/IO pode receber dados via DMA.
2. Interrupt 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.

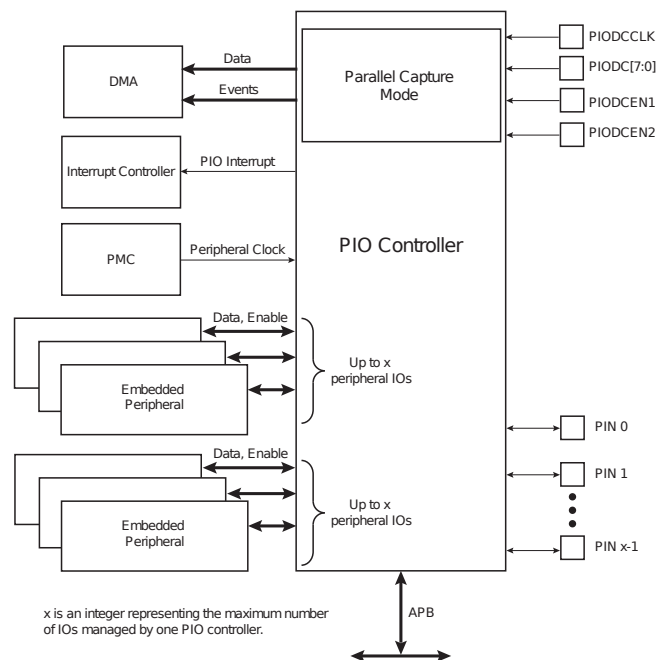


Figure 3: Block Diagram - pg. 346

6