

## 08 - Projeto PIO Input

Rafael Corsi  
rafael.corsi@insper.edu.br

8 de março de 2017

*Entregar até o final da aula*

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 Entendimento

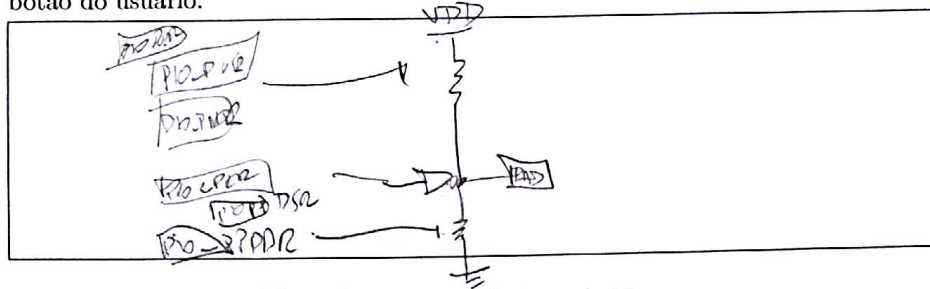
Passos necessários para o entendimento correto de como um pino digital, deve ser configurado para ser controlado pelo PIO.

1. Identifique no kit de desenvolvimento (SAME-70-XPLD) qual botão é reservado para o usuário :

O botão ao lado do botão de RESET.

PIO A | Pio. ID 10 | PIN 11

2. Via o manual do kit de desenvolvimento, redesenhe o esquema elétrico referente ao botão do usuário:



3. Identifique o pino, o PIO e o bit correspondente ao botão.

Pino	PIO	Bit
11	A	31

4. Preencha a tabela a seguir com o valor lido nos casos em que o botão não está pressionado e quando está.

Botão Pressionado (SW0)	Valor digital lido no PIO
Sim	0
Não	1

5. Identifique os registradores que considera necessário modificar (leia a secção do datasheet referente a configurar o PIO em modo de entrada).

## 2 Programação parte 1 - SW0

Com base no projeto passado, utilize o botão reservado ao usuário para ativar o LED (Botão não pressionado = LED acesso/ Botão pressionado = Led apagado). Importe o projeto base localizado em : /codigos/08-PIO-INPUT para o seu repositório e modifique-o de acordo com os passos a seguir.

1. Importe as configurações do LED do projeto 07-PIO-INPUT. Coloque essas configurações dentro da função : ledConfig().
2. Ative o clock do PIO que controla o botão no PMC:

Registrador: PMC\_PCR0 → quando Bit 10

3. Ative o PIO para controlar o pino do botão

Registrador: PIO\_OSR

4. Desative o buffer de saída

Registrador:	PIO_PDR
--------------	---------

5. Ative o pull-up

Registrador:	PIO_PDR
--------------	---------

Explique porque isso é necessário :

Para garantir que o led ira acender.

6. Decida se irá ativar ou não o debouncing

- não ativado : deve tratar o problema por sw

Registrador:	
--------------	--

- ativado : deve configurar os registradores (mais indicado)

Registrador:	PIO_PDR
Registrador:	PIO_PDR
Registrador:	PIO_PDR

7. faça a leitura periódica no while(1) para checar se o botão foi ou não pressionado alterando, altere o status do led dependendo do botão.

Registrador:	PIO_PDR
--------------	---------

6 - While(1)

Qual a alternativa para evitar que o status do botão seja (ou precise ser) verificado continuamente?

### 3 Programação parte 2 - Blink

Utilize o botão para controlar se o LED irá piscar ou não.