

ELF52 - Sistemas Microcontrolados

Pinos de Entrada/Saída - Parte 2

Professor:

Prof. Marcos Eduardo

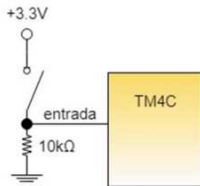
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GPIO - Chaves e LEDs

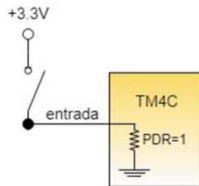
Entrada com Chaves

- Há as seguintes formas de interfacear com uma chave:

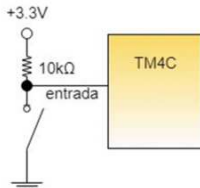
Lógica positiva, resistor externo



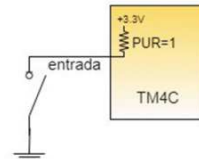
Lógica positiva, resistor interno



Lógica negativa, resistor externo



Lógica negativa, resistor interno



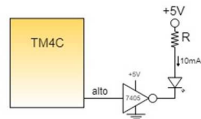
Saída com LEDs

- Há as seguintes formas de interfacear com LEDs:
- Uma porta no TM4C1294 suporta no máximo 12mA, mas o microcontrolador não suporta todas as portas drenando/suprindo este máximo de corrente para todas as portas. (Ver Datasheet seção 27.3.2.1);
- O valor **default** é cada porta drenar/suprir 2mA por porta;
- Se precisar que uma porta forneça mais corrente que ela suporta, deve-se utilizar um driver com circuito integrado ou transistor.

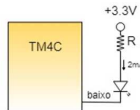
Lógica positiva, corrente baixa



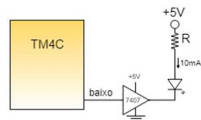
Lógica positiva, corrente alta



Lógica negativa, corrente baixa



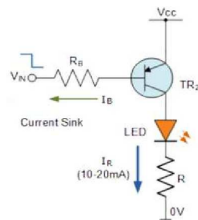
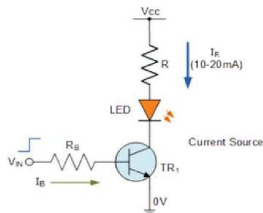
Lógica negativa, corrente alta



Saída com LEDs

- Para drivers com CIs:
 - Para lógica positiva: Exemplo \rightarrow 74xx05 ou 74xx06;
 - Para lógica negativa: Exemplo \rightarrow 74xx07.

- Para transistores, usar operação como chave:
 - Calcular R_C e R_B ;
 - Região corte-saturado;
 - **Como calcular?**



Saída com LEDs

- Para transistores, operação como chave:

- Calcular R_C e R_B ;
- Região corte-saturado;

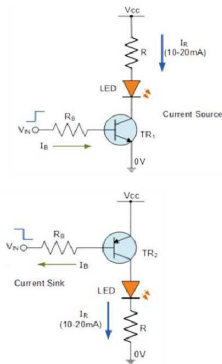
$$I_C = \frac{V_{CC} - V_{LED} - V_{CE}}{R_C}$$

$$I_B = \frac{I_C}{\beta}$$

$$I'_B = 5I_B \quad (\text{para garantir a saturação})$$

$$R_B = \frac{V_{IN} - V_{BE}}{I'_B}$$

(1)



Exercícios

1 Exemplo de inicialização do GPIO.

Verificar a inicialização da porta J e porta F. Os pinos J0 e J1 estão ligados às chaves tácteis USR_SW1 e USR_SW2, respectivamente e os pinos F4 e F0 estão ligados aos LEDs 3 e 4, respectivamente.

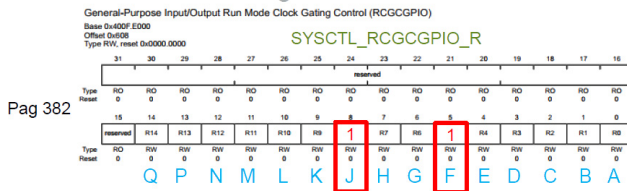
- a) Baixe e abra o projeto GPIO1 no moodle;
- b) Faça o *build* e execute passo-a-passo para verificar a inicialização das portas;
- c) Execute e teste pressionando os dois botões e verificar se os LEDs acendem;
- d) Faça um fluxograma do código.

Exercício (Passo a passo)

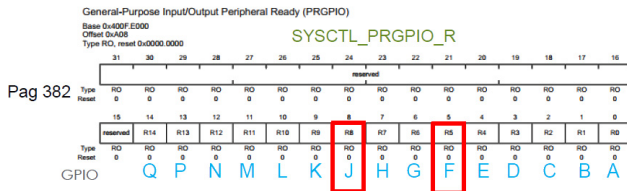
- A seguir será demonstrado o passo-a-passo para a inicialização dos GPIOs J e F para este exercício, conforme os slides apresentados na parte 1, correspondentes à seção 10.6 do *datasheet*.

Exercício (Passo a passo)

- 1 Como vamos utilizar os ports F e J, setar os bits 5 e 8 no registrador **RCGCGPIO** e depois esperar enquanto os bits 5 e 8 do registrador **PRGPIO** não estão setados



Setar estes bits

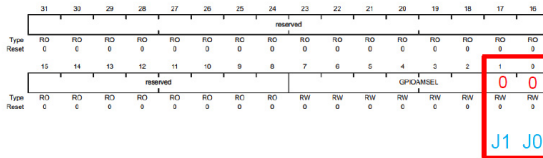


Ficar testando estes bit, enquanto **NÃO** sejam 1

Exercício (Passo a passo)

- 2 a) Como vamos utilizar os pinos J0 e J1 como entrada digital zerar pelo menos os bits 0 e 1 do **GPIOAMSEL** do Port J.

GPIO_PORTJ_AHB_AMSEL_R



Zerar estes bits

Pag 786

Exercício (Passo a passo)

- 2 b) Como vamos utilizar os pinos F0 e F4 como saída digital zerar pelo menos os bits 0 e 4 do **GPIOAMSEL** do Port F.

GPIO_PORTF_AHB_AMSEL_R

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								AMSEL							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
												F4	F3	F2	F1	F0

Zerar estes bits

Pag 786

Exercício (Passo a passo)

- 3 a) Como vamos utilizar os pinos J0 e J1, como GPIO, ou seja, sem função alternativa, zerar os bits correspondentes ao J0 e ao J1 do **GPIOPCTL** do Port J.

GPIO_PORTJ_AHB_PCTL_R

Pag 788

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMC7				PMC6				PMC5				PMC4			
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC3				PMC2				PMC1		0	0	PMC0		0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

J1

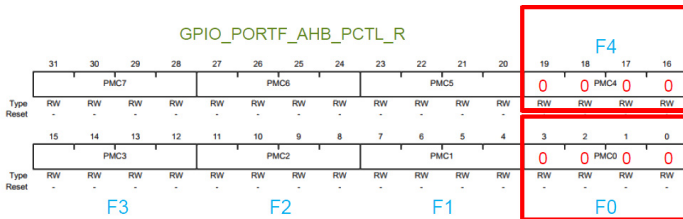
J0

Zerar estes bits

Exercício (Passo a passo)

- 3 b) Como vamos utilizar os pinos F0 e F4, como GPIO, ou seja, sem função alternativa, zerar os bits correspondentes ao F0 e ao F4 do **GPIOCTL** do Port F.

Pag 788

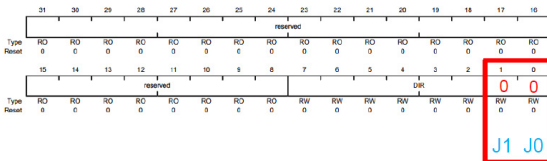


Zerar estes bits

Exercício (Passo a passo)

- 4 a) Como vamos utilizar os pinos J0 e J1 como entrada digital, zerar os bits 0 e 1 do **GPIO_DIR** do Port J.

GPIO_PORTJ_AHB_DIR_R

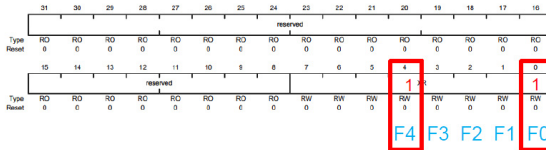


Zerar estes bits

Exercício (Passo a passo)

- 4 b) Como vamos utilizar os pinos F0 e F4 como saída digital, setar os bits 0 e 4 do **GPIODIR** do Port F.

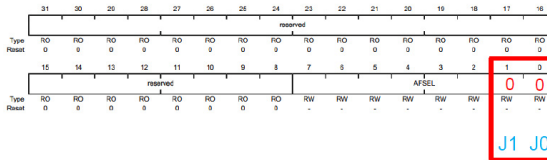
GPIO_PORTF_AHB_DIR_R



Exercício (Passo a passo)

- 5 a) Como vamos utilizar os pinos J0 e J1 como GPIO, desabilitar a função alternativa, zerar pelo menos os bits 0 e 1 do **GPIOAFSEL** do Port J.

GPIO_PORTJ_AHB_AFSEL_R



Zerar estes bits

Exercício (Passo a passo)

- 5 b) Como vamos utilizar os pinos F0 e F4 como GPIO, desabilitar a função alternativa, zerar pelo menos os bits 0 e 4 do **GPIOAFSEL** do Port F.

GPIO_PORTF_AHB_AFSEL_R

Pag 770

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	0	RW	RW	RW	0
Reset	0	0	0	0	0	0	0	0	-	-	-	0	-	-	-	0
									F4	F3	F2	F1	F0			

Zerar estes bits

Exercício (Passo a passo)

- 6 a) Como vamos utilizar os pinos J0 e J1 como GPIO, habilitar a função digital, setar os bits 0 e 1 do **GPIOEN** do Port J.

GPIO_PORTJ_AHB_DEN_R

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								DEN							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	-	-	-	-	-	-	1	1
															J1	J0

Setar estes bits

Pag 782

Exercício (Passo a passo)

- 6 b) Como vamos utilizar os pinos F0 e F4 como GPIO, habilitar a função digital, setar os bits 0 e 4 do **GPIODEN** do Port F.

GPIO_PORTF_AHB_DEN_R

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	-	-	-	1	-	-	-	1
												F4	F3	F2	F1	F0

Setar estes bits

Pag 782

Exercício (Passo a passo)

- Como vamos utilizar os pinos J0 e J1 como entrada para chaves e não há resistor de pull up externos, habilitar os pull up internos, setar os bits 0 e 1 do **GPIOPUR** do Port J.

GPIO_PORTJ_AHB_PUR_R

Pag 777

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Rreset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								PUE							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Rreset	0	0	0	0	0	0	0	0	-	-	-	-	-	-	1	1
															J1	J0

Setar estes bits

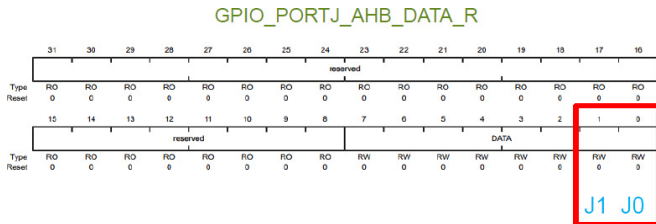
- Obs: Como os pinos F0 e F4 são saídas, não faz sentido habilitar os pull up para eles.

Exercício (Passo a passo)

- Em relação à leitura das chaves e escrita nos LEDs, devemos utilizar os registradores **GPIODATA**. Para ler as chaves, vamos ler o registrador **GPIODATA** do Port J. Para acender ou apagar os LEDs vamos escrever no registrador **GPIODATA** do Port F.

Exercício (Passo a passo)

- Como as chaves estão conectadas em resistores de pull up internos, se as chaves estão soltas retornam o valor 1. Se as chaves estão pressionadas, retornam o valor 0, no registrador **GPIODATA** do Port J.



- Se a chave USR_SW1, que está ligada ao pino J0, estiver solta, o bit 0, quando lido, retorna 1, se ela estiver pressionada o bit 0 retorna 0. Se a chave USR_SW2, que está ligada ao pino J1, estiver solta, o bit 1, quando lido, retorna 1, se ela estiver pressionada o bit 0 retorna 0.

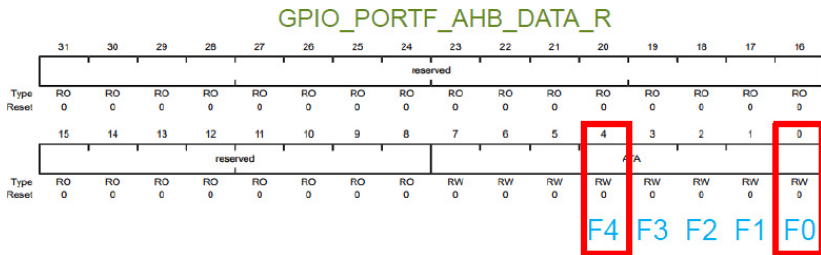
Exercício (Passo a passo)

- Como ativamos os dois pinos do port J, J0 e J1, de uma forma geral não conseguimos ler os bits separadamente. Só conseguimos ler **GPIODATA** do Port J inteiro, que pode retornar os valores:

GPIODATA (Port J)	Estado das chaves	
	USR_SW1 (J0)	USR_SW2 (J1)
11	SOLTA	SOLTA
10	PRESSIONADA	SOLTA
01	SOLTA	PRESSIONADA
00	PRESSIONADA	PRESSIONADA

Exercício (Passo a passo)

- Para escrever nos LEDs, LED3 e LED4, que estão ligados respectivamente aos pinos F4 e F0, devemos escrever diretamente no Port F.



- Escrevendo 0 no bit F4, apaga se o LED3. Escrevendo 1 neste bit, acende se o LED3. Escrevendo 0 no bit F0, apaga se o LED4, escrevendo 1 acende se o LED4. Deve se tomar cuidado, entretanto para ao escrever em um LED não interferir o outro (escrita amigável).

Clock e SysTick

Clock

- Microcontroladores podem ter *clock*:
 - Externo: provido geralmente por um cristal;
 - Interno: geralmente um oscilador R-C mais impreciso e de menor frequência.

PLL

- Normalmente a velocidade de execução de um microcontrolador é determinada pelo cristal externo, no caso do TM4C1294 é 25MHz;
- Muitos microcontroladores possuem um **PLL** que possibilita **ajustar a velocidade** de execução por software;
- Normalmente a frequência é um *tradeoff* entre velocidade de execução e potência elétrica;
- Para informações avançadas sobre o gerenciamento de *clock* (Seção 5.2.5 do *Datasheet*).

SysTick

- Contador simples para gerar atrasos e gerar interrupções periódicas em todos os Cortex-M;
- Fácil de portar para outros microcontroladores;
- 4 passos para ativar:
 - 1 Limpar o bit **ENABLE** no registrador **NVIC_ST_CTRL_R** para desligar o SysTick durante a inicialização;
 - 2 Sete o registrador **NVIC_ST_RELOAD_R**;
 - 3 Escreva qualquer valor no registrador **NVIC_ST_CURRENT_R** para limpar o contador.
 - 4 Sete os bits **CLK_SRC** e **ENABLE** no registrador **NVIC_ST_CTRL_R**. Como interrupções ainda não serão tratadas, não é necessário setar o bit **INTEN**.

SysTick

- Quando a contagem no registrador **CURRENT** mudar de 1 para 0, o *flag* **COUNT** será setado;
- Se ativar o PLL para rodar o microcontrolador em 80MHz então o contador do SysTick decrementa a cada 12,5 ns;
- Em geral se o período do *clock* de barramento é t o *flag* de contagem será setado a cada $(n + 1)t$, tal que n é o valor do registrador **RELOAD**;
- Se escrever no registrador **CURRENT**, o contador será zerado e o *flag* de contagem no registrador CTRL será limpo.

Exercícios

2. Piscar um LED. Baseando-se no exemplo anterior, criar um projeto que pisque um LED a cada intervalo, quando pressionado um botão.
 - a) Baixe o projeto gpio2 do moodle;
 - b) Abra o projeto no *Keil MDK*;
 - c) Faça um fluxograma do problema proposto;
 - d) Modifique o arquivo gpio.s para inicializar os GPIO para uma chave e um LED;
 - e) Modifique o arquivo main.s para fazer o que foi pedido no enunciado;
 - f) Primeiramente, faça apenas o LED acender;
 - g) Depois que esta parte estiver pronta, faça o LED piscar;
 - h) Para fazer o LED piscar utilize a rotina SysTick_Wait1ms.

Tarefa de Casa

- 3 Ler o capítulo 10 do datasheet.
 - a) Aspectos gerais dos GPIO;
 - b) Funções alternativas dos pinos de I/O (Tabela 10-2);
 - c) Descrição do funcionamento;
 - d) Inicialização e Configuração;
 - e) Mapa de Registradores e suas descrições.

Dúvidas?