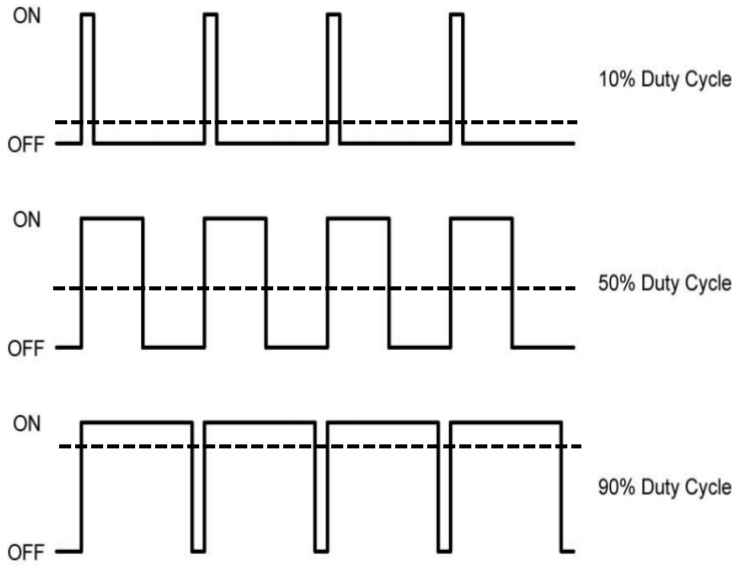
1. **De um exemplo de um sinal digital que pode ser utilizado em um projeto de eletrônica embarcada.**

Um sinal digital comumente utilizado em eletrônica embarcada é a modulação por largura de pulso (PWM), a ideia por trás deste sinal é controlar a tensão média do sinal por meio de seu ciclo de trabalho, como mostra a figura a seguir:



As linhas tracejadas representam a tensão média aplicada (e observada pela carga), conforme ocorre a variação de seu ciclo de trabalho. Este sinal pode ser utilizado em aplicações para controle de temperatura na qual a velocidade da ventoinha (cooler) é regulada pela tensão média. Outra aplicação é sua utilização no controle de servo motores.

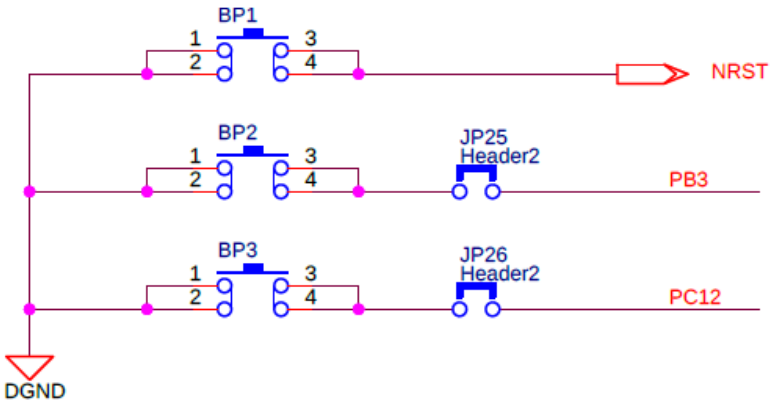
2. **Qual o valor dos resistores de pull-up e pull-down?**

Conforme visto no manual do microcontrolador, tanto o resistor de pull-up como o de pull-down possuem valores nominais de 100kΩ, o fabricante indica uma faixa de valores que estes resistores podem alcançar, dependendo de parâmetros como temperatura de funcionamento do microcontrolador, indo de 70kΩ até 130kΩ.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$R_{PULLDOWN}$	Pull-down Resistor	PA0–PA31, PB0–PB14, PC0–PC31, NRST	70	100	130	kΩ
R_{PULLUP}	Pull-up Resistor	PA0–PA31, PB0–PB14, PC0–PC31, NRST	70	100	130	kΩ

2.1. Qual o valor lido pelo PIO quando o botão não estiver pressionado e qual o valor lido quando o botão estiver pressionado?

Os botões presentes neste microcontrolador possuem lógica inversa por estarem conectados ao terra do circuito:



Portanto o PIO receberá nível lógico 0 quando botão for pressionado e nível lógico 1 quando a chave estiver aberta.

3. Qual o valor máximo que PIO_SCDR pode assumir? Quando PIO_SCDR for zero, por quanto o clock principal é dividido?

O PIO _SCDR recebe um valor em binário que é aplicado à equação abaixo, sendo DIV o valor armazenado no PIO_SCDR:

$$t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$$

Sendo assim, quando PIO_SCDR for zero, DIV = 0, logo o período resultante tem o dobro do período do clock principal, resultando consequentemente na metade da freqüência do clock principal. Os valores que podem ser armazenados em PIO_SCDR são indicados na tabela abaixo:

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	DIV					

7	6	5	4	3	2	1	0
DIV							

Portanto, podemos concluir que DIV pode receber os valores de 0 até 13, sendo então 13 o valor máximo que PIO_SCDR pode assumir.

4. Interprete os diagramas de tempo a seguir (referentes ao filtro de *glitch* e *debouncing*).

Figure 31-4. Input Glitch Filter Timing

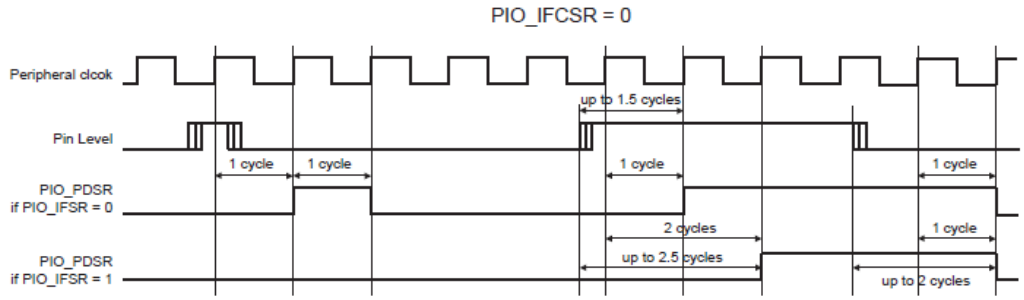
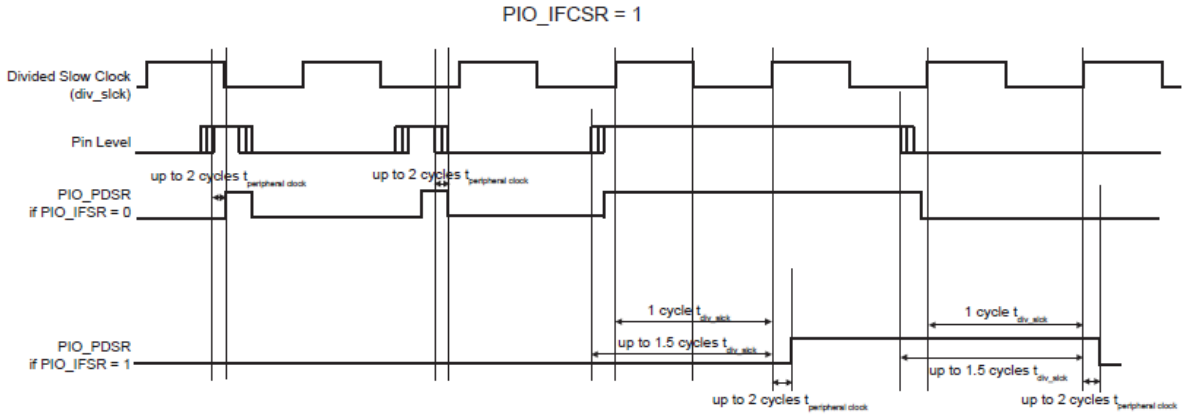


Figure 31-5. Input Debouncing Filter Timing



O diagrama do filtro de *glitch* mostra que apenas após um ciclo de clock do periférico, a resposta aparece em PIO_PDSR. Quando ocorre uma borda de subida no clock e o pino está em nível alto, no PIO_PDSR, após 1 ciclo, aparece nível alto se PIO_IFSR = 0, ou após 2 ciclos com o pino em nível alto se PIO_IFSR = 1, ocorrendo o mesmo para nível baixo.

Já o diagrama do filtro de *debouncing*, quando PIO_IFSR = 0, o estado do pino apenas aparece em PIO_PDSR após 2 ciclos do clock do periférico, e quando PIO_IFSR = 1, o estado do pino apenas aparece em PIO_PDSR após um ciclo do clock do divisor e 2 ciclos do clock do periférico.

5. O que pode acontecer caso configuremos o pino do botão como saída?

Caso configuremos o pino do botão como saída, causamos um curto circuito no sistema e consequentemente queimará a unidade de controle.

6. Qual a alternativa para evitar que o status do botão seja verificado continuamente?

Para evitar que o status do botão seja verificado continuamente pode-se fazer um processamento em paralelo no microcontrolador, de modo que o mesmo só seria acionado se houvesse mudança de estado na saída do botão, isto é, caso o botão fosse apertado um pulso seria enviado e acionaria o código.