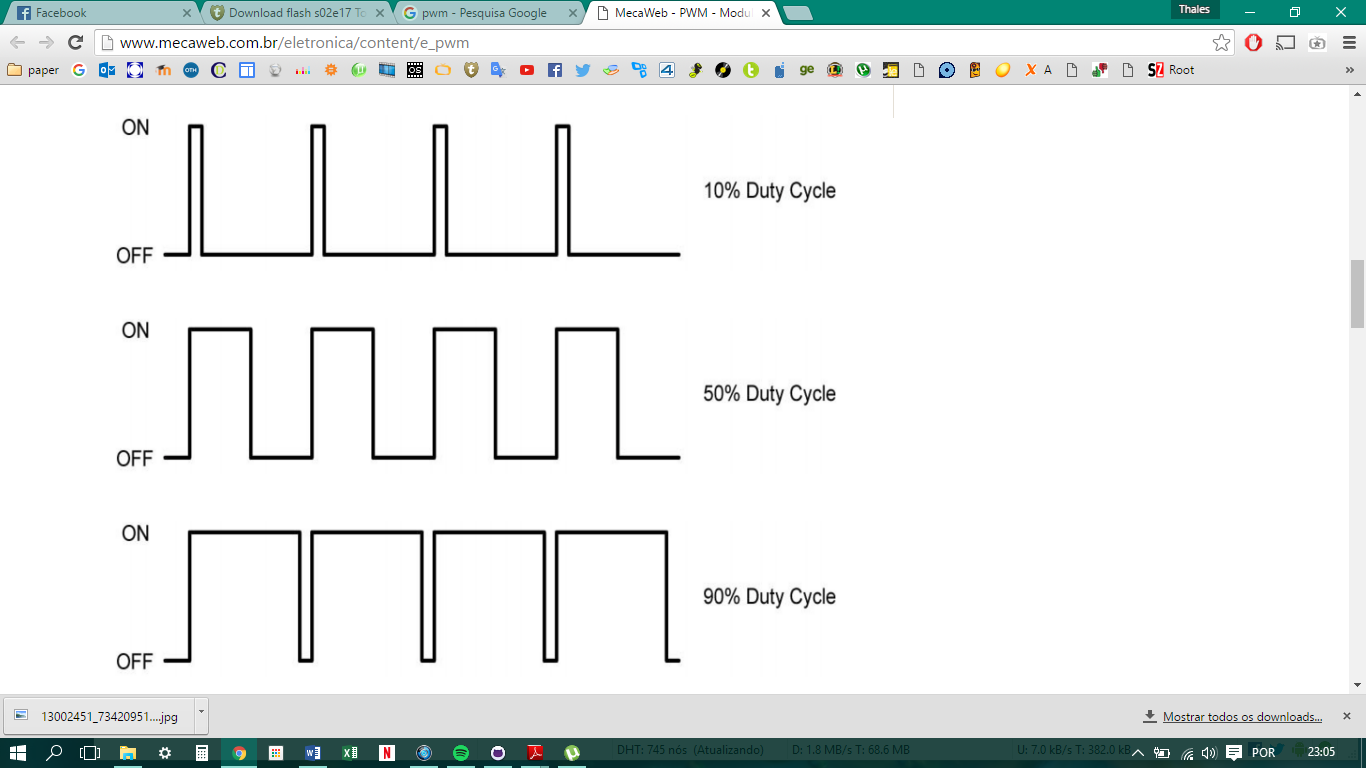
1. **De um exemplo de um sinal digital que pode ser utilizado em um projeto de eletrônica embarcada.**

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:

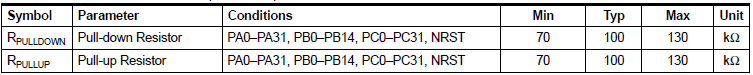


As linhas tracejadas representam a tensão média aplicada (e observada pela carga), conforme ocorre a variação de seu ciclo de trabalho. Este sinal pode ser utilizado em aplicações para controle de temperatura na qual a velocidade da ventoinha (cooler) é regulada pela tensão média. Outra aplicação é sua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

2 - Valores resistores

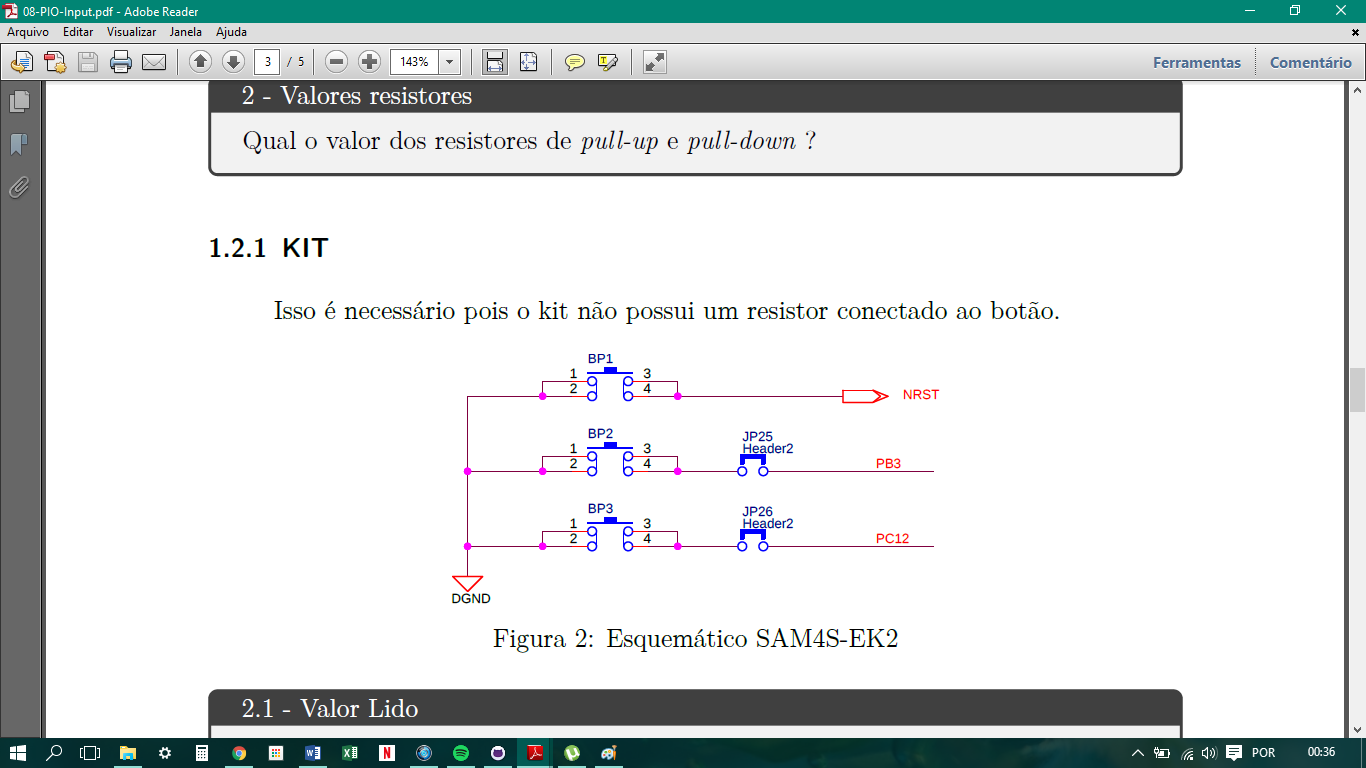
1. **Qual o valor dos resistores de pull-up e pull-down?**

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.



* 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:



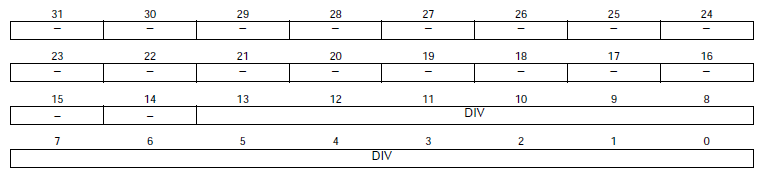
Portanto o PIO receberá nível lógico 0 quando botão for pressionado e nível lógico 1 quando a chave estiver aberta.

1. **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:

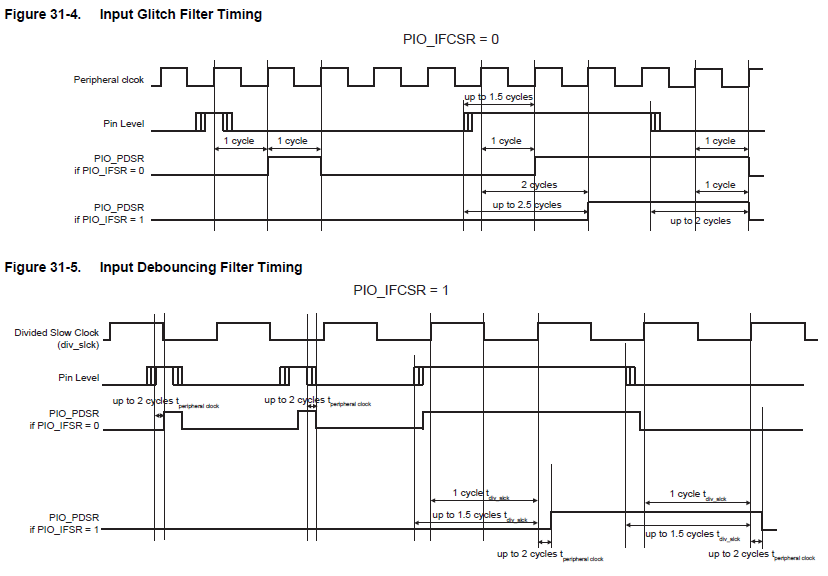


Sendo assim, quando PIO\_SCDR for zero, DIV = 0, logo o período resultante tem o dobro do período do clock principal, resultando consequentemente na metade da freqüência do clock principal. Os valores que podem ser armazenados em PIO\_SCDR são indicados na tabela abaixo:



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

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

****

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.

1. **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.

1. **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.