

Cálculo do Timer 1 do Arduino

Cada ciclo de máquina do ATmega328P corresponde à 01 ciclo de clock do oscilador (pág 09 do datasheet), no caso do Arduino, 16MHz. Cada ciclo de máquina tem, portanto, um período de $62,5ns$.

O Timer1 é de 16 bits, ou seja, conta até $2^{16} = 65536 \rightarrow$ *Iniciando em 0, conta até 65535*. Levando em conta o período do ciclo de máquina, é possível contar, sem utilizar prescaler, um tempo de até $65535 \cdot 62,5ns = 4,0959ms$ ou $4095,9\mu s$.

O Timer1 possui prescaler de 8, 64, 256 e 1024. Abaixo, o quanto pode-se contar com cada prescaler:

- *Sem prescaler* $\rightarrow 4,096ms \rightarrow 4096\mu s$
- *Prescaler 8* $\rightarrow 8 \cdot 4,096 = 32,768ms \rightarrow 32768\mu s$
- *Prescaler 64* $\rightarrow 64 \cdot 4,096 = 262,144ms \rightarrow 262144\mu s$
- *Prescaler 256* $\rightarrow 256 \cdot 4,096 = 1,048s \rightarrow 1048,576ms \rightarrow 1048576\mu s$
- *Prescaler 1024* $\rightarrow 1024 \cdot 4,096 = 4,194s \rightarrow 4194,304ms \rightarrow 4194304\mu s$

Ao utilizar como fonte de clock para o timer a mesma fonte do μC , a cada ciclo de máquina ocorre um incremento do TCNT1. Com isso em mente, para calcular o valor a ser carregado no TCNT1, deve-se observar os valores que podem ser gerados utilizando cada prescaler, conforme acima, e decidir qual o mais apropriado para o tempo que se pretende gerar. Após decidir qual prescaler será utilizado, ou mesmo se não será necessário utilizá-lo, deve-se carregar o TCNT1 com o valor conforme a seguir:

$$\begin{cases} \text{Tempo máximo que o prescaler selecionado pode gerar} = x \\ \text{Tempo que se pretende gerar} = y \\ \text{ciclo de máquina} = 62,5ns \text{ ou } 0,0625\mu s \text{ ou } 0,0000625ms \end{cases}$$

$$Valor = \frac{\frac{x - y}{\text{ciclo de máquina}}}{\text{prescaler}}$$

Timer	Tamanho	Interrupção	Funções Arduino UNO
TIMER0	8 bits (0 – 255)	- Compare Match - Overflow	- delay(), millis(), micros() - analogWrite() pinos 5, 6
TIMER1	16 bits (0 – 65535)	- Compare Match - Overflow - Input Capture	- Funções da biblioteca servo - analogWrite() pinos 9, 10
TIMER2	8 bits (0 – 255)	- Compare Match - Overflow	- tone() - analogWrite() pinos 3, 11

Deve-se observar as funções que cada Timer possui na programação com linguagem arduino (tabela acima) para evitar possíveis conflitos.