



TIMER

EJEMPLOS MICROLADDER



EDICION EDITADA PARA NANO LADDER |  
**DAGEL SISTEMAS ELECTRONICOS**

## 1 Ejemplo de retardo a la conexión.

En el siguiente ejemplo pretendemos conseguir los siguientes objetivos, cuando se mantenga presionado un pulsador:

- Que una bocina avise sobre el encendido inminente de un motor de extracción
- Que al mantener 3 segundos presionado el pulsador se active el motor de extracción.
- Que la bocina y el motor cesen su actividad cuando se suelte el pulsador.

### 1.1 Introducción a la función TON.

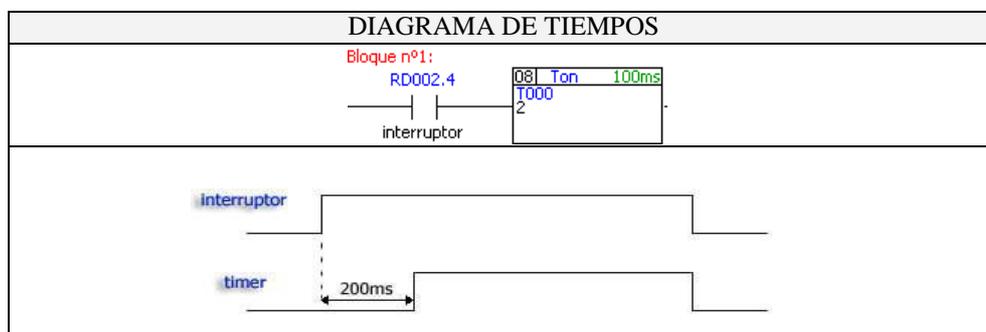
Esta función activa un temporizador con retardo, pasado ese retardo el temporizador se pone a “1” hasta que la señal de entrada que lo activa baje a “0”.

Cuadro de dialogo de la función “TON”

FUNCION “TON”	
Tipo	Base de tiempos del temporizador: (1, 10, 100, 1000)ms.
Flanco	Sin Flanco

OPERANDOS	
Tipo	Temporizador que se va a utilizar
Retardo	Es el factor de multiplicación de la base de tiempos, puede ser un literal (2, 30, 50 etc...) o el contenido de una variable INT8U que hayamos definido (RD025, etc...)

COMPORTAMIENTO DEL TEMPORIZADOR	
-	El tiempo total de retardo es: (base de tiempos x retardo).
-	Ej: (100ms x 2) = 200ms = 0,2 seg.



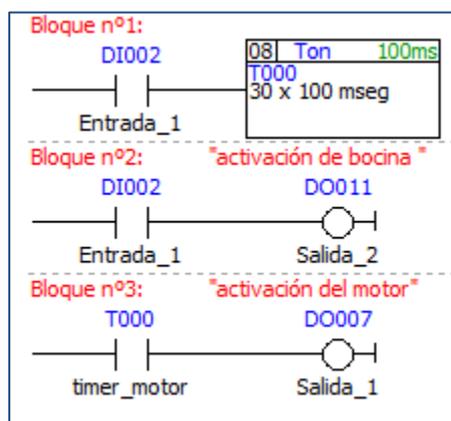
Para este Ejemplo deberemos tener la siguiente configuración.

Pin	Configuración	Función
P02	Entrada Digital (DI)	Entrada1: Pulsador
P07	Salida Digital (DO)	Salida1: Motor
P11	Salida Digital (DO)	Salida2: Bocina

Timer	Configuración	Función
T000	Ton/ 30 X 100 mseg	Timer a la Conexion

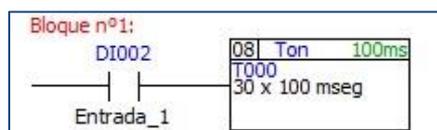
## 1.2 Programa

De esta manera el aspecto final del programa realizado para tal función sería el siguiente:



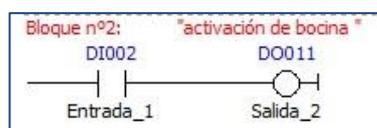
*Programa de encendido temporizado de un motor  
Y activación de una bocina.*

En el primer bloque la función “TON” se ejecuta cuando se presiona el pulsador asociado a la Entrada 1, poniendo su estado a 1.

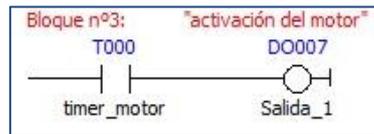


Cuando se cumple dicha condición, la función “TON” activa el temporizador, espera 3 segundos (retardo x base de tiempos) y se pone a “1”, hasta que se deje de presionar el pulsador

En el siguiente bloque asociamos el estado del pulsador directamente a la salida digital 2 (DO011) que controla el estado de la bocina. Así mientras que se mantenga presionado el pulsador la bocina estará sonando, con lo que se cumple la primera función del programa



En el siguiente bloque asociamos el T000 a través de un contacto a la salida digital 1 (DO007) que controla el estado del motor de extracción.



De esta manera cuando mantengamos presionado el pulsador durante 3 segundos, el temporizador se pondrá a "1" y por lo tanto el contacto asociado al temporizador también se pondrá a "1", activando el motor de extracción, con lo que conseguimos el segundo objetivo del programa.

Por último, cuando soltemos el pulsador, el timer volverá a ser "0", y el motor cesará su actividad. La bobina al estar asociada directamente al pulsador, cuando este deje de ser presionado, también se desactivará.

### Observaciones

Los temporizadores debido al ciclo de scan llevado a cabo por el programa y a la base de tiempos utilizada tienen el siguiente margen de error:

El margen de error está comprendido  
entre 0 y [ciclo de scan del programa + (1 x la base de tiempos)]

Por ejemplo, el Tiempo total con el mayor margen de error posible de un temporizador de 25 segundos, con una base de tiempos de 100ms es:

$$T.\text{Total timer} = \text{tiempo del timer} - \text{margen de error}$$

$$T.\text{Total timer} = 25\text{seg} - (\text{ciclo de scan del programa} + (1 \times \text{la base de tiempos} ((100\text{ms})))$$

$$T.\text{Total timer} = 25\text{seg} - (5\text{ms (aprox)} + (100\text{ms}))$$

$$T.\text{Total timer} = (25000 - 105)\text{ms} = 24895\text{ms}.$$

Como en este caso la temporización es tan solo de 3 segundos, no conviene escoger una base de tiempos de 1000 ms, ya que el margen de error sería aproximadamente de  $\pm 1$  segundo. Escogeremos una base de tiempos de 100 ms, y un factor de multiplicación (campo: "retardo") de 30.