

Práctica 5 a): i) Determina el tiempo máximo que se puede temporizar con el *Timer 0*, teniendo en cuenta que la frecuencia de reloj (F_{osc}) del microcontrolador PIC18F452 es de 8MHz y que durante la temporización sólo se puede producir 1 desbordamiento del *Timer 0*.

tiempo máximo (modo 8 bits) = ???

tiempo máximo (modo 16 bits) = ???

ii) Repite el apartado anterior, pero ahora con una frecuencia de reloj de 10MHz. ¿Por qué los tiempos máximos son diferentes a los obtenidos en el apartado *i*)?

Práctica 5 b): Escribe un programa que, ejecutado por el μC en el circuito de la página siguiente, haga que se cumpla lo siguiente cada 100 mseg. (a es una entrada, m es una salida y Q es una variable que tienes que declarar tu en el código):

_ si $Q = 0$ y $a = 0$: se deja $Q = 0$ y se pone $m = 0$

_ si $Q = 0$ y $a = 1$: se pone $Q = 1$ y se deja $m = 0$

_ si $Q = 1$ y $a = 1$: se dejan $Q = 1$ y $m = 0$

_ si $Q = 1$ y $a = 0$: se ponen $Q = 2$ y $m = 1$

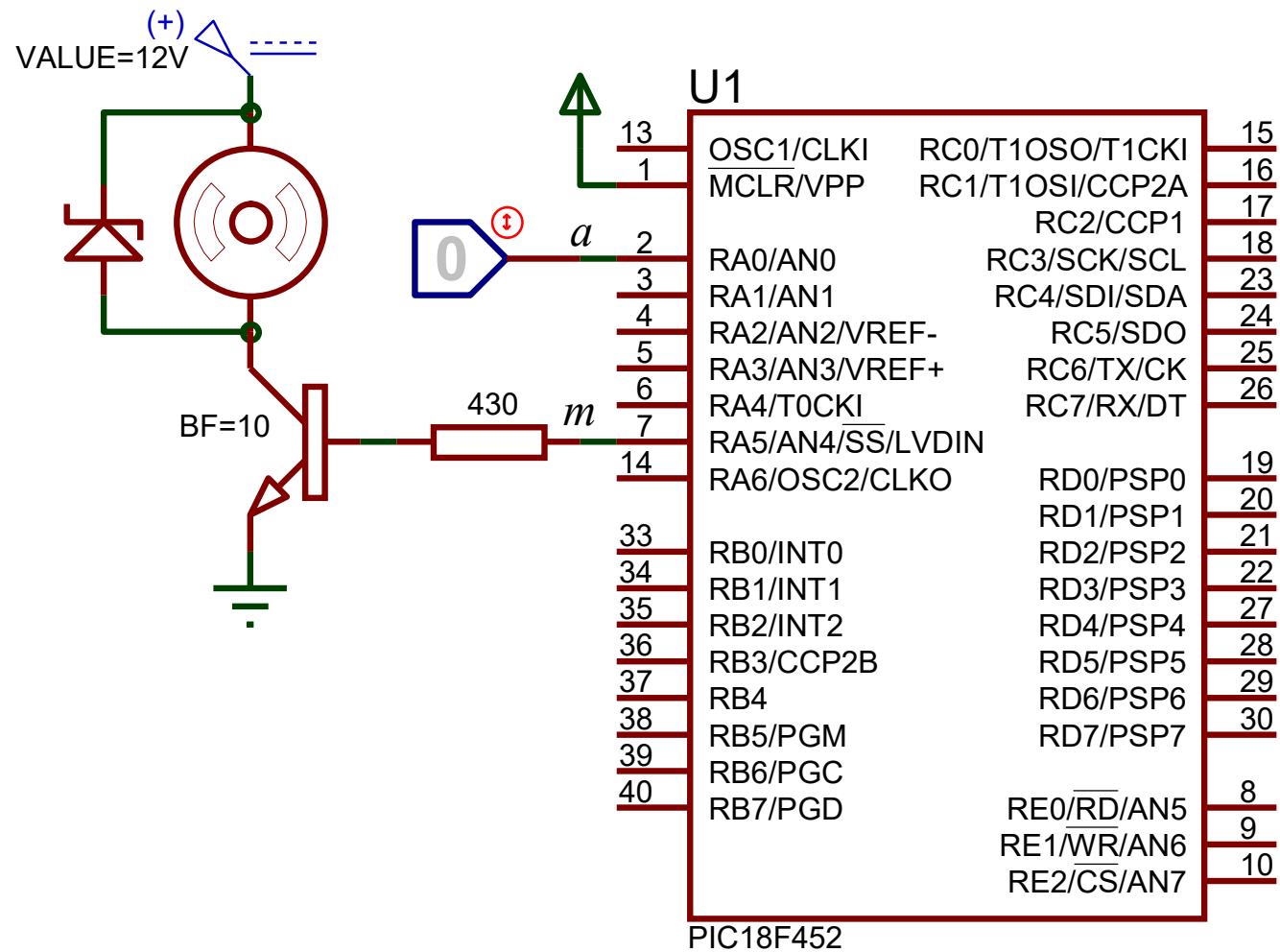
_ si $Q = 2$ y $a = 0$: se dejan $Q = 2$ y $m = 1$

_ si $Q = 2$ y $a = 1$: se pone $Q = 3$ y se deja $m = 1$

_ si $Q = 3$ y $a = 1$: se dejan $Q = 3$ y $m = 1$

_ si $Q = 3$ y $a = 0$: se ponen $Q = 0$ y $m = 0$

No puedes utilizar las funciones `delay_ms()` y `delay_us()`.



Componentes ISIS: PIC18F452, DIODE-SC, NPN, MOTOR, RES, LOGICTOGGLE.

Edit Component ? X

Part Reference:

Nominal Voltage:

Element:

Nominal Revs:

Load Resistance:

Other Properties:

☐ Exclude from Simulation
☒ Exclude from PCB Layout
☐ Exclude from Current Variant

☐ Attach hierarchy module
☐ Hide common pins
☐ Edit all properties as text

Configuración MOTOR

Edit Component ? X

Part Reference:

Part Value:

Element:

Simulator Primitive Type:

Other Properties:

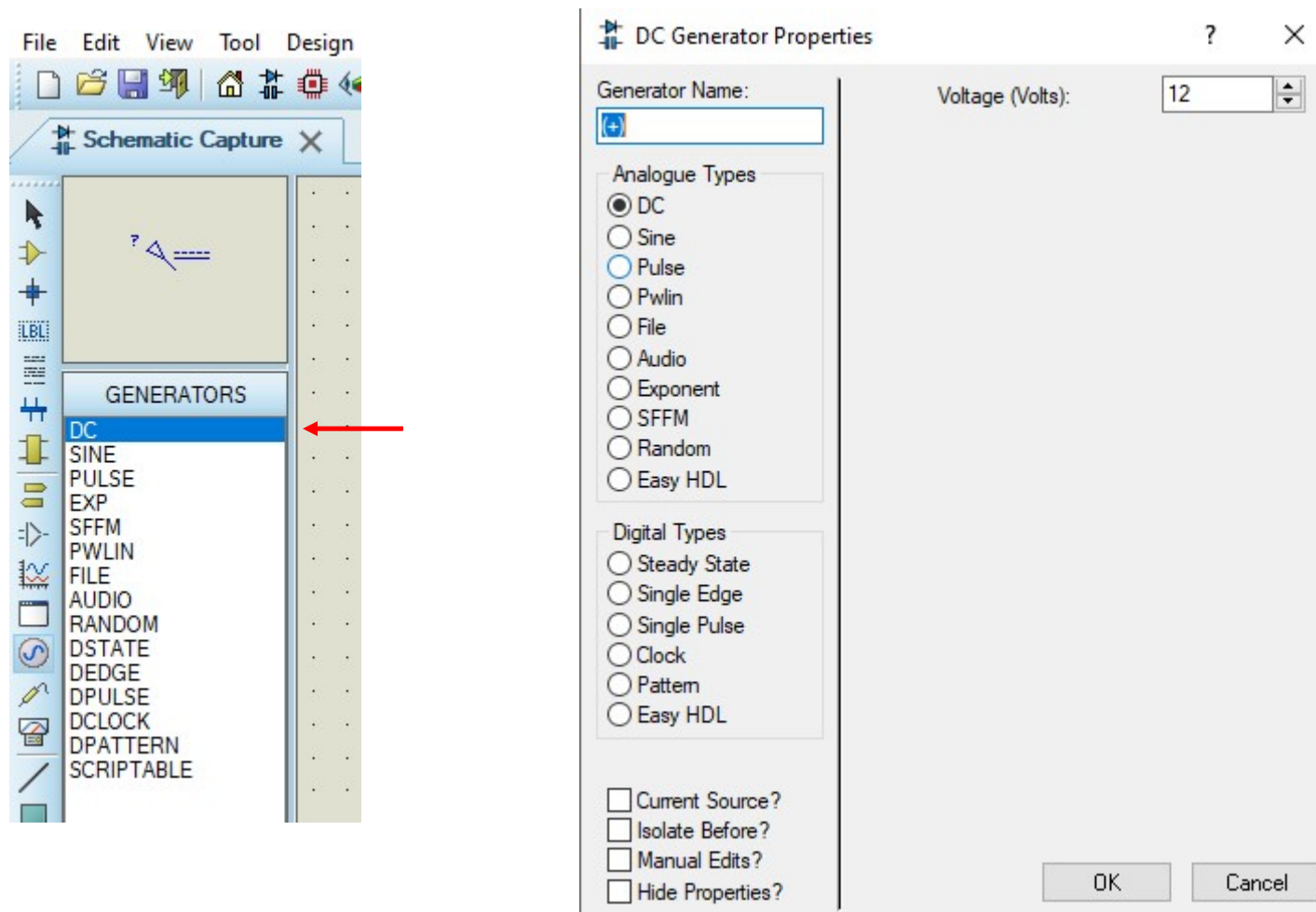
BF=10

☐ Exclude from Simulation
☐ Exclude from PCB Layout
☐ Exclude from Current Variant

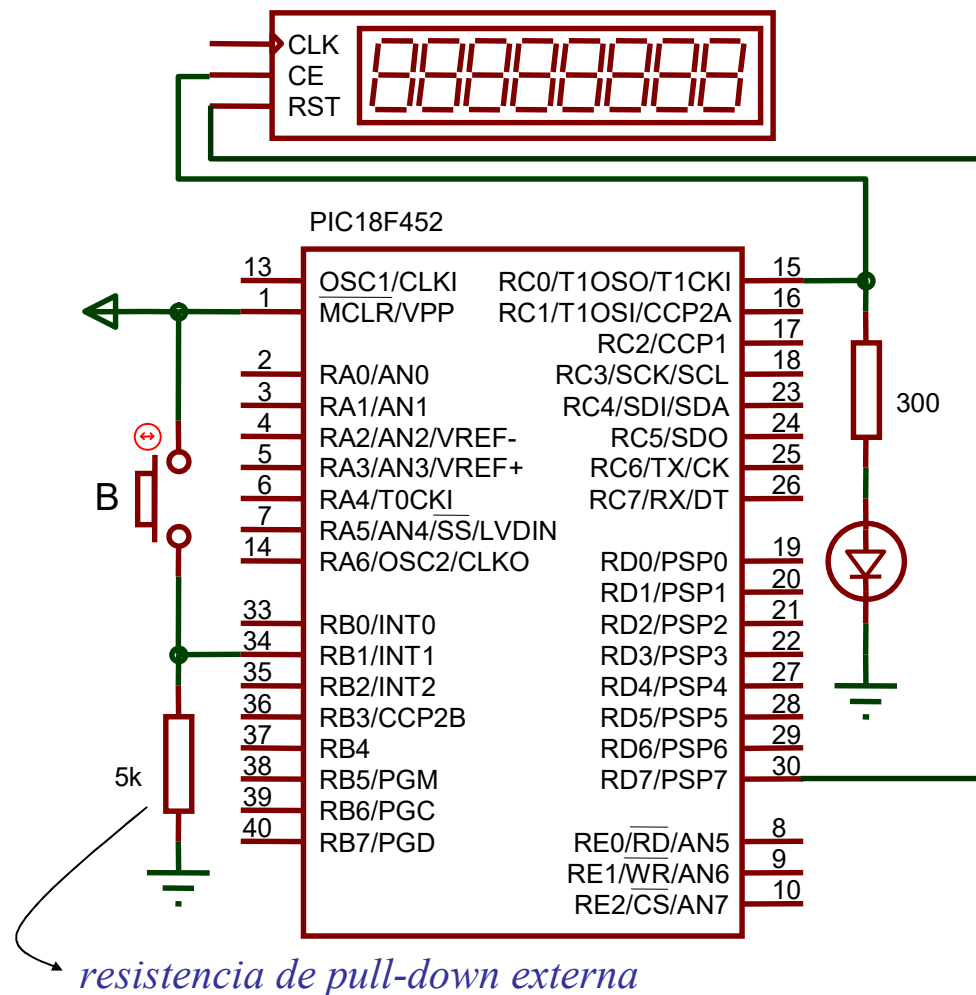
☐ Attach hierarchy module
☐ Hide common pins
☐ Edit all properties as text

Configuración transistor NPN

Configuración fuente de tensión de 12V



Práctica 5 c): Estando el *led* apagado, si se presiona el botón B, el *led* debe de encenderse, permaneciendo en dicho estado durante 12 segundos. No puedes utilizar las funciones `delay_ms()` y `delay_us()`. Si se pulsa el botón estando el *led* encendido debe de ignorarse dicha acción. Componentes ISIS: PIC18F452, RES, Button, LED-BLUE, COUNTER TIMER



Algo en lo que pensar: ¿es posible realizar una temporización de 12 segundos con el *Timer 0* sin que se produzca un desbordamiento?

¿seguro?

Nota: el *Counter Timer* es un aparato, no un componente electrónico. Por defecto, su *reset* se activa con un flanco de subida.

Nota: no pongas nada en el bucle infinito