

(66.09) Laboratorio de microcomputadoras

Proyecto:

TP6 - Timers

Profesor:	Ing. Guillermo Campiglio
Cuatrimestre / Año:	1c/2020
Turno clases prácticas:	Miércoles
Jefe de trabajos prácticos:	Pedro Ignacio Martos
Docente guía:	Pedro Ignacio Martos

Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Mauro Fabricio	Toscano,Go nnella	96890										

Observaciones:

Fecha de aprobación		

Firma JTP

Coloquio	
Nota final	
Firma profesor	

Índice

[Objetivo del Trabajo](#)

[Descripción del trabajo](#)

[Diagrama de conexiones en bloques](#)

[Circuito esquemático](#)

[Listado de componentes](#)

[Flujo del programa](#)

[Código de programa](#)

[Código](#)

[Resultado](#)

[Conclusiones](#)

Objetivo del Trabajo

Cambiar la frecuencia del parpadeo de un led usando timers

Descripción del trabajo

Se utilizará un Arduino con un procesador ATmega 2560. Se conectarán 1 par led resistencia al PC0 y dos switches de un funduino joystick al PD0 y PD1 (Int 0 e Int 1, digital pin 20 y 21).

Para cambiar la frecuencia de parpadeo, se utilizará el timer 1 de 16 bits, mediante interrupciones generadas por el overflow del mismo. Además, se utilizará el timer 3 para una rutina de delay que evite detectar rebotes.

En detalle, las frecuencias de parpadeo serán dadas por el preescaler en 64 si se apreta el primer interruptor, en 256 si se apreta el segundo, y en 1024 si se aprietan ambos. De no apretar ningún botón, se mantendrá prendido el led. Para el timer 3 que usamos para evitar rebotes, usaremos el preescaler 8 para un funcionamiento normal, y el preescaler en 1024 para hacer pruebas y verificar que ande.

Teniendo en cuenta que la frecuencia del procesador es de 16 MHz, y que el timer 1 tiene 16 bits, podemos calcular el tiempo que está prendida (o apagada) la luz como:

$$t = \frac{preEscaler}{16 * 10^6} * 2^{16} segundos$$

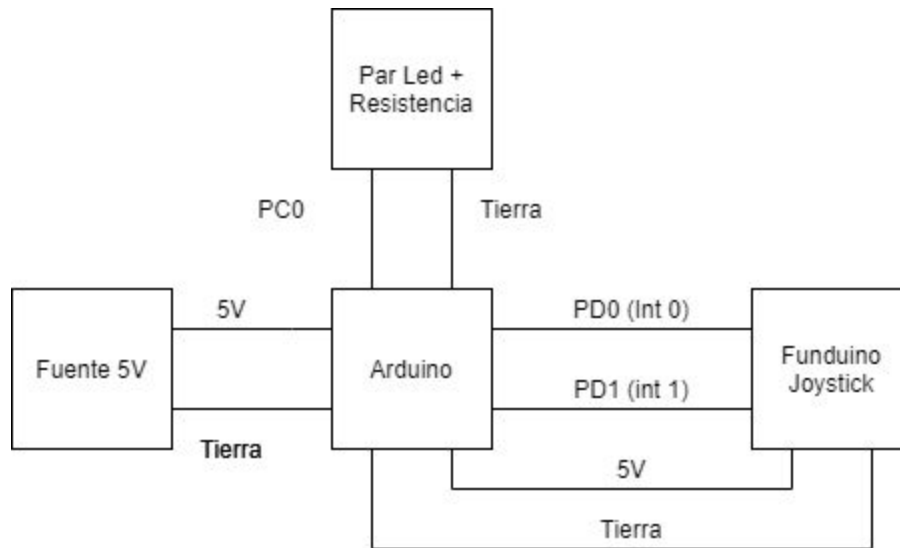
Y la frecuencia como:

$$f = 1/2t$$

Entonces, tenemos los siguientes tiempos:

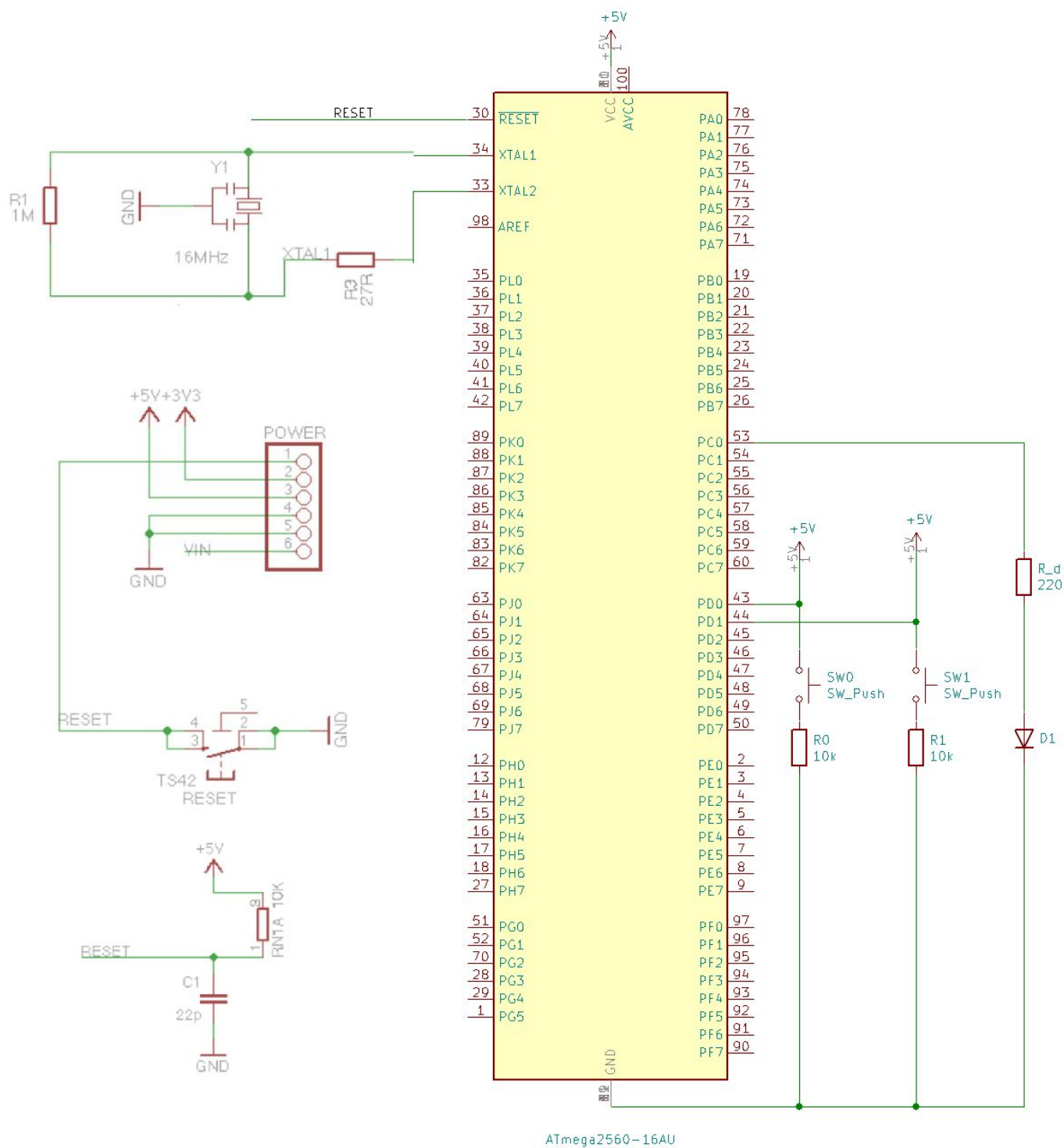
Pre escaler	t	f = 1 / 2t
8	~ 32 ms	No aplica
64	~ 262 ms	~ 1.9 hz
256	~ 1.05 s	~ 0.333 hz
1024	~ 4.2 s	~ 0.119 hz

Diagrama de conexiones en bloques



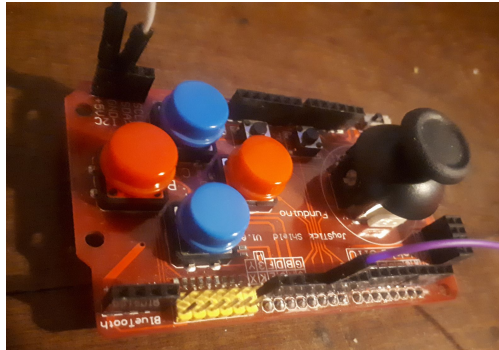
Circuito esquemático

Nota: El circuito del funduino desconocido, lo reduzco a dos switches puesto con resistencias de pull up. Lógicamente funciona igual.

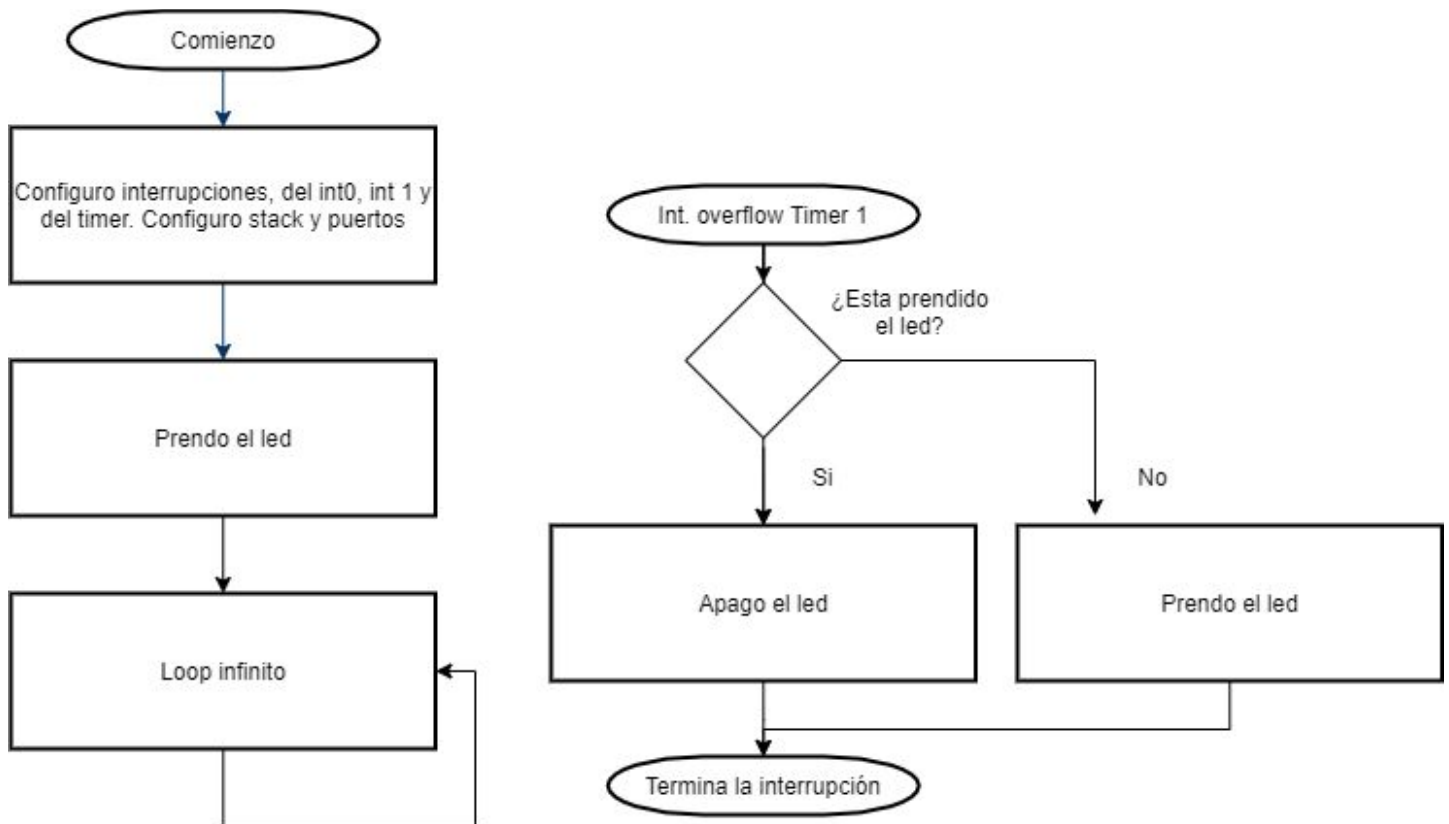


Listado de componentes

- Arduino Mega 2560 (Ya se tenía, aproximadamente \$1700)
- Resistencia de 220 ohms (Paquete de 10 por 51\$)
- Led 5mm (Paquete de 10 por 63\$)
- Protoboard (Ya se tenía, aproximadamente 250\$)
- Cables (Paquete de 40 macho macho, y 40 hembra hembra 197\$ cada uno)
- Funduino joystick shield V1.A (400\$):

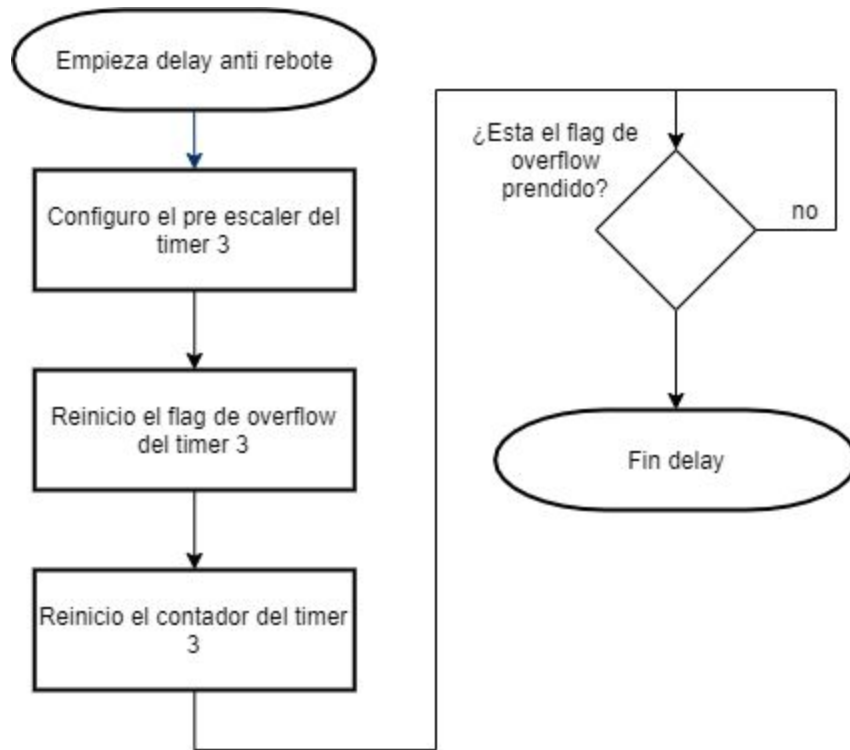


Flujo del programa





Adicionalmente, el delay usado para evitar los rebotes funciona así:



Código de programa

[Codigo](#)

Resultado

El led cambia la frecuencia a la que parpadea de acuerdo a lo esperado. Además, configurando el preescaler del timer 3 en 1024 se verifica visualmente el delay. Entonces con el preescaler en 8 se evitaría que los rebotes lleguen al programa y el comportamiento del dispositivo no cambiaría para el usuario.

Conclusiones

Los timers nos sirven para realizar acciones cada tiempos periódicos. Podemos hacerlas aprovechando interrupciones generadas cuando el timer llega a un overflow, o mediante polling del mismo. Finalmente, mediante el uso del prescaler, podemos ajustar la velocidad a la que cuentan, de acuerdo a lo que necesitamos.