

Tareas

Arduino no posee un sistema operativo y tampoco incorpora ningún otro sistema que gestione la multitarea como estamos acostumbrados en otros entornos de programación (por ejemplo en Scratch), el programa se ejecuta directamente por el microcontrolador y somos nosotros en el propio programa los que tenemos que gestionar la manera de ejecutar varias tareas simultáneamente intentando simular un sistema multitarea.

La teoría de un sistema multitarea es dividir los procesos en pequeños bloques, cada uno de estos bloques debe realizar un pequeño trabajo (en poco tiempo) y dejar paso al siguiente sin bloquear el funcionamiento. Al realizar este proceso de forma continua y a mucha velocidad el resultado final es que todos los bloques se ejecutan simultáneamente.

¿Qué debemos evitar dentro de los bloques de las tareas?

- Bloques de tipo “esperar”
- Bucles o repeticiones muy largas
- Condiciones que bloqueen la ejecución de la tarea

Para implementar un sistema sencillo de tareas utilizaremos el bloque “tiempo transcurrido”. Este bloque nos permite obtener el tiempo en ms que ha pasado desde el inicio del programa (reset).

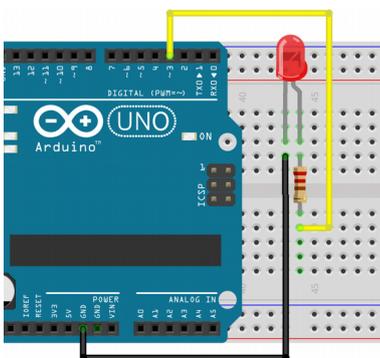
Tiempo transcurrido (milisegundos)

Tareas - 1

Parpadeo led cada 1s

CÓDIGO DE PROYECTO:

El programa comprobará si han pasado 1000 ms (o más) desde la última ejecución de la tarea, si es así ejecutará el código de encender/apagar el led.



```

Inicializar
  Establecer estado del led = Off
  Establecer ultimo tiempo = Tiempo transcurrido (milisegundos)

Bucle
  Establecer tiempo actual = Tiempo transcurrido (milisegundos)
  Establecer diferencia de tiempo = tiempo actual - ultimo tiempo
  si diferencia de tiempo >= 1000
  hacer
    Establecer ultimo tiempo = Tiempo transcurrido (milisegundos)
    si estado del led
    hacer
      Establecer estado del led = Off
      Escribir digital Pin 3 OFF
    sino
      Establecer estado del led = On
      Escribir digital Pin 3 ON
  
```

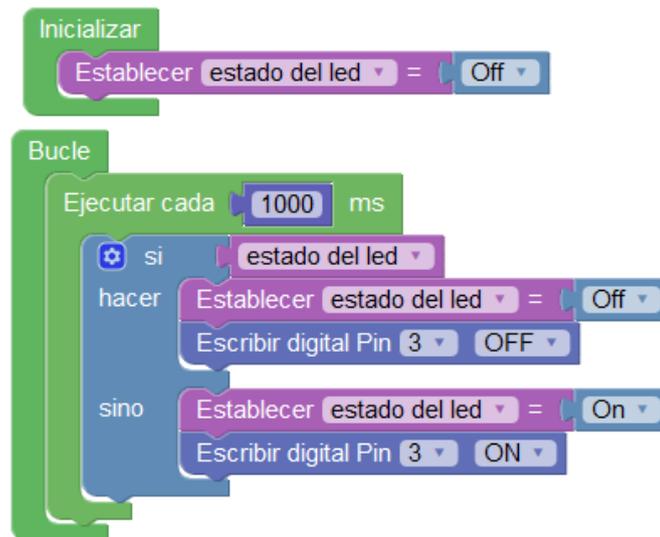
Tareas - 2

Bloque de tareas simplificado

CÓDIGO DE PROYECTO:

Todo el programa anterior de la práctica *Tareas-1* se puede simplificar con un bloque que incorpora ArduinoBlocks llamado “*ejecutar cada*”, donde le especificamos cada cuanto tiempo se deben ejecutar los bloques contenidos en su interior.

Programa equivalente a la práctica *Tareas-1*



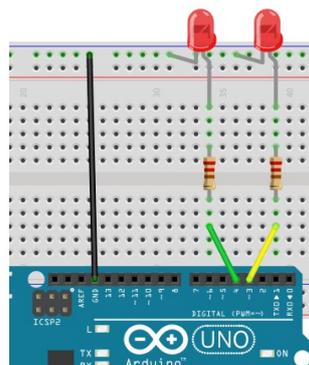
Tareas - 3

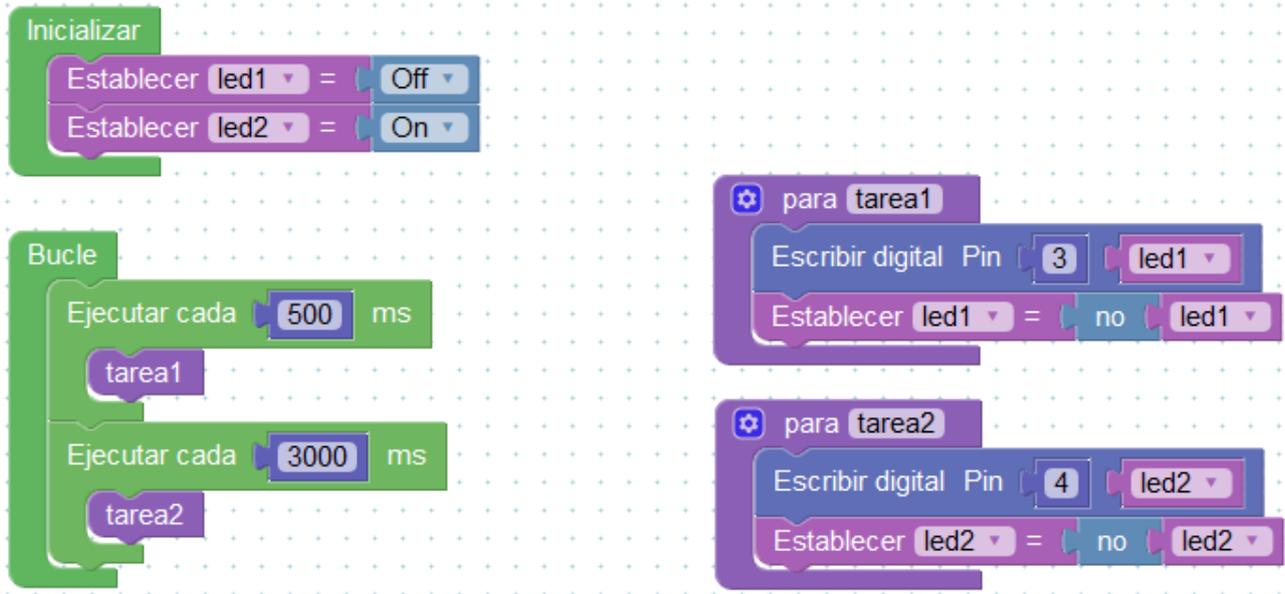
Dos tareas a la vez

CÓDIGO DE PROYECTO:

Utilizando bloques “*ejectura cada*” vamos a realizar dos tareas simultáneas

- Tarea 1 (cada 500ms): hará parpadear un led conectado al pin 3 cada 500 ms
- Tarea 2 (cada 3000ms): hará parpadear un led conectado al pin 4 cada 3000 ms



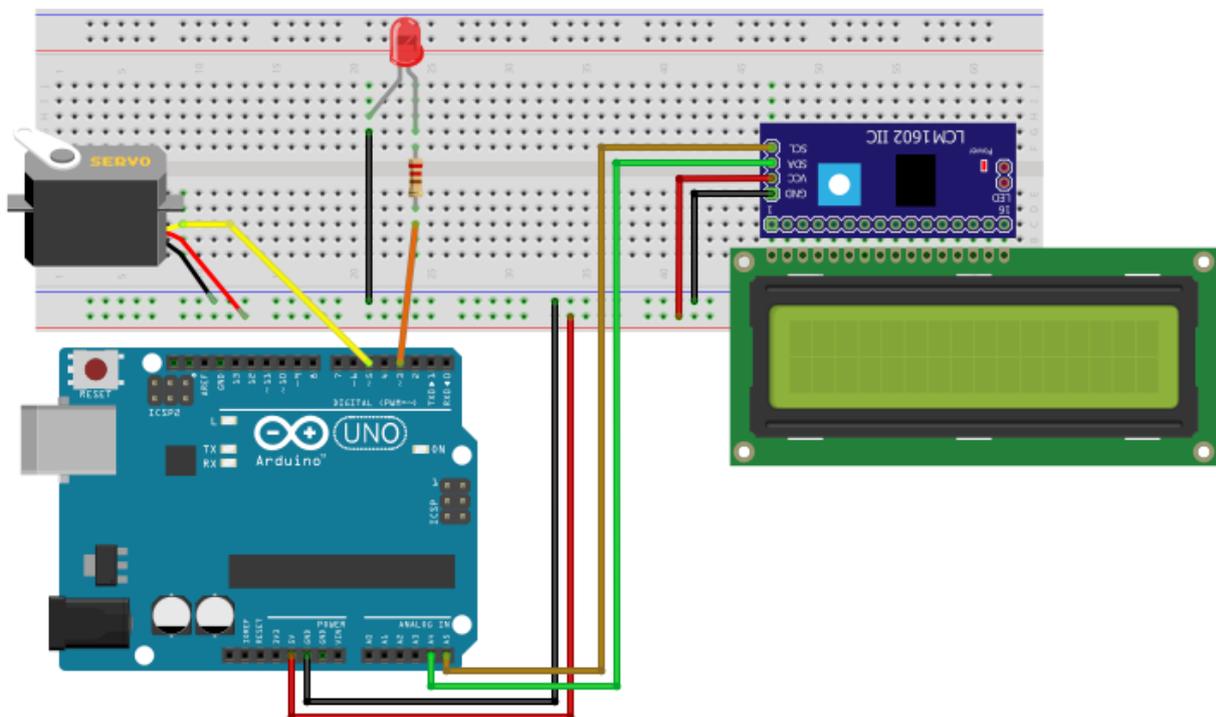


Tareas - 4

Tres tareas a la vez

CÓDIGO DE PROYECTO:

Conecta la pantalla LCD con el módulo i2c, un led al pin 3 y un servo al pin 5.



El programa debe realizar 3 tareas:

- Tarea 1: Cada 250 ms parpadea el led
- Tarea 2: Cada 500 ms el servo se mueve 30 grados
- Tarea 3: Cada 1000 ms en la pantalla se muestra el valor de un variable que va aumentando

```

Inicializar
  LCD iniciar (i2c)
  establecer led encendido a 0
  establecer posicion servo a 0
  establecer contador a 0

Bucle
  Ejecutar cada 250 ms
    tarea1
  Ejecutar cada 500 ms
    tarea2
  Ejecutar cada 1000 ms
    tarea3

para tarea1
  si led encendido = 0
  hacer
    establecer led encendido a 1
    Escribir digital Pin 3 ON
  sino
    establecer led encendido a 0
    Escribir digital Pin 3 OFF

para tarea2
  Servo Pin 5 Grados posicion servo Retardo (ms) 0
  establecer posicion servo a posicion servo + 10
  si posicion servo > 180
  hacer
    establecer posicion servo a 0

para tarea3
  LCD limpiar
  LCD imprimir Columna 0 Fila 0 "Tarea 3:"
  LCD imprimir Columna 0 Fila 1 Número entero contador
  establecer contador a contador + 1
  
```



Modifica el tiempo de cada tarea en el programa anterior...