

Guía de Ejercicios Prácticos 1

“Manejo de GPIO e Interrupciones”

Objetivos

- Manejo del módulo GPIO del microcontrolador RP2040.
- Control de los periféricos utilizando pooling e interrupciones.
- Manejo de componentes electrónicos básicos como leds, resistencias, displays 7 segmentos, capacitores y transistores.
- Utilización de conocimientos previos.
- Desarrollo de la capacidad de interpretar la documentación relacionada al hardware y correspondiente a las librerías de software utilizadas.

Funciones útiles para esta guía

A continuación se listan algunas funciones útiles del Software Development Kit de Raspberry. Esta lista no es exhaustiva, por lo tanto se recomienda consultar las restantes. Para saber más sobre el manejo de estas funciones debe consultar la [documentación](#). (Como dato orientativo el nombre de las funciones empieza siempre con el módulo al que pertenece)

Inicialización de los pines

```
void gpio_init (uint gpio)
void gpio_init_mask (uint gpio_mask)
void gpio_set_function (uint gpio, gpio_function_t fn)
void gpio_set_function_masked (uint32_t gpio_mask, gpio_function_t fn)
```

Escritura

```
static void gpio_put (uint gpio, bool value)
static void gpio_put_masked (uint32_t mask, uint32_t value)
static void gpio_xor_mask (uint32_t mask)
static void gpio_clr_mask (uint32_t mask)
static void gpio_set_mask (uint32_t mask)
```

Lectura

```
static bool gpio_get (uint gpio)
void gpio_set_pulls (uint gpio, bool up, bool down)
static void gpio_pull_up (uint gpio)
static void gpio_pull_down (uint gpio)
```

Interrupciones

```
void gpio_set_irq_enabled (uint gpio, uint32_t event_mask, bool enabled)
void gpio_set_irq_callback (gpio_irq_callback_t callback)
void gpio_set_irq_enabled_with_callback (uint gpio, uint32_t event_mask, bool enabled,
gpio_irq_callback_t callback)

void irq_set_enabled (uint num, bool enabled)
void irq_set_priority (uint num, uint8_t hardware_priority)
```

Delays

```
void sleep_ms (uint32_t ms)
```

```
void sleep_us (uint64_t us)
```

Ejercicio 1 “Blinking”

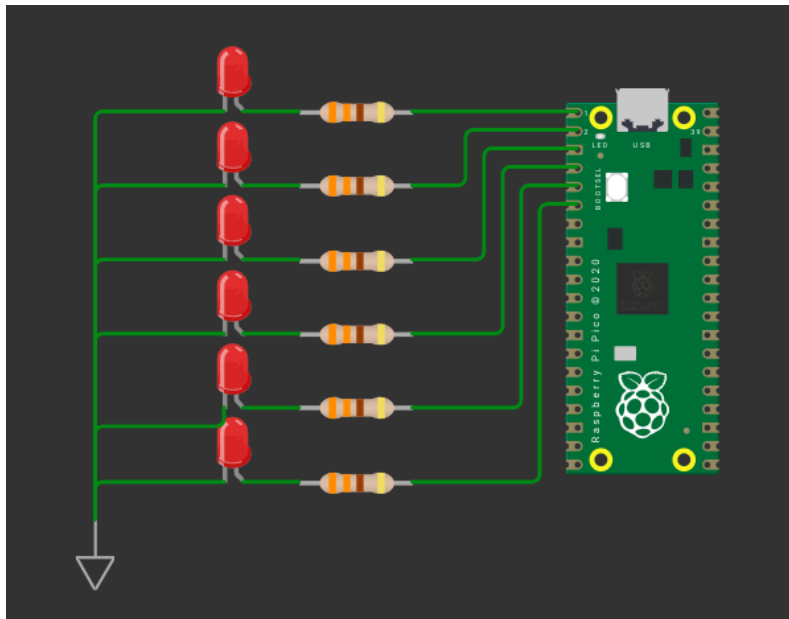
Implemente un proyecto donde se haga parpadear un led (puede ser el propio de la Raspberry Pico) a una frecuencia de 1[Hz].

Ejercicio 2 “Auto fantástico”

En este proyecto el alumno deberá implementar un circuito para replicar el funcionamiento de las luces del mítico “[Auto fantástico](#)” utilizando la placa de desarrollo Raspberry Pico.

Circuito propuesto

El circuito debe ser similar al que se observa a continuación:

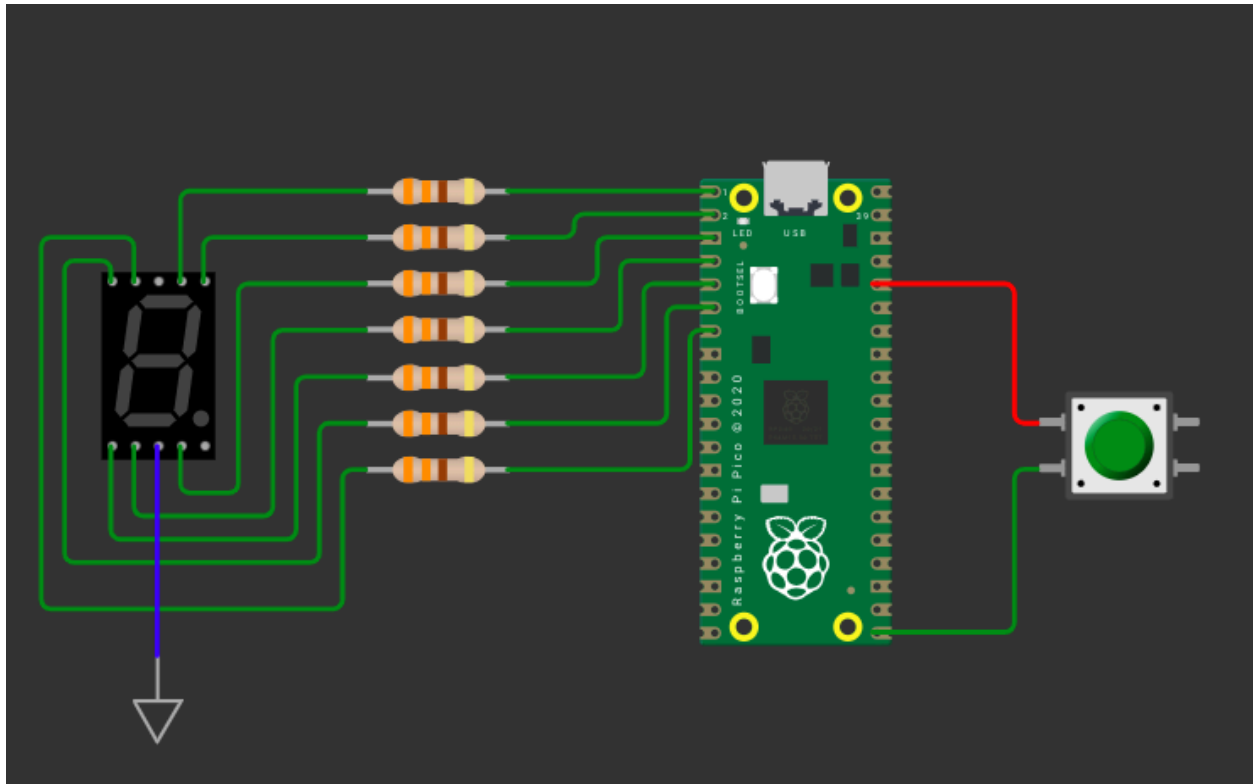


Revise el circuito porque no está completamente bien.
Compruebe que los pines de la placa tengan las salidas del periférico que necesita.
Recalcule las resistencias. En la imagen son de 330[ohm], pero puede necesitar resistencias mas grandes para limitar más la corriente.

Ejercicio 3 “Contador”

En este proyecto se busca implementar un contador de pulsos, el circuito deberá tener un pulsador y un display de 7-segmentos, de forma que el software cuente la cantidad de veces que se presione el pulsador y se muestre el valor en el display.

Circuito propuesto

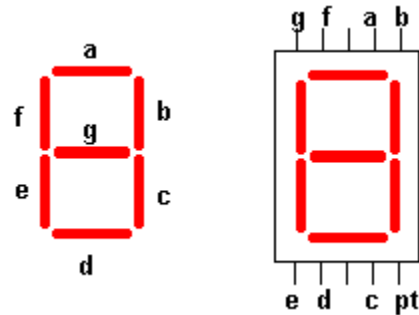


Revise el circuito porque no está completamente bien.
Compruebe que los pines de la placa tengan las salidas del periférico que necesita.
Recalcule las resistencias. En la imagen son de 330[ohm], en este caso va a necesitar resistencias mas grandes ya que cuando estén prendidos todos los segmentos del display la corriente superará a la máxima soportada por el puerto GPIO de la placa.

Algunas consideraciones

Las conexiones del display de 7-segmentos se pueden ver en la siguiente imagen. A la izquierda la identificación de los segmentos y a la derecha los correspondientes pines (el pin “pt” corresponde al punto decimal que suelen tener éstos display), los pines centrales corresponden

a la conexión común (estos serán los cátodos, si el display es cátodo-común o los ánodos si el display es ánodo-común).



En la siguiente imagen se observa una captura de pantalla de la hoja de datos del microcontrolador RP2040, en ella se pueden observar las especificaciones eléctricas. En un recuadro rojo se resaltan las correspondientes a la corriente máxima que puede salir (source) y entrar (sunk) por la totalidad de los pines gpio del microcontrolador.

RP2040 Datasheet

Low @ IOVDD=1.8V					12mA depending on setting
Output Voltage Low @ IOVDD=2.5V	V_{OL}	0	0.4	V	$I_{OL} = 2, 4, 8$ or 12mA depending on setting
Output Voltage Low @ IOVDD=3.3V	V_{OL}	0	0.5	V	$I_{OL} = 2, 4, 8$ or 12mA depending on setting
Pull-Up Resistance	R_{PU}	50	80	k Ω	
Pull-Down Resistance	R_{PD}	50	80	k Ω	
Maximum Total IOVDD current	$I_{IOVDD,MAX}$		50	mA	Sum of all current being sourced by GPIO and QSPI pins
Maximum Total VSS current due to IO (IOVSS)	$I_{IOVSS,MAX}$		50	mA	Sum of all current being sunk into GPIO and QSPI pins

Table 625. USB IO characteristics

Parameter	Symbol	Minimum	Maximum	Units	Comment
Pin Input Leakage Current	I_{IL}		1	μ A	
Single Ended Input Voltage High	V_{SEH}	2		V	

Consideraciones de software

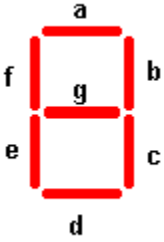
Respecto al pulsador se espera que se detecten las pulsaciones mediante pooling.

Recordar configurar las correspondientes resistencias de pull-down internas.

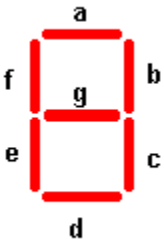
Tecnicatura Universitaria en Automatización y Robótica - Microprocesadores

Se propone la implementación de una “Lookup Table” para la conversión del dígito entero en los valores correspondientes que tienen que tomar los pines para la correcta representación del número.

A continuación se puede observar una tabla donde se muestran los valores que deben tomar los pines de un display 7-segmentos **ánodo común** para mostrar los símbolos.

	Símbolo	g	f	e	d	c	b	a
	0	1	0	0	0	0	0	0
	1	1	1	1	1	0	0	1
	2	0	1	0	0	1	0	0
	3	0	1	1	0	0	0	0
	4	0	0	1	1	0	0	1
	5	0	0	1	0	0	1	0
	6	0	0	0	0	0	1	0
	7	1	1	1	1	0	0	0
	8	0	0	0	0	0	0	0
	9	0	0	1	1	0	0	0

A continuación la tabla que corresponde para un display **cátodo común**. Observe que la diferencia es que los pines deben tomar valores negados respecto al caso anterior.

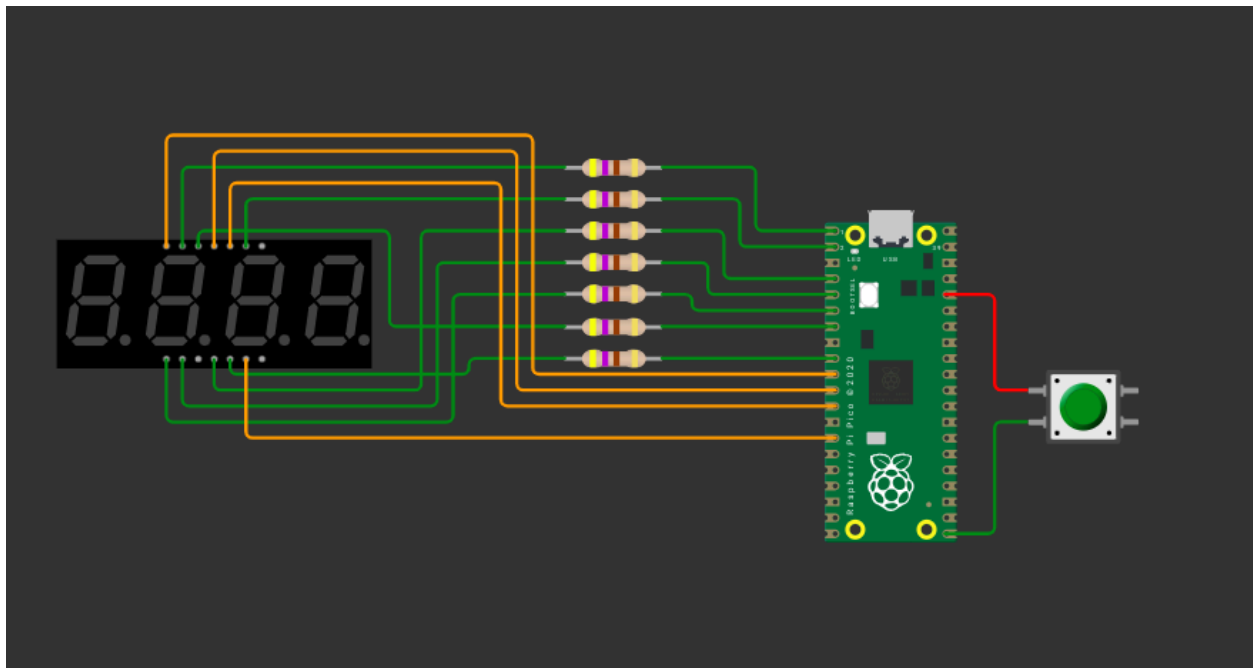
	Símbolo	g	f	e	d	c	b	a
	0	0	1	1	1	1	1	1
	1	0	0	0	0	1	1	0
	2	1	0	1	1	0	1	1
	3	1	0	0	1	1	1	1
	4	1	1	0	0	1	1	0
	5	1	1	0	1	1	0	1

	6	1	1	1	1	1	0	1
	7	0	0	0	0	1	1	1
	8	1	1	1	1	1	1	1
	9	1	1	0	0	1	1	1

Ejercicio 4 “Contador extendido”

Este proyecto es una extensión del proyecto anterior pero utilizando un display de 4 x 7-segmentos, el objetivo es el mismo, la diferencia radica en la cantidad de dígitos que se pueden mostrar.

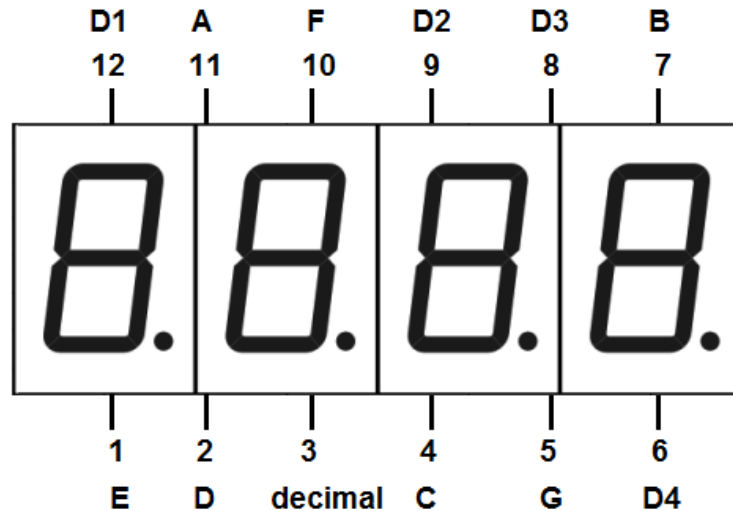
Circuito propuesto



Revise el circuito porque no está completamente bien.
Compruebe que los pines de la placa tengan las salidas del periférico que necesita.
Recalcule las resistencias. En la imagen son de 470[ohm], compruebe que este valor es adecuado calculando la corriente que entregarán los pines cuando estén prendidos todos los segmentos del display (tenga en cuenta el método a través del cual encenderá los displays).

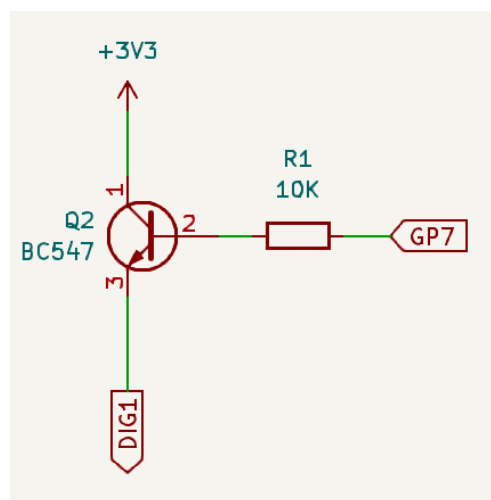
Algunas consideraciones

A continuación se puede observar el “pinout” de estos displays. En este caso los segmentos se identifican con las mismas letras que en el anterior y cada dígito tiene su propio “cátodo”(o “ánodo”) identificado con los pines D1, D2, D3 y D4 (correspondientes a los displays desde el de la izquierda hasta el de la derecha).



Para estos pines también hay que considerar la corriente necesaria para manejarlos, si son correctamente switcheados y sólo hay un display de 7-segmentos encendido a la vez la corriente debería ser similar a la del circuito implementado para el ejercicio anterior.

De todas maneras siempre es recomendable utilizar algún componente para manejar la corriente. Se sugiere algún circuito como el siguiente.



En este circuito el transistor se encuentra en “corte” mientras el pin GP7 del RP2040 está en 0, una vez que el pin es puesto a 1 la base del transistor es excitada produciendo que el transistor

Tecnicatura Universitaria en Automatización y Robótica - Microprocesadores

entre en saturación y conduzca toda la corriente que el circuito sea capaz de entregar (en este caso estará limitada por las resistencias que tienen los segmentos del display). Esto logra que la corriente no salga directamente del microcontrolador.

Consideraciones de software

Para este problema se espera que las pulsaciones sean detectadas mediante una interrupción por flanco ascendente.

Recordar configurar las correspondientes resistencias de pull-down internas.
--