

UNIVERSIDAD DE ORIENTE  
NÚCLEO ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN  
LABORATORIO DE PROYECTOS DIGITALES AVANZADOS



## **Display Siete Segmentos De Cuatro Dígitos**

Prof. Rene Cabrera

Elaborado por:

Brazon Eulises

Barcelona, 8 de febrero del 2023

## **ABSTRACTO**

A continuación, se estará realizando el uso del rasberripy pico para la utilización de un display siete segmentos, compuesto por 4 dígitos.

## **OBJETIVOS**

### **Objetivo General:**

- Utilizar un display siete segmentos de 4 dígitos

### **Objetivos Específicos**

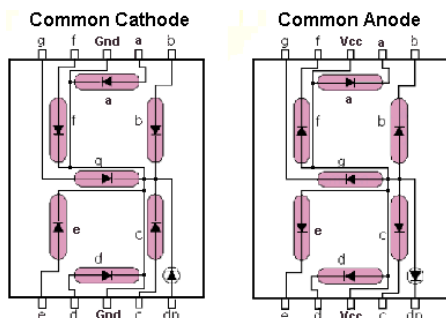
- Conectar un display siete segmentos
- Generar la codificación necesaria para poder mostrar los datos correctamente
- Generar una multiplexación para conectar 4 dígitos y un mismo bus de datos
- 

## **MATERIALES REQUERIDOS**

- Display siete segmentos de cuatro dígitos
- Cuatro transistores 2N2222
- Jumpers para la conexión
- Protoboard
- Resistencia  $220\Omega$  (8unidades) y  $1K\Omega$  (4unidades).
- Rasberripy
- Alimentación 5V.
- Herramientas para la manipulación y testeo (multímetro, pinzas, entre otros).

## DESCRIPCION DEL PROYECTO

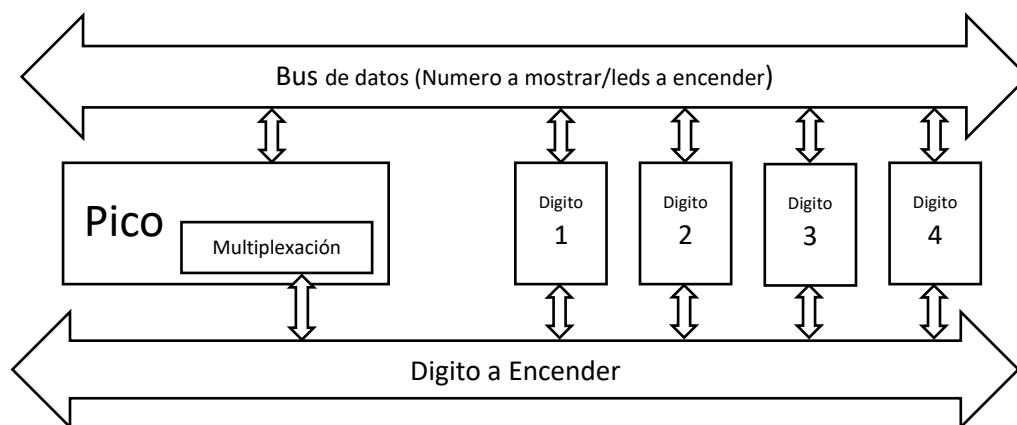
Mediante el uso del raspberripy pico se procederá a poner en funcionamiento un display de siete segmentos de 4 dígitos, para ello es necesario entender la lógica detrás de ello, un display siete segmentos en su interior posee siete leds dispuestos de tal manera, que al encender los leds indicados podremos visualizar el número deseado, esto poseen un pin en común para la alimentación, que bien



puede ser el ánodo, o el cátodo, es por ello que se conocen como ánodo común o cátodo común, son estos pines comunes los que se van a estar alimentando a través de un transistor, para controlar si el dígito a mostrar estará prendido o apagado.

Con el pin en común se controla el encendido del display, y para suministrar la información del dígito a mostrar debemos indicar a través de los pines a,b,c,d,e,f,g el estado lógico de cada led, como lo que se quiere realizar es manejar varios dígitos, el realizar todas estas conexiones de manera independiente para cada uno, ocuparía demasiados pines, para optimizarlos, se crea entonces un bus de datos, a donde van a estar conectados todos los dígitos, y se va encendiendo y apagando un dígito a la vez, en lo que se conoce como multiplexación, la transición entre el tiempo de encendido de cada dígito, si se hace a una velocidad muy alta, el ojo humano no alcanza a percibir estas variaciones, percibiendo así que todos los dígitos están encendido en todo momento.

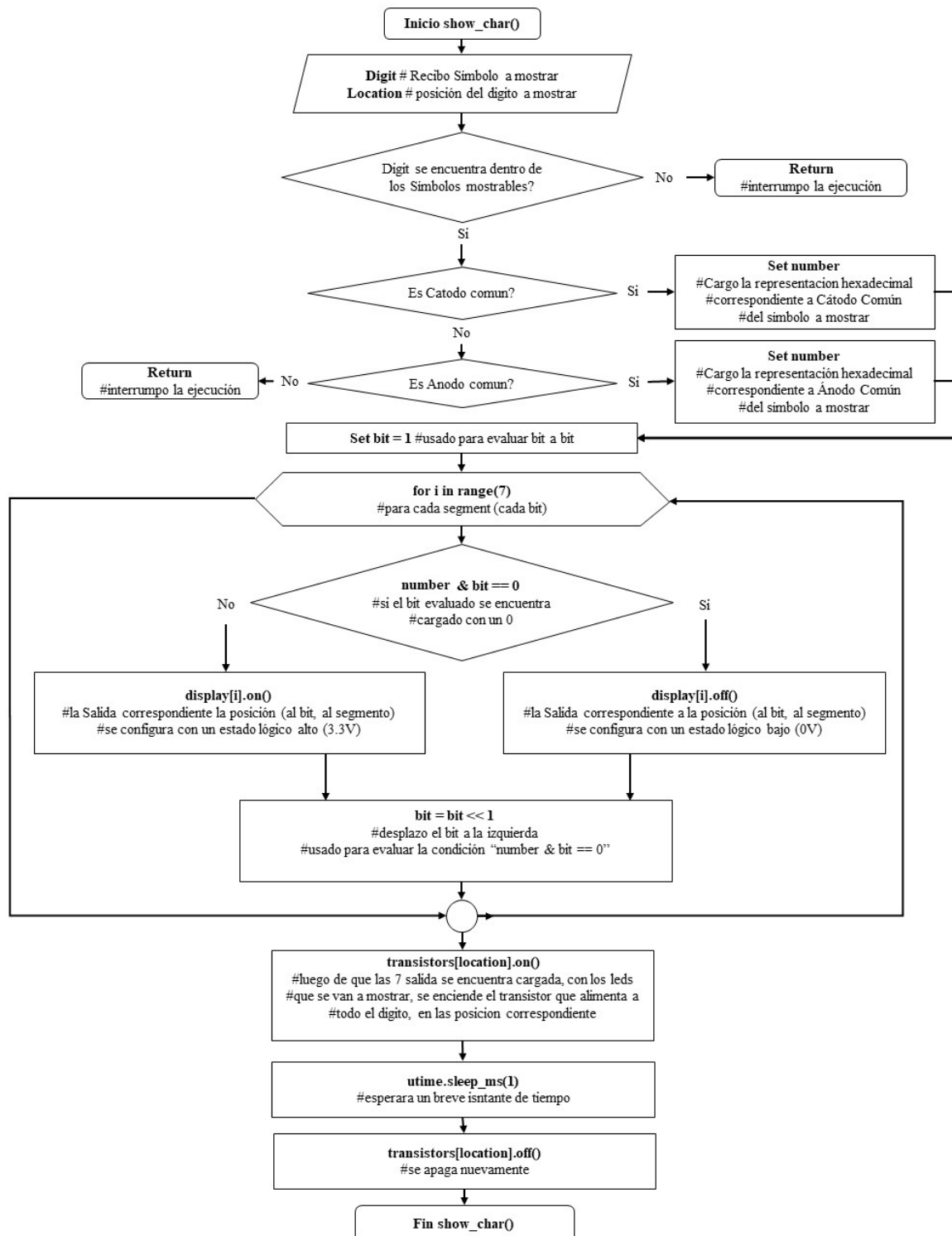
### Diagrama de estado:



En la utilización del código se implementó la librería `display7seg`, proporcionada por Sergio Castaño, que ayuda a la utilización de estos displays, donde parte de la simplificación del código radica en la representación de los leds a encender a través de dos números en hexadecimal, cuyo valores tienen relación con los led que se estarán encendiendo, debido a que cada número tiene su representación binaria, y cada uno de estos bit se pueden utilizar para indicar el estado de encendido o apagado de un led.

A esta librería, se le añadió un método adicional para agregar representaciones adicionales que no fueran solos números, el cual fue utilizado para mostrar un pequeño aviso de "HOLA" antes de realizar un conteo.

## Diagrama de flujo del método show\_char() añadido a la librería



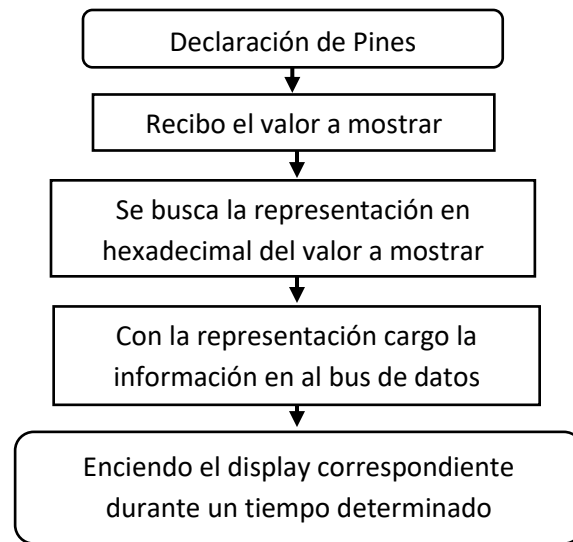
Código referente alojado en: <https://github.com/EulisesBrazon/display-7-segmentos>

**Tabla de las representaciones Hexadecimales agregadas:**

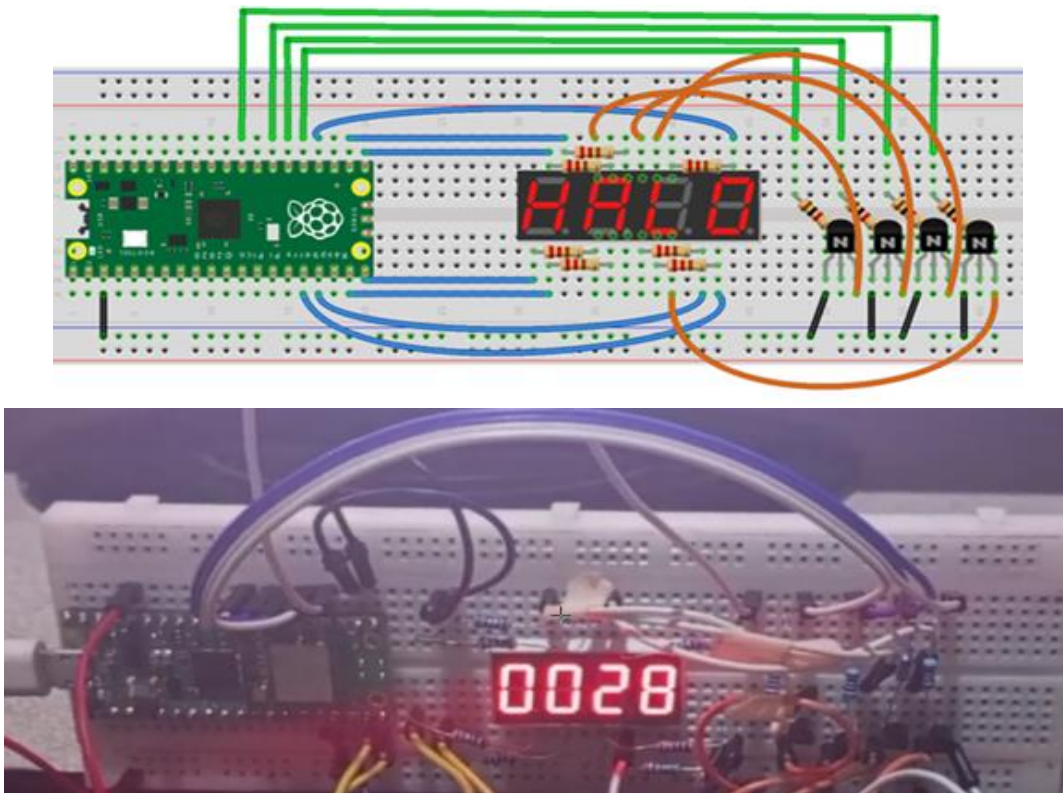
	Codigo 7 Segmento Anodo Común							
Digit	g	f	e	d	c	b	a	Hex
A	0	0	0	1	0	0	0	08
b	0	0	0	0	0	1	1	03
C	1	0	0	0	1	1	0	46
c	0	1	0	0	1	1	1	27
d	0	1	0	0	0	0	1	21
E	0	0	0	0	1	1	0	06
F	0	0	0	1	1	1	0	0E
G	1	0	0	0	0	1	0	42
H	0	0	0	1	0	0	1	09
h	0	0	0	1	0	1	1	0B
i	1	0	0	1	1	1	1	4F
J	1	1	0	0	0	0	1	61
L	1	0	0	0	1	1	1	47
P	0	0	0	1	1	0	0	0C
q	0	0	1	1	0	0	0	18
r	1	0	0	1	1	1	0	4E
S	0	0	1	0	0	1	0	12
U	1	0	0	0	0	0	1	41
u	1	1	0	0	0	1	1	63
Y	0	0	1	0	0	0	1	11

	Codigo 7 Segmento Catodo Común							
Digit	g	f	e	d	c	b	1	Hex
A	1	1	1	0	1	1	1	77
b	1	1	1	1	1	0	0	7C
C	0	1	1	1	0	0	1	39
c	1	0	1	1	0	0	0	58
d	1	0	1	1	1	1	0	5E
E	1	1	1	1	0	0	1	79
F	1	1	1	0	0	0	1	71
G	0	1	1	1	1	0	1	3D
H	1	1	1	0	1	1	0	76
h	1	1	1	0	1	0	0	74
i	0	1	1	0	0	0	0	30
J	0	0	1	1	1	1	0	1E
L	0	1	1	1	0	0	0	38
P	1	1	1	0	0	1	1	73
q	1	1	0	0	1	1	1	67
r	0	1	1	0	0	0	1	31
S	1	1	0	1	1	0	1	6D
U	0	1	1	1	1	1	0	3E
u	0	0	1	1	1	0	0	1C
Y	1	1	0	1	1	1	0	6E

### Diagrama de flujo General:



### Circuito Implementado:



### Video del circuito en Funcionamiento

<https://youtu.be/6wZ65Oaz1jM>

### Información Referencial

<https://controlautomaticoeducacion.com/micropython/display-7-segmentos-pico/>