



Laboratorio 5

Cronómetro simple con D7S

Josué Meneses Díaz

31 de julio de 2024

Objetivos

- Controlar dos Displays de 7 segmentos (D7S) utilizando los pines digitales de un Arduino.
- Utilizar transistores NPN como conmutadores.
- Multiplexar el uso de los displays para optimizar las conexiones de un circuito.

1 Materiales

- 1 x Arduino UNO.
- 2 x Display de 7 segmentos de cátodo común.
- 2x 2N2222 [1]
- 7 x resistencia 330 Ω (referencial).
- 2 x resistencias de 4k7 Ω

2 Procedimiento

En este laboratorio utilizaremos la placa de desarrollo Arduino UNO [2] para controlar el estado de dos D7S. Construiremos un “cronómetro simple” que realice la cuenta desde 0s hasta 59s, mostrando en los D7S el tiempo transcurrido cada segundo. Al llegar al segundo 59, el cronómetro vuelve a 0 reiniciando la cuenta.

El programa de control debe ser desarrollado utilizando el IDE de Arduino [3] junto con los Pines digitales disponibles en la placa de desarrollo. Utilice las siguientes conexiones:

- Pin D0: segmento a
- Pin D1: segmento b
- Pin D2: segmento c
- Pin D3: segmento d
- Pin D4: segmento e
- Pin D5: segmento f
- Pin D6: segmento g

Para el control de cada D7S, utilizaremos dos transistores NPN 2N2222 [1], los cuales permitirán la activación de cada uno de los displays de manera independiente. Utilice las siguientes conexiones para los transistores:

- Pin D8: base transistor 1, Q1
- Pin D9: base transistor 2, Q2

Como se observa en la Figura 1 los pines D0 hasta D6 controlan tanto los segmentos del display 01 como del 02. El control del número mostrado por cada display es realizado mediante la técnica llamada **multiplexación** [4]. Esta técnica consiste en la desactivación de uno de los display para mostrar un valor deseado en el display que



queda activo. La activación/desactivación de cada display es controlada mediante el uso de dos transistores NPN conectados al cátodo común de los display.

2.1 Código de control

Desarrolle un programa que muestre el tiempo transcurrido cada segundo utilizando ambos D7S, donde Q1 muestre las unidades y Q2 las decenas. Simplifique el código desarrollado utilizando funciones dentro de su programa. Por ejemplo, puede definir una función de conmutación como

```
5
6 int PIN_Q1 = 8; // Transistor Q1 (Unidad)
7 int PIN_Q2 = 9; // Transistor Q2 (Decena)
8
9 bool conmutacion(int dt, bool state){
10     /* Función de conmutación de Q1 y Q2
11        - Si state = true  => Q1 = 1 y Q2 = 0
12        - Si state = false => Q1 = 0 y Q2 = 1
13        */
14
15     digitalWrite(PIN_Q1, LOW);
16     digitalWrite(PIN_Q2, LOW);
17
18     if ( state == true ){
19         digitalWrite(PIN_Q1, HIGH);
20         digitalWrite(PIN_Q2, LOW);
21     } else {
22         digitalWrite(PIN_Q1, LOW);
23         digitalWrite(PIN_Q2, HIGH);
24     }
25     delay( dt );
26
27     return !state;
28 }
```

Note que en la transición (conmutación) entre los displays **ambos** deben ser apagados, de no ser así habrá un instante donde ambos D7S se encontrarán encendidos, requiriendo el doble corriente en el circuito pudiendo dañar el MCU.

Utilice una variable de tiempo Δt que defina el tiempo de activación/desactivación entre cada D7S (multiplexación). Pruebe los siguientes tiempos de conmutación:

- 10ms
- 30ms
- 100ms
- 500ms

Utilizando los tiempos anteriores, determine un dt óptimo de funcionamiento del circuito.

Una vez terminadas las actividades de esta guía **no desarme su circuito**, este será utilizando en los próximos laboratorios.

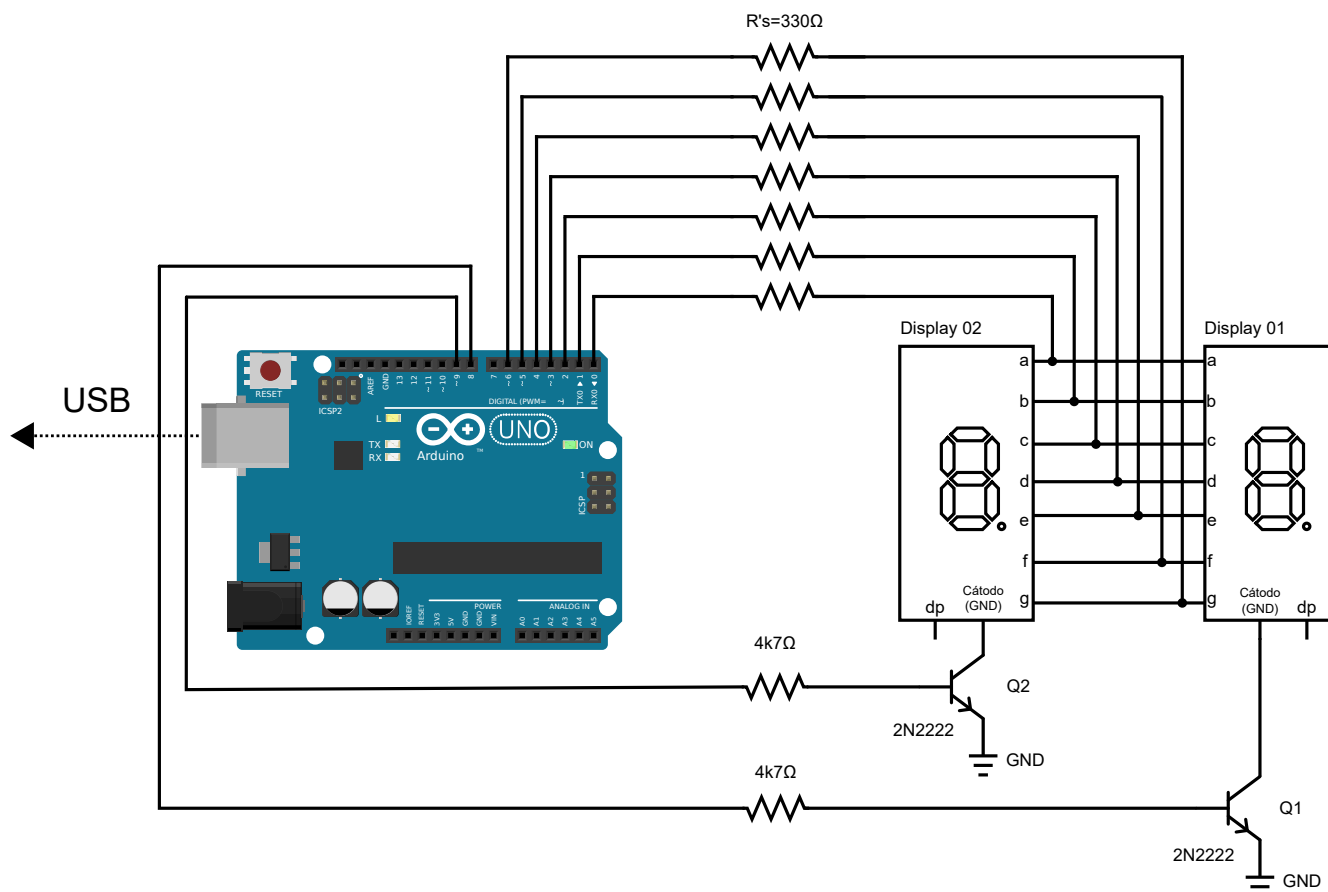


Figura 1: Circuito control de D7S utilizando Arduino UNO.

3 Preguntas reporte

1. Agregue un **diagrama de flujo** del programa diseñado junto con el código final implementado para el control del circuito. Explique la filosofía de su programa. Adjunte el programa *.ino* (Arduino [2]) y la simulación *.sim1* (Simulde [5]) desarrollada. (2 puntos)
2. ¿Qué ventajas encuentra en la multiplexación de los D7S respecto a su control directo mediante los pines digitales del MCU? ¿Qué desventajas encuentra?. (1 punto)
3. Diseñe una modificación (diagrama esquemático) del circuito de la Figura 1 utilizando ahora el CI7448 [6]. ¿Qué ventajas y desventajas encuentra a esta implementación frente al circuito estudiado (Figura 1)?. (2 puntos)
4. Modifique el circuito del punto 3 utilizando las entradas/salidas, \overline{BI} , $\overline{RB0}$ y \overline{RBI} , para suprimir el cero a la izquierda para los valores comprendidos entre el 0 y el 9. Incluya esta modificación como otra simulación en Simulde. (1 punto)



Referencias

- [1] O. Semiconductor, «P2N2222A - Amplifier Transistors NPN Silicon». [En línea]. Disponible en: <https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf>
- [2] Arduino, «Arduino IDE». Accedido: 21 de junio de 2022. [En línea]. Disponible en: <https://www.arduino.cc/en/software>
- [3] Arduino, «Arduino Reference - Arduino Reference». Accedido: 17 de junio de 2022. [En línea]. Disponible en: <https://www.arduino.cc/reference/en/>
- [4] P. Horowitz y W. Hill, *The Art of Electronics*. Cambridge Univ. Press, 2015.
- [5] SimulIDE, «Home Page SimulIDE». Accedido: 21 de junio de 2022. [En línea]. Disponible en: <https://www.simulide.com/p/home.html>
- [6] T. Instrument, «SN7447A SN7448A Data Sheet, Product Information and Support | TI.Com». Accedido: 8 de abril de 2022. [En línea]. Disponible en: <https://www.ti.com/product/SN7447A?keyMatch=7447>

Fecha de Entrega

El reporte de este laboratorio puede ser entregado hasta el miércoles 07-08-2024, mediante la plataforma [Moodle del curso](#).

Para subir el reporte, comprimir en un solo archivo los documentos y subirlos a la plataforma con el nombre de los integrantes del grupo:

lab5_SUSNOMBRES.rar/zip/7zip