UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE LAS CIENCIAS EXACTAS E INGENIERÍAS

INGENIERIA ROBOTICA

SEMINAR ON EMBEDDED SYSTEMS PROGRAMMING PROBLEMS

"ACTIVITY 4"

Materia: Seminar on Embedded Systems Programming Problems
Teacher: Daniel Martinez

Castillo Hernández Luis Fernando
Almanza Castañeda Favio Enrique
Lira Alamo Damian

FEBRUARY 25, 2025

INTRODUCTION:

The objective of this practice is to implement a four-digit counter using an ESP32/ESP8266 and common-anode 7-segment displays. To achieve this, multiplexing is employed, allowing multiple digits to be controlled using a limited number of microcontroller pins.

MATERIALS USED:

- ESP32
- Common-anode 7-segment displays (2 dual displays or 4 individual units)
- Botón pulsador
- **NPN transistors (2N2222)** (used to switch each digit individually)
- **330Ω resistors** (to limit current through the segments)
- Jumper wires and a breadboard

DEVOLPMEN:

Pin Configuration

- In the setup() function, all required pins are configured as outputs. These are categorized into:
- Segment control pins: Connected to the ESP32/ESP8266 to define digit segments (2, 4, 5, 18, 19, 21, 20).
- Digit selection pins: (42, 41, 13, 12) used to activate each of the four digits individually via transistors.

Numeric Representation

- A predefined array dp[10] stores the binary values corresponding to numbers 0–9, matching the segment mapping for a 7-segment display.

Counter logic:

Within the loop() function, a nested loop structure implements a counter from 0000 to 5999, suitable for time-based applications (e.g., a clock displaying minutes and seconds).

The loops iterate as follows:

- Thousands and tens place: Ranges from 0 to 5.
- Hundreds and units place: Ranges from 0 to 9

Digit Multiplexing

Since all digits share the same segment control pins, a multiplexing approach is required:

- Activate one digit at a time using digitalWrite() on the digit control pins.
- Display the corresponding value using mostrarNumero(byte number), which updates the segment configuration based on the desired number.
- Rapid switching between digits (delay(5)) ensures smooth visualization without noticeable flicker.

```
35  for (int mil = 0; mil <6; mil++){
36    for (int centenas = 0; centenas < 10; centenas++){
37      for (int decenas = 0; decenas < 6; decenas++) {
38        for (int unidades = 0; unidades < 10; unidades++) {
39          for (int i = 0; i < 8; i++) {
40
41
42            digitalWrite(42, LOW); // Activar unidades
43            digitalWrite(41, LOW);
44            digitalWrite(13, LOW);
45            digitalWrite(12, HIGH);
46            mostrarNumero(dp[unidades]);
47            delay(5);
48
49
50            digitalWrite(42, LOW);
51            digitalWrite(41, LOW); // Activar decenas
52            digitalWrite(13, HIGH);
53            digitalWrite(12, LOW);
54            mostrarNumero(dp[decenas]);
55            delay(5);
56
```

RESULTS AND OBSERVATIONS:

- Expected functionality: The display correctly cycles through numbers from 0000 to 5999.
- Potential improvement: Optimizing the multiplexing timing for more stable visual output.
- Detected issue: Possible misconfiguration in digit activation logic (i.e., incorrect HIGH/LOW states). A thorough verification of digit selection pins is recommended.

CONCLUSIONS:

"This practice effectively demonstrates the application of multiplexing for controlling multi-digit displays using a microcontroller with limited I/O resources. The ESP32/ESP8266 provides an efficient platform for implementing numerical displays, such as digital clocks or counters, while maintaining low power consumption."