Lab practical 4: Clock-format counter 59-59

Lab Report

Universidad de Guadalajara

Seminario de programación de sistemas embebidos D01-I9893

Team
Ángel de Jesús Vázquez Zarate
Naresh Satoshi López Ojeda
Daniel Esau Gonzalez Molina
Fernanda Hermosillo Gonzáles

Summary-In this practice, a clock-format counter circuit (59-59) was built using an ESP32 microcontroller**.**

Materaials

- ESP32 microcontroller
- 4 transistors 2n2222
- Resistors
- GM4-5641CURG 4-digit 7-segment common cathode display
- Connecting wires
- Breadboard

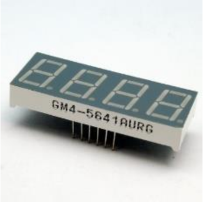
Theoretical Framework

ESP32 Microcontroller

The ESP32 is a low-cost, high-performance microcontroller developed by Espressif Systems. It is widely used in electronics and IoT (Internet of Things) projects due to its advanced features and versatility. It can be programmed using Arduino IDE**,** MicroPython, or Espressif IDF and is applied in various projects, such as remote controls, home automation, monitoring systems, cloud-connected sensors, displays (like in your clock-format counter project), robots, drones, and other smart devices**.**

Display GM4-5641CURG

The GM4-5641CURG is a 4-digit, 7-segment common cathode display with red light emission. This type of device is used to display numbers from 0 to 9, as well as some additional symbols in electronic devices. Each of the seven segments that make up a digit is an LED that can be turned on or off independently to form the desired characters. Since it is a common cathode display, all the LED cathodes are connected together and share a common voltage reference.



transistors 2n2222

The 2N2222 is an NPN bipolar junction transistor (BJT) that allows for the proper transfer of electrical power between different parts of an electrical circuit.

electrical resistance

electronic component that limits the flow of electric current in a circuit.

Multiplexed

refers to a technique where multiple signals share the same physical connection or resource but are displayed or processed sequentially at high speed, giving the illusion of simultaneous operation.

Circuit

ESP32 connection diagram to the display

A7 → 26

B7 → 27

C7 → 32

D7 → 33

E7 → 25

F7 → 14

G7 → 13

DP7 → 35

Digit selection pins (with resistors)

uniSeg → 16

decSeg → 17

uniMin → 18

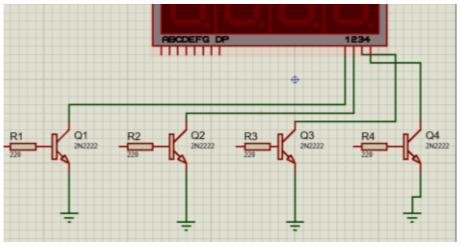
decMin → 19

Transistors and multiplexing

In this project, the 4 transistors are used to control the digits of the 7-segment display. This is necessary because the ESP32 cannot supply enough current to directly power all the segments of a digit at the same time.

How Does Multiplexing Work in a 7-Segment Display?

- One digit is turned on at a time: Although the display has 4 digits, only one is activated at a time.
- Numbers are displayed quickly: The ESP32 first shows the number on the first digit, then moves to the second digit, and so on until the fourth digit.
- Very high speed: This process happens so fast (several times per second) that the human eye perceives all digits as being lit simultaneously.



Code

The code implements a digital clock-style counter using a 4-digit 7-segment display controlled by an ESP32. It counts seconds and minutes, updating every 10 milliseconds. The display is multiplexed using transistors, and each digit is shown sequentially for stability. The program ensures that the display updates smoothly while correctly handling the time logic (resetting seconds at 60 and minutes at 60).

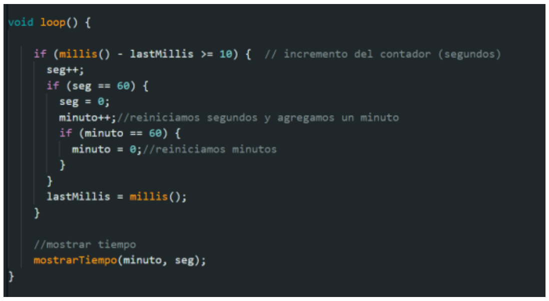
Important functions

Loop()

Increments the seconds counter every 10 milliseconds, When seconds reach 60, they reset to 0, and minutes increase by 1, hen minutes reach 60**,** they reset to 0.

millis()

returns an unsigned long value representing the number of milliseconds that have elapsed since the board started, This value increases continuously as time passes, allowing you to track how much time has passed or to create timed delays without stopping your program.



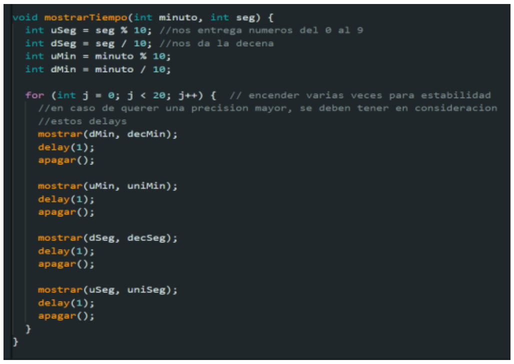
```
void loop() {

  if (millis() - lastMillis >= 10) {  // incremento del contador (segundos)
    seg++;
    if (seg == 60) {
      seg = 0;
      minuto++;//reiniciamos segundos y agregamos un minuto
      if (minuto == 60) {
        minuto = 0;//reiniciamos minutos
      }
    }
    lastMillis = millis();
  }

  //mostrar tiempo
  mostrarTiempo(minuto, seg);
}
```

Function loop

mostrarTiempo()

The function mostrarTiempo() breaks down the minutes and seconds into individual digits (units and tens), Each digit is displayed sequentially using multiplexing, mostrar() lights up the appropriate segments to show numbers 0-9, apagar() turns off all segments before switching to the next digit.

- **uSeg**: Units digit of the seconds (last digit of seg).
- **dSeg**: Tens digit of the seconds (first digit of seg).
- **uMin**: Units digit of the minutes (last digit of minuto).
- **dMin**: Tens digit of the minutes (first digit of minuto

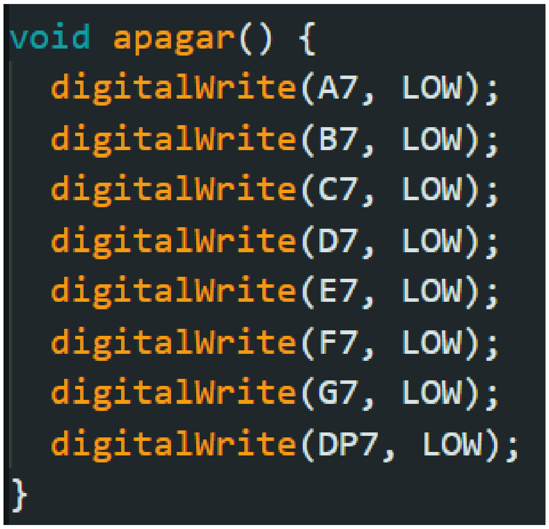


```
void mostrarTiempo(int minuto, int seg) {
  int uSeg = seg % 10; //nos entrega numeros del 0 al 9
  int dSeg = seg / 10; //nos da la decena
  int uMin = minuto % 10;
  int dMin = minuto / 10;

  for (int j = 0; j < 20; j++) {  // encender varias veces para estabilidad
    //en caso de querer una precision mayor, se deben tener en consideracion
    //estos delays
    mostrar(dMin, decMin);
    delay(1);
    apagar();

    mostrar(uMin, uniMin);
    delay(1);
    apagar();

    mostrar(dSeg, decSeg);
    delay(1);
    apagar();

    mostrar(uSeg, uniSeg);
    delay(1);
    apagar();
  }
}
```

Function mostrarTiempo

Apagar()

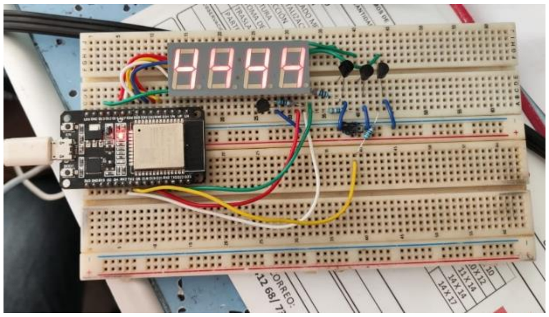
turns off all segments before switching to the next digit.



```
void apagar() {
  digitalWrite(A7, LOW);
  digitalWrite(B7, LOW);
  digitalWrite(C7, LOW);
  digitalWrite(D7, LOW);
  digitalWrite(E7, LOW);
  digitalWrite(F7, LOW);
  digitalWrite(G7, LOW);
  digitalWrite(DP7, LOW);
}
```

Function apagar

Results



conclusion

This code and circuit create a simple digital clock counter using an ESP32 microcontroller, seven-segment displays, and transistors for multiplexing. The primary goal is to display minutes and seconds in a 59:59 format and increment the time in real-time.

ESP32 controls the segments of the 7-segment displays through transistors, which handle the switching of the common cathodes.

Transistors are essential to prevent the ESP32 from being overloaded with current, ensuring the correct operation of the display and the multiplexed display ensures that each digit is displayed in sequence very quickly, making it appear as though all digits are illuminated simultaneously.

This circuit and code was an ideal project for learning the basics of time management and display handling using an ESP32 and how to use multiplexing techniques for better efficiency in the circuit

References

- Arduino. (n.d.). *ESP32 Documentation.* Retrieved from https://www.arduino.cc
- IEEE Xplore. (2020). *Multiplexing Techniques in Digital Displays.* Retrieved from https://ieeexplore.ieee.org