University of Guadalajara

University Center of Exact Sciences and Engineering

Practice report

Counter from 00:00 to 99:99 using ESP32

Presented by:
Zahara Nathalia Ibarra Casillas
Kevin Eduardo López Martínez
Brian Michel Rubio Martínez

I9398 - D01
Embedded Systems Programming Troubleshooting Seminar
Made in LaTeX

Índice

1. Introduction

This activity was made inside of the class schedule, which is why we are going to explain only topics that we studied at the time.

1.1. Objective

(I) Use the ESP32 development board to create a stopwatch that counts to 9999.

(II) Use the ESP32 development board to create a clock that, once the seconds reach 60, starts counting the minutes.

2. Development

First, we will see some fundamental concepts to understand why to use the ESP32 microcontroller and the development of our code.

2.1. Theoretical framework

(I) What is the keyword const used for in Arduino?
The const keyword stands for constant. It is a variable qualifier that modifies the behavior of the variable, making a variable read-only". This means that the variable can be used just as any other variable of its type, but its value cannot be changed. You will get a compiler error if you try to assign a value to a const variable.
Constants defined with the const keyword obey the rules of variable scoping that govern other variables. This, and the pitfalls of using define, makes the const keyword a superior method for defining constants and is preferred over using define.

(II) What is the unsignedlong data type in Arduino used for?
Unsignedlong (32 bit)- unsigned number between 0 and 4294967295. This type is commonly used to store the result of the millis() function, which returns the time the current code has been running, in milliseconds.

(III) What is the purpose of the digitalWrite function in Arduino?
The digitalWrite() function, normally used in the loop() function, is used to write a value (or set a state) to a digital pin. The possible values or states are HIGH or LOW.

(IV) What is the purpose of the digitalRead function in Arduino?
The digitalRead() function, normally used in the loop() function, is used to read a value (or set a state) of a digital pin. The possible values or states are HIGH or LOW. The value read can be stored in a variable or dynamically checked in a condition.

2.2. Methodology/Procedure

"segmentPins" stores the pins to which the segments of the 7-segment display (A-G) are connected, using pins 6, 7, 12, 11, 10, 8 and 15 of the ESP32, allowing it to be activated or deactivated independently. transistor1 to transistor4 (that are connected to the pins 13, 5, 4 and

3 of the ESP32) control which digit is active at a given time, using multiplexing. It should be noted that we use to define these constants with “const int ” because we will use them only to read the components. We also define a “numbers” array using the constant “const byte” to define which segments should be turned on (1) or off (0) to represent the numbers from 0 to 9.

We also defined auxiliary variables such as “seconds” and “minutes”, which stores the actual time counted. Also “lastUpdateTime” is used to record the time of the last count (we use the “unsigned long” data type to return the time the code has traveled in milliseconds), and “updateInterval” is used to define the update interval of 250 ms, simulating a “fast second”.

In the setup() function, the segment and transistor pins are configured as outputs, and all segments and transistors are turned off to start in a known state using the “digitalWrite()” function.

In the loop() function we use the millis() function to get the time in milliseconds since startup and then compare “currentTime” with “lastUpdateTime” to increment seconds every 250ms.
If the seconds reach 60, they are reset to 0 and the minutes are incremented and if the minutes reach 99, they are also reset to 0.
After that, the minutes and seconds digits are separated into tens and ones and the 4 digits on the screen are updated one at a time with the function “mostrarDigito()” .

The function “mostrarDigito()” turns off all transistors to avoid overlapping numbers, then turns on the transistor corresponding to the digit we want to display and activates the display segments according to the numbers matrix and introduces a short delay(2) to give time to display the number before updating another digit.

3. Results

Below, we add some pictures of the microcontroller with the code already loaded and the circuit assembled.

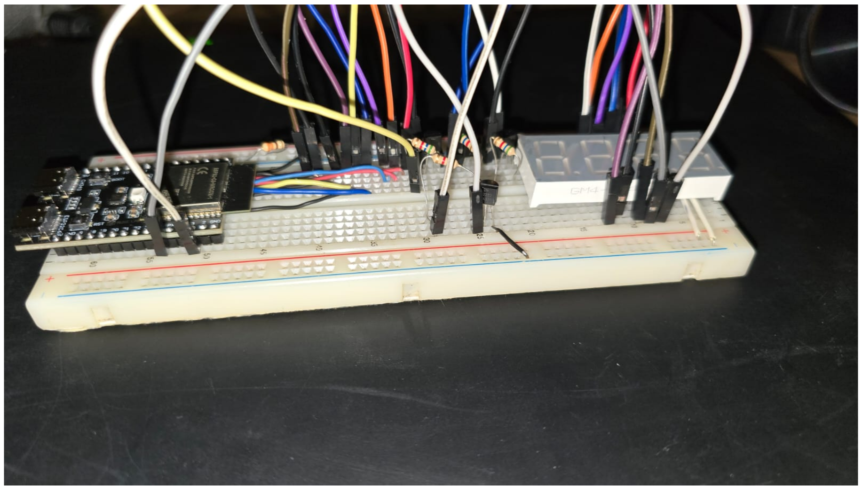
Figura 1: Photo taken of the mounted circuit.

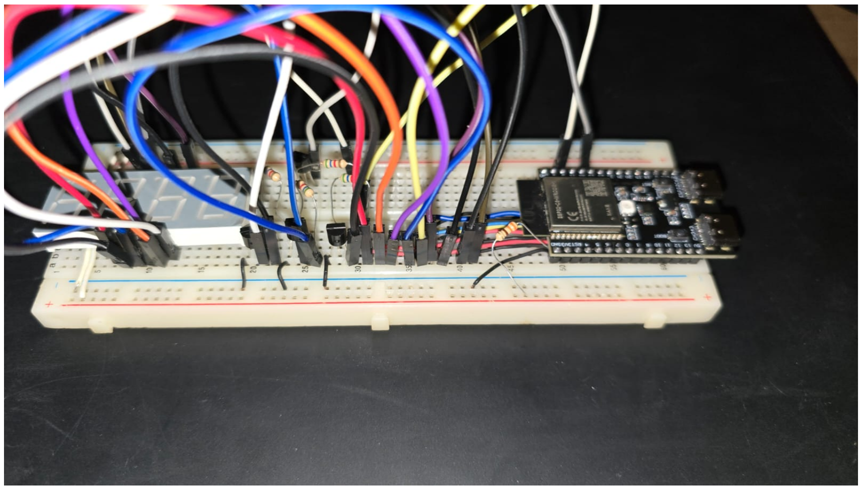
Figura 2: Another photo taken of the mounted circuit.

4. Conclusion

The most remarkable function is “mostrarDigito()” since it is in charge of multiplexing, activating a single digit at a time and configuring the display segments according to the corresponding number. It took a bit of work to define the constants and the matrices to read the display and the transistors and set the state they were in, but we just had to set them, so, we reaffirm, the most remarkable function is mostrarDigito().

Referencias

[1] Arduino. *const - Guía de Referencia de Arduino*[Online]. Available: https://www.arduino.cc/reference/es/language/variables/variable-scope-qualifiers/const/

[2] Arduino. *Introducción a los Tipos de Dato con Arduino*[Online]. Available: https://arduino.cl/introduccion-a-los-tipos-de-dato-con-arduino/: :text=unsignedlong

[3] M. Bellan, (2019, July 04). *Salidas digitales con Arduino*[Online]. Available: https://www.programoergosum.es/tutoriales/salidas-digitales-con-arduino/

[4] M. Bellan, (2019, July 04). *Entradas digitales con Arduino*[Online]. Available: https://www.programoergosum.es/tutoriales/entradas-digitales-con-arduino/