

Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería



SISTEMAS EMBEBIDOS

Práctica 3: UART

Docente: Evangelina Lara Camacho

Alumno: Gómez Cárdenas Emmanuel Alberto

Matricula: 01261509

Objetivo

El alumno se familiarizará con el uso del periférico UART usando el sistema embebido ESP32 DevKit v1 para desarrollar aplicaciones para sistemas basados en microcontrolador para aplicarlos en la resolución de problemas de cómputo, de una manera eficaz y responsable.

Equipo

Computadora personal con conexión a internet.

Teoría

- **Describe el modo UART half-duplex del ESP32**

El término dúplex es utilizado para definir un dispositivo capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

- **Dúplex (dúplex completo o full-duplex)**

- Permiten canales de envío y recepción simultáneos.

- **Semidúplex (half-duplex)**

- Es una conexión en la que los datos fluyen en una u otra dirección, pero no los dos al mismo tiempo

- **Simplex**

- Únicamente permiten la transmisión en un sentido.

Half-duplex en el ESP32

El controlador UART del ESP32 no cuenta con un soporte integrado para la comunicación automática en half-duplex, sin embargo, este tipo de UART puede ser logrado mediante software (controlando el pin RTS manualmente).

Para poder utilizar UART en modo half-duplex se hace lo siguiente:

1. **Configurar UART:** No hay pasos específicos en comparación con full-duplex.
2. **Establecer UART al modo RS485:** Utilizando la función `uart_set_mode()` mandando `UART_MODE_RS485_HALF_DUPLEX` como argumento.
3. **Configurar PIN:** Conectar el pin RTS al pin ~RE/DE del controlador RS485 para habilitar la transmisión o la recepción.

Desarrollo

Implemente en dos ESP32 ESP-IDF una pancarta ASCII haciendo uso de **UART y tareas**. La implementación debe ser eficiente en el uso de recursos de cómputo (procesador, memoria y periféricos).

Un ESP32 recibe del usuario una cadena de texto máximo 25 caracteres usando UART. El ESP32 envía por UART la cadena al otro ESP32. Este ESP32 convierte los caracteres a un formato de pancarta ASCII y los despliega en la terminal por medio de UART. Limite los caracteres válidos a las letras del alfabeto en mayúscula y minúscula, números, espacio y los caracteres especiales '!', ',', '+' y '-'.

Fig. 1. Diagrama a bloques.



La pancarta ASCII es de su elección, pero debe ser similar a las dos de los siguientes ejemplos.

Fig. 2. Ejemplo 1 de pancarta.



Fig. 3. Ejemplo 2 de pancarta.



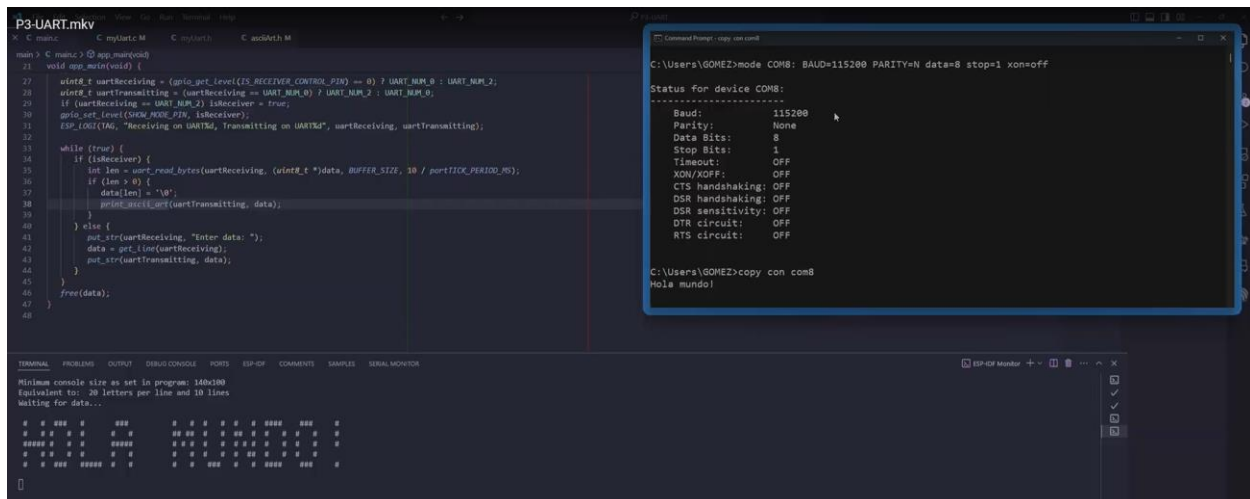
UARTs en uso:

- **UART0:** Utilizado para la interfaz con la terminal.
 - o **ESP32-1:** Usa la terminal para recibir datos.
 - o **ESP32-2:** Usa la terminal para mostrar los datos recibidos mediante UART2.
- **UART2:** Utilizado para la interfaz con el otro ESP32.
 - o **ESP32-1:** Los datos obtenidos en el UART0 son enviados mediante el UART2 hacia el otro ESP32.
 - o **ESP32-2:** Recibe los datos del otro ESP32 y los manda al UART0.

Diferenciación entre ESP32

Debido a que el mismo programa es usado en ambos ESP32, he decidido utilizar el pin 23 para indicar si el ESP32 es un receiver o un transmitter.

- **Transmitter:** Configurado al conectar el pin 23 a tierra, este recibe los datos mediante la consola (UART0) y los transmite al otro ESP32 (UART2).
- **Receiver:** Configurado al conectar el pin 23 a 3V3, el LED interno (GPIO2) del ESP32 será encendido para indicar que es el ESP32 receptor. Este recibe los datos del otro ESP32 (UART2) y los enviará a la consola (UART0) en formato pancarta.



The screenshot shows an IDE with a file named 'P3-UART.mkv'. The code is written in C and includes comments in Spanish. It defines two UARTs (UART0 and UART2) and implements a loop that checks if the device is a receiver or a transmitter based on the state of pin 23. If it's a receiver, it reads data from UART2 and sends it to UART0. If it's a transmitter, it reads data from UART0 and sends it to UART2. The code also includes a delay function and a main loop that keeps the program running.

```
1 // P3-UART.mkv
2 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
3 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
4 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
5 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
6 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
7 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
8 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
9 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
10 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
11 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
12 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
13 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
14 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
15 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
16 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
17 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
18 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
19 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
20 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
21 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
22 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
23 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
24 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
25 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
26 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
27 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
28 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
29 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
30 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
31 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
32 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
33 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
34 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
35 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
36 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
37 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
38 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
39 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
40 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
41 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
42 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
43 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
44 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
45 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
46 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
47 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
48 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
49 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
50 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
51 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
52 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
53 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
54 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
55 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
56 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
57 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
58 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
59 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
60 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
61 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
62 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
63 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
64 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
65 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
66 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
67 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
68 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
69 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
70 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
71 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
72 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
73 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
74 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
75 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
76 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
77 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
78 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
79 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
80 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
81 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
82 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
83 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
84 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
85 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
86 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
87 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
88 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
89 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
90 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
91 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
92 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
93 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
94 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
95 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
96 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
97 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
98 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
99 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
100 // C:\Users\GOMEZ\Documents\Arduino\Sketches\ESP32-UART\ESP32-UART.ino
```

The terminal window shows the status of the COM8 port. The status is 'Status for device COM8:'. The baud rate is 115200, parity is None, data bits are 8, stop bits are 1, timeout is OFF, XON/XOFF is OFF, CTS handshaking is OFF, DSR handshaking is OFF, DSR sensitivity is OFF, DTR circuit is OFF, and RTS circuit is OFF. The terminal also shows the text 'C:\Users\GOMEZ>copy con com8' and 'Hola mundo!'.

La evidencia del funcionamiento se encuentra en [Drive](#)

Conclusiones y Comentarios

El uso del UART es muy importante ya que es muy fiable, fácil de usar además de ser versátil, gracias a esto es un protocolo fundamental en una amplia gama de aplicaciones, especialmente en sistemas embebidos.

Dificultades en el Desarrollo

Al estar usando el mismo UART para la comunicación a consola, esto hace que, si está mal configurado, es imposible depurar mediante texto, ya que puede provocar conflictos.

Igualmente, al configurar ambos ESP32 para enviar/recibir con el UART2 y no recibir datos, se me complicó el descubrir si el problema era en el ESP32 que enviaba datos o el que recibía.

Referencias

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/index.html>

Código

El código fuente puede ser encontrado en el [Repositorio de GitHub "Sistemas Embebidos"](#)

Al ser múltiples archivos, solo presentaré archivo principal "main.c"

```
#include "myUart.h"

#define IS_RECEIVER_CONTROL_PIN GPIO_NUM_23
#define SHOW_MODE_PIN GPIO_NUM_2

static const char *TAG = "P3 - UART";
bool isReceiver = false;

void gpio_setup() {
    gpio_config_t io_conf;
    io_conf.intr_type = GPIO_INTR_DISABLE;
    io_conf.mode = GPIO_MODE_INPUT_OUTPUT;
    io_conf.pin_bit_mask = 1ULL << IS_RECEIVER_CONTROL_PIN
                        | 1ULL << SHOW_MODE_PIN;
    io_conf.pull_down_en = GPIO_PULLDOWN_ENABLE;
    io_conf.pull_up_en = GPIO_PULLUP_DISABLE;
    gpio_config(&io_conf);
    ESP_LOGI(TAG, "GPIO setup complete");
}
```

```
void app_main(void) {
    init_UARTs();
    gpio_setup();
    char *data = (char *)malloc(BUFFER_SIZE);

    // Set mode
    isReceiver = (gpio_get_level(IS_RECEIVER_CONTROL_PIN) == 1);
    uint8_t uartReceiving = (isReceiver) ? UART_ESP : UART_CONSOLE;
    uint8_t uartTransmitting = (!isReceiver) ? UART_ESP : UART_CONSOLE;
    gpio_set_level(SHOW_MODE_PIN, isReceiver);
    ESP_LOGI(TAG, "Receiving on UART%d, Transmitting on UART%d",
              uartReceiving, uartTransmitting);

    while (true) {
        if (isReceiver) {
            int len = uart_read_bytes(uartReceiving, (uint8_t *)data,
                                      BUFFER_SIZE, 10 / portTICK_PERIOD_MS);

            if (len > 0) {
                data[len] = '\0';
                print_ascii_art(uartTransmitting, data);
            }
        } else {
            put_str(uartReceiving, "Enter data: ");
            data = get_line(uartReceiving);
            put_str(uartTransmitting, data);
        }
    }
    free(data);
}
```