

Actividad de laboratorio: Explorando la comunicación RS232

Comunicaciones Digitales UMNG

1. OBJETIVOS :

- Comprobar la relación de tiempo de bit / tasa de transmisión de datos
- Analizar los detalles de la capa física del estándar Rs 232
- Analizar la estructura del protocolo de comunicación serial, identificando bit de inicio, bit de parada, datos y paridades.
- Desarrollar habilidades de programación empleando Micropython

2. COMPETENCIAS A DESARROLLAR:

El estudiante deberá tener dominio de al menos los siguientes conceptos y herramientas así:

- Capacidad en el manejo del osciloscopio digital para la medida de señales transitorias.
- Medición del tiempo de bit en una señal digital
- Análisis de las señales digitales al modificar el parámetro paridad y tasa de baudios
- Establecer la relación tiempo de bit y tasa de baudios.
- Analizar una señal digital respecto al tiempo
- Habilidad para interconectar dispositivos mediante el protocolo de comunicación
- Desarrollo de algoritmos de comunicación empleando Python, MicroPython y Matlab

3. PROCEDIMIENTO:

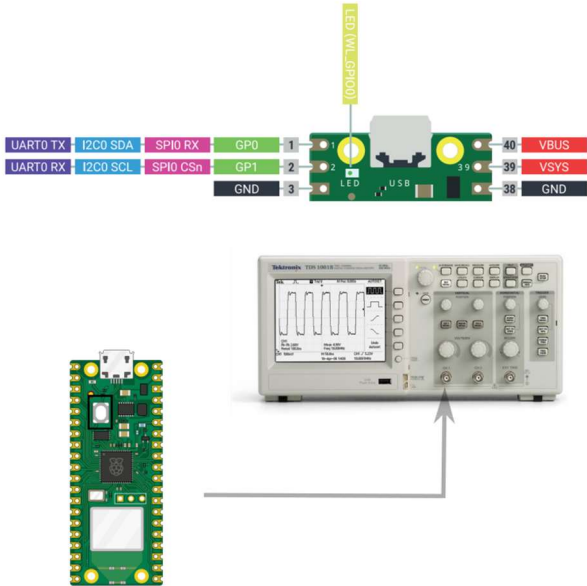
Empleando un dispositivo Raspberry Pi Pico o Pico 2W:

3.1 Realice la configuración inicial del dispositivo

Descargue el Firmware en: <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>

Siga las instrucciones descritas en esta página para descargar el archivo .UF2 correspondiente al dispositivo.

3.2 Identifique los terminales TX y RX en las UART del dispositivo (1 y 2 para el UART0). Conecte un osciloscopio digital entre la terminal TX y tierra del dispositivo (Consulte las especificaciones técnicas del dispositivo). Tenga en cuenta que el Terminal TX es el pin 1 y el terminal GND es el pin 3.



3.3. El siguiente programa le permitirá enviar caracteres ASCII a través del terminal TX.

```
rs232.py x
1 import machine
2 import utime
3 from machine import Pin,UART
4 led = machine.Pin("LED", machine.Pin.OUT)
5 uart = UART(0, baudrate=9600,bits=8,parity=0, tx=Pin(0), rx=Pin(1))
6 while True:
7     led.on()
8     uart.write("U")
9     utime.sleep(1)
10    led.off()
11    utime.sleep(1)
12
```

Ejecute el programa y mida en el osciloscopio la señal recibida. Configure adecuadamente el disparo (trigger) del

osciloscopio para que se capture la señal. Mida el tiempo de bit y verifique si corresponde a la medida teórica esperada.

3.4 Parámetros de la UART en Micropython.

Consulte siempre la documentación oficial de Micropython:
<https://docs.micropython.org/en/latest/>

La configuración de la UART del dispositivo en Micropython:
<https://docs.micropython.org/en/latest/library/machine.UART.html>

Responda:

- ¿Cuál es la clase disponible en Micropython para la comunicación serial RS 232?
- Cree una tabla con los métodos disponibles para la clase UART. Explique cada uno de ellos.
- ¿Como se modifica la tasa de baudios?

3.5 Medida de los tiempos de bit.

Envíe ahora el caracter Ascii "W". Elija al menos seis diferentes tasas de baudios y mida los tiempos de bit. Tabule los datos (ver tabla sugerida) y capture las imágenes para documentar esta prueba. Compruebe que $t_b=1/\text{tasa de baudios}$. Calcule los porcentajes de error entre los datos teóricos y experimentales.

Baudios	Tiempo de bit (Teórico)	Tiempo de bit (experimental)	Δt	% error	Voltaje máximo	Voltaje mínimo
300						
..						
..						
..						
115200						
max						

3.6. Análisis de la estructura del protocolo RS232.

3.6.1 Envíe **diez caracteres ASCII diferentes** al carácter empleado en el numeral anterior. Con ayuda del osciloscopio digital capture cada uno de los datos enviados. **Envíe los datos a _____ baudios, 8 bits de datos y paridad par.**

3.6.2 Identifique en el osciloscopio su correspondiente valor binario. Analice la estructura del protocolo RS232. Dibuje también la señal de **forma manual** identificando: el bit de inicio, los bits de datos, el bit de paridad y el bit de parada. Identifique los bits LSB y el MSB en cada una de las señales dibujadas.

3.6.3 Mida el tiempo total requerido para el envío de cada carácter, contrástelo con lo explicado en la clase.

3.6.4 Modifique la paridad a paridad IMPAR. Capture las mismas señales y compare las tramas. Identifique las diferencias existentes.

3.6.5 Modifique la paridad a SIN PARIDAD. Capture las mismas señales y compare las tramas. Identifique las diferencias existentes.

3.7 Medida del tiempo de la trama

3.7.1 Modifique el programa para que envíe una trama ASCII de 60 caracteres enviados a 600 baudios, paridad par y 8 bis de datos. Capture la señal resultante en el osciloscopio y mida el tiempo total de la trama generada.

3.7.2 Realice la misma prueba anterior, pero **elimine el bit de paridad**. Capture la señal resultante en el osciloscopio y mida el tiempo total de la trama generada. Compare con el resultado anterior.

3.7.3 Capture la trama "UMNG LIDER EN INGENIERIA EN TELECOMUNICACIONES" enviada a 57600 baudios, 7 bits de datos, paridad par y dos bits de parada. Mida el tiempo total del mensaje. Contraste los valores teóricos contra los experimentales.

4. Reto de programación:

4.1. Interconecte dos dispositivos Raspberry Pi_Pico cruzando las líneas TX y Rx entre los dispositivos (*Tenga en cuenta que deben tener la misma señal GND de referencia entre los dos circuitos*).

4.2 Diseñe un programa para transmisión TX.py y un programa para recepción RX.py para ejecutarse en los dispositivos Raspberry pi. El programa de transmisión envía cada dos segundos un carácter ASCII. Al ser recibido este carácter el dispositivo de recepción enciende un led durante cinco segundos y envía como respuesta otro carácter que al recibirse en el dispositivo transmisor, enciende un led que **parpadea** durante tres segundos. El proceso se repite de forma permanente. En las líneas de TX y RX de los dispositivos se conectan los canales del osciloscopio digital para analizar el proceso. El dispositivo receptor debe además crear el archivo **recibidos.txt** donde incluye el contador de numero de recepciones (Utilice un salto de línea para que quede un solo número por línea en el archivo txt).

4.3 Modifique los programas para realizar una comunicación Full-Duplex, utilice el osciloscopio digital para validar el funcionamiento.

Observaciones:

- Para realizar las medidas es necesario que ajuste el disparo del osciloscopio para que pueda capturar eventos transitorios. Revise las opciones del disparo (Trigger).
- En el informe realice una comparación entre los parámetros técnicos de las Raspberries Pi Pico W y Pi pico 2W. Analice las diferencias.