## Lab. Fundamentos de Sistemas Embebidos Práctica 4

### Raspberry Pi: Puerto serie y módulo UART

#### RG Ramírez-Chavarría

## Objetivo

El alumno conocerá la configuración, la programación y el uso del módulo UART (*Universal Asynchronous Receiver Transmitter*) para implementar comunicaciones de tipo serial, particularmente bajo el estándar RS-232.

### 1. Pre-requisitos

- Descargue e instale los paquetes y bibliotecas de su elección (WiringPI, BCM358, pigpio C#, Python RPio, etc...)
- Un convertidor USB a RS-232 (eg. MAX232, USB-TTL, etc...)
- Un programa emulador de terminal serie (eg. Putty, Termite, etc...), en la PC Windows.
- Un programa emulador de terminal serie (eg. minicom), en la tarjeta Raspberry Pi.

## 2. Conexión entre Raspberry Pi y una PC

- Dependiendo del modelo de su tarjeta, ubique en el conector las terminales UART\_TXD y UART\_RXD.
- Las terminales anteriores deberán conectarse a Rx y Tx del adaptador USB a RS-232 que esté utilizando.
- Las tierras (GND) de la tarjeta y del adaptador deben estar unidas.
- Verifique que los voltajes estén a 3.3 volts.
- Conecte el adaptador al USB de la PC, y en el administrador de dispositivos ubique el puerto COMX en donde se instaló el dispositivo.

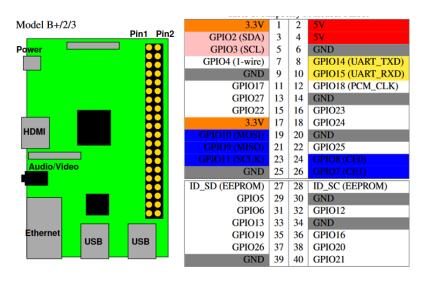


Figura 1: Tarjeta Raspberry Pi y distribución de terminales.

### 3. Configuración UART en Raspberry Pi

- Abra el programa emulador de terminal serie en la PC, establezca el puerto COMX y una velocidad de transmisión de 115200 baudios.
- Abra el archivo /boot/config.txt
- Agregue las lineas siguientes al final del archivo
   # Enable UART
   enable\_uart=1
- Reinicie el sistema con sudo reboot.
- En la terminal de la PC (abierta en el COMX asociado) deberá observar una leyenda indicando el reinicio del sistema y posteriormente la opción para iniciar sesión (Fig. 2).

```
[ 415.243583] reboot: Restarting system
Raspbian GNU/Linux 10 raspberrypi ttyAMA0
raspberrypi login:
```

Figura 2: Emulador de terminal en la PC.

#### • Deshabilitar el servicio de consola

- Ahora, la comunicación UART entre la Raspberry Pi y la PC se ha establecido. Sin embargo, el UART de la RPi es usado por la consola por default.
- Usando la conexión SSH (no UART), el siguiente paso es deshabilitar la consola:

sudo systemctl disable serial-getty@ttyAMAO.service (\*Observe que ttyAMAO debe corresponder con el nombre de la interfaz, al final de la segunda línea de la Fig. 2.

Abra el archivo de la consola sudo nano /boot/cmdline.txt

Elimine el fragmento "console=serial0,115200" de la línea, y guarde el archivo.

Reinicie el sistema sudo reboot

# 4. Emulador de terminal serie en Raspberry Pi

Para enviar y recibir datos en la Raspberry Pi, es necesario usar un programa emulador, en este caso usaremos minicom.

- Para instalar minicom use el comando sudo apt-get install minicom -y.
- Para lanzar el emulador y conectar el puerto UART, use el siguiente comando minicom -b 115200 -o -D /dev/ttyAMAO.
- Verá una pantalla como la de la Fig. 3.

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyAMA0

Press CTRL-A Z for help on special keys
```

Figura 3: Terminal minicom habilitada en el puerto ttyAMAO.

Prueba: Escriba la cadena Hola PC en minicom (RPi), y observe dicha cadena en el emulador serial de la PC. Sí todo está correcto, en el emulador serial de Windows escriba la cadena Hola Raspberry, y observe dicha cadena en el emulador minicom de las Raspberry Pi.

### 5. Manejando el UART con un lenguaje de programación

Una vez que ha verificado que la conexión Raspberry Pi - PC se ha establecido correctamente.

#### Actividad 1

Desarrolle un programa en C o Python que escriba de la Raspberry a la PC, la siguiente cadena,  $i=0,\dots,15$  veces la

```
0: FSE 2020-1 Comunicacion UART RPi - FSE
:
15: FSE 2020-1 Comunicacion UART RPi - FSE
```

\*\*Observe en cada iteración la cadena se escribe debajo de la cadena anterior y no en la misma línea.

#### Actividad 2

Desarrolle un programa en C o Python que sea capaz de recibir datos provenientes de la PC hacia la Raspberry Pi. LA cadena que enviará desde la PC deberá tener la estructura siguiente:

numero entero, iniciales, numero binario, numero decimal

en donde numero entero es número entre 0 y 4095, iniciales es una cadena con las iniciales de los nombres de cada equipo, numero binario es un 0 ó 1, y numero decimal es un número entre 0.0 y 3.3.

Una vez recibida la cadena, el programa deberá **separar** cada campo de la trama, e imprimirlos en la terminal **minicom** con el siguiente formato:

```
Número entero: xxxx
Iniciales: xxxx
Bandera: x
Voltaje: xxx
```

en donde xx... son los datos previamente recibidos por el puerto UART.

\*\*Considere que al final de la cadena numero entero, iniciales, numero binario, numero decimal, enviada desde la PC debe haber un "salto de línea o enter" de forma automática, y que el programa en la RPi debe ser capaz de detectarlo.

# 6. Entregables

Deberá crear un repositorio en github con el nombre FSE 2020-1.

Allí deberá haber una carpeta con el nombre Practicas y dentro de ella habrá subcarpetas con el nombre Practica X con  $X \in [1, N]$ .

Dentro de la carpeta Practica 4 incluya los archivos de esta práctica.

- 1. En un archivo README escriba Objetivos y Descripción de la Práctica
- 2. Nombre de los integrantes
- 3. Link a un video-tutorial en youtube en donde se explique el funcionamiento de la práctica.

Fecha límite: Martes 24/09/2019