Manual de usuario

X-BOARD
Raspberry Pi

XIDE®

XIDE® es un kit de hardware integrado por módulos **X-NODE** y tarjetas de expansión **X-BOARD**, diseñado para incorporar en menos de 24 hrs, prototipos de hardware para proyectos de Internet de las Cosas **IoT.**

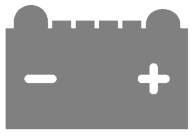
MICROSIDE®

XIDE® es un proyecto realizado por **Microside Technology**, empresa orgullosamente mexicana, especializada en diseño y producción de soluciones tecnológicas para IoT.

X-BOARD
Características

Compatible con estándar de conexión mikroBUS™

Compatible con estándar Qwiic® y STEMMA QT®



Compatible con niveles lógicos de voltaje

3.3V < > 5V

X-BOARD
Raspberry Pi

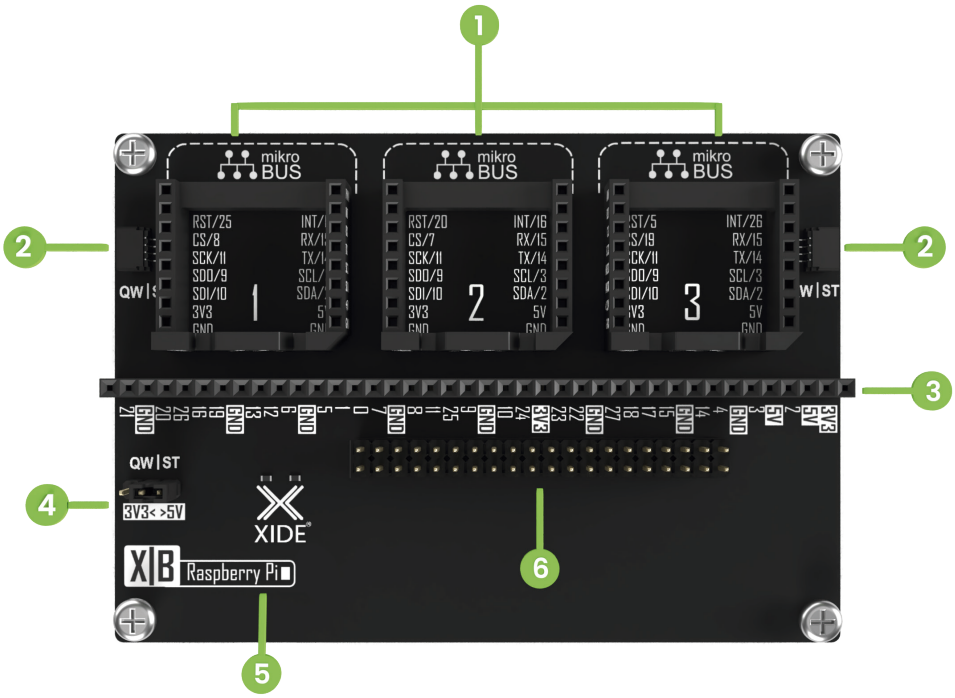
I. Introducción

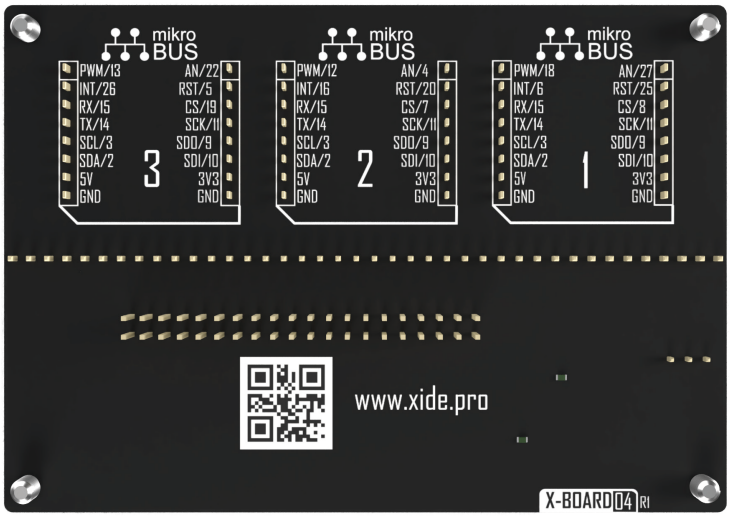
La **X-BOARD Raspberry Pi** es una tarjeta de expansión diseñada principalmente para evaluar de forma rápida sensores, actuadores o módulos de comunicación integrados en los X-NODE en conjunto con las tarjetas Raspberry Pi (modelos 4B, 3B+, Zero y Zero W). Es ideal para el desarrollo de prototipos y proyectos electrónicos donde se requiera una integración sencilla, tamaño compacto y compatibilidad con herramientas de desarrollo con los estándares **mikroBUS™**, **Qwiic®** y **STEMMA QT®**. Cuenta con un selector de voltaje entre 3.3 V y 5 V para los conectores JST, tres zócalos de conexión con estándar **mikroBUS™** y un header hembra con el acceso a todos los pines de la tarjeta Raspberry Pi.

II. ¿Cómo funciona?

Para una comunicación con herramientas de desarrollo que poseen estándares **mikroBUS™**, **Qwiic®** o **STEMMA QT®,** simplemente bastará con insertar la tarjeta Raspberry Pi en el puerto dedicado y los módulos **X-NODE** o tarjetas compatibles en los zócalos o conectores de la **X-BOARD Raspberry Pi**. El usuario podrá seleccionar el voltaje de operación en los conectores JST y usar el header hembra para disponer de los pines de la tarjeta Raspberry Pi colocada, de esta forma puede utilizar un analizador lógico en diversas pruebas de funcionamiento.

La tarjeta **X-BOARD Raspberry Pi** es compatible con el estándar **mikroBUS™** de Mikroe® para una conexión fácil e integra conectores JST que admiten el estándar **Qwiic®** de SparkFun® y el estándar **STEMMA QT®** de Adafruit® para una comunicación entre diversos módulos y tarjetas de desarrollo por medio del protocolo I^2C de manera rápida y sencilla.





III. Descripción del hardware

1. Conectores estándar **mikroBUS™**
2. Conectores JST compatibles con **Qwiic®** y **STEMMA QT®**
3. Header para disposición de los pines de cada conector **mikroBUS™**
4. Selector de voltaje en el conector JST 3.3V <> 5V
5. Modelo de X-BOARD
6. Conector para tarjeta **Raspberry Pi**

IV. Especificaciones técnicas

Modelo	Raspberry Pi
Módulos compatibles	Raspberry Pi Modelo 4B, 3B+, Zero y Zero W.
Estándares compatibles	Estándar mikroBUS™, estándar Qwiic® y estándar STEMMA QT®
Características	3 Zócalos de conexión con estándar mikroBUS™, 2 conectores JST compatibles con el estándar Qwiic® o estándar STEMMA QT® y un header de 40 pines para disposición de los pines de las tarjetas Raspberry Pi.
Tamaño	72 x 102 x 14 mm
Voltaje	3.3V o 5V

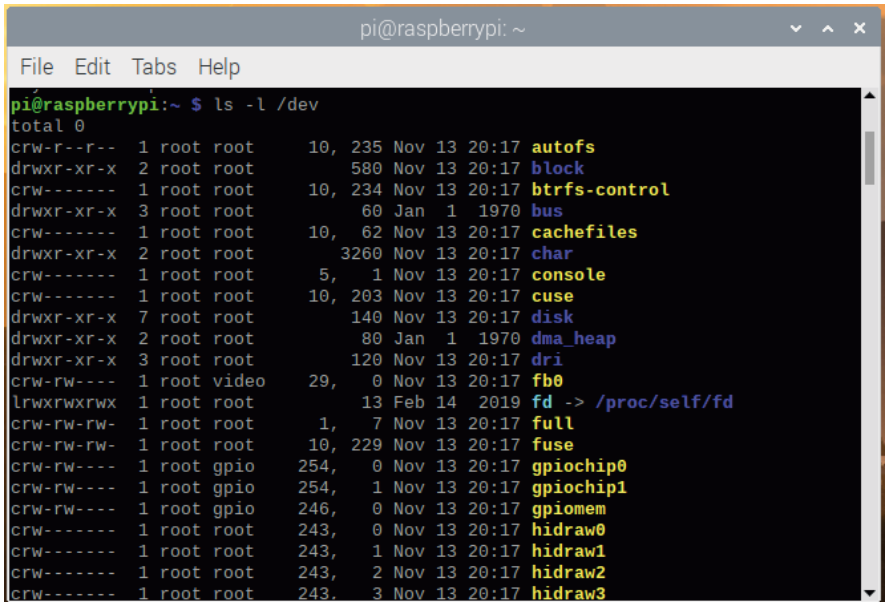
V. Ejemplo de uso

A continuación encontrarás ejemplos prácticos para el uso de la **X-BOARD Raspberry Pi** utilizando Python y la terminal Putty en el sistema operativo Raspbian.

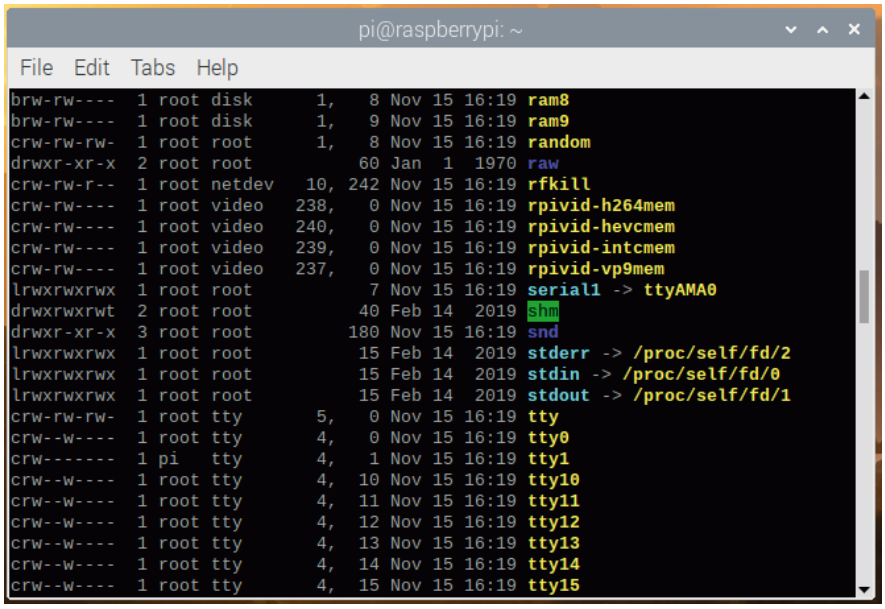
Habilitar puerto serial

Antes de iniciar es necesario habilitar el puerto serial en nuestra **Raspberry Pi,** para ello ejecutamos el siguiente comando:

```
ls -l /dev
```



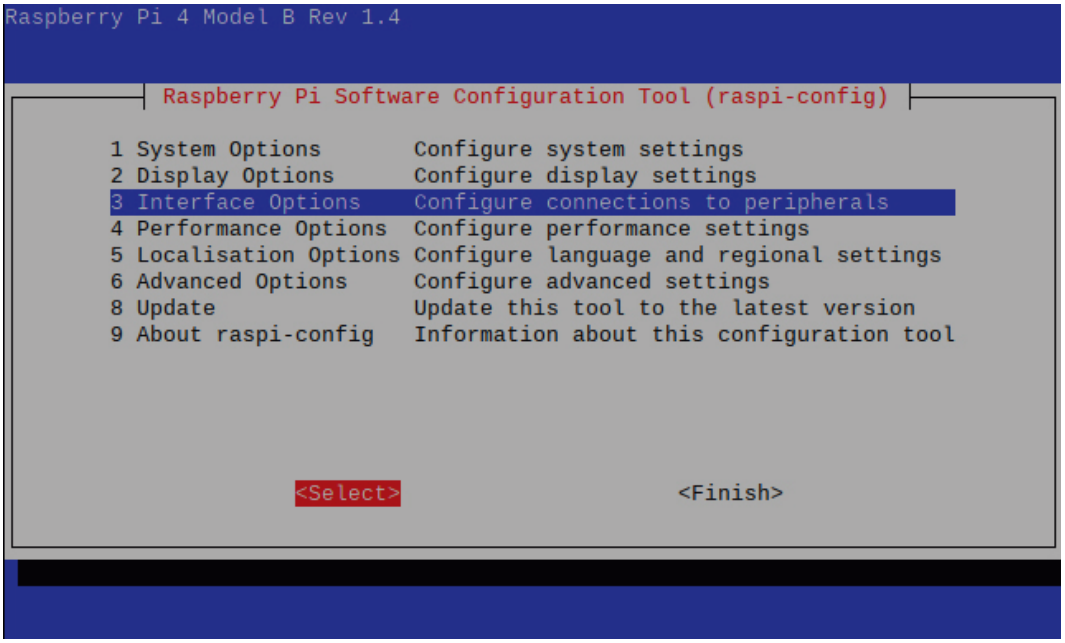
Nos mostrará una lista de puertos, buscaremos el puerto (ttyS0), en el caso que no se encuentre, seguir lo siguientes pasos para habilitarlo



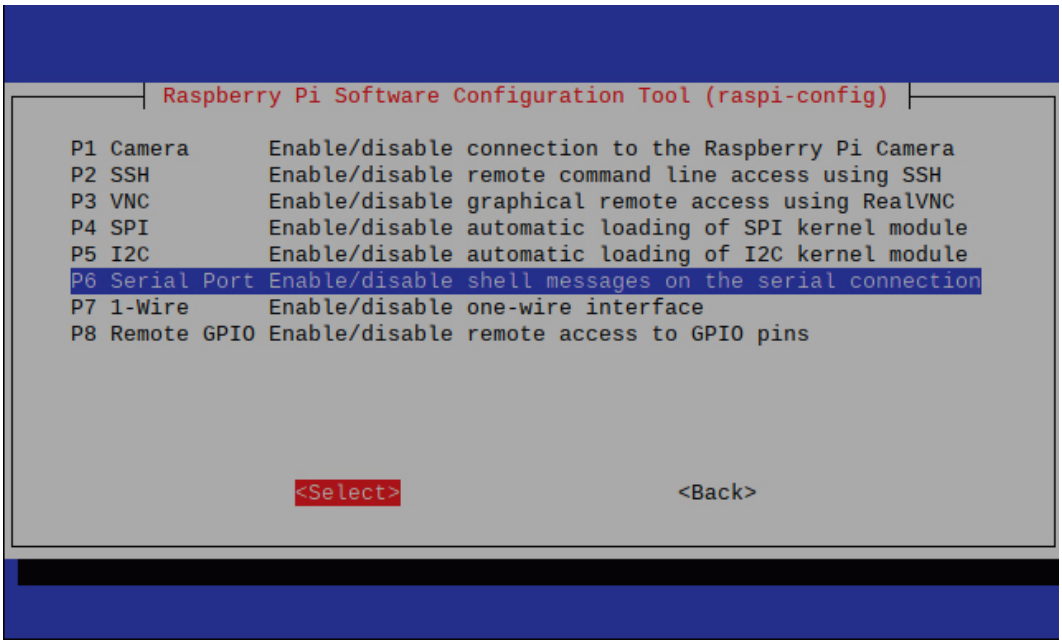
Para habilitar el puerto serial (ttyS0) vamos a ejecutar el siguiente comando sudo raspi-config

e inmediatamente nos abrirá una ventana.

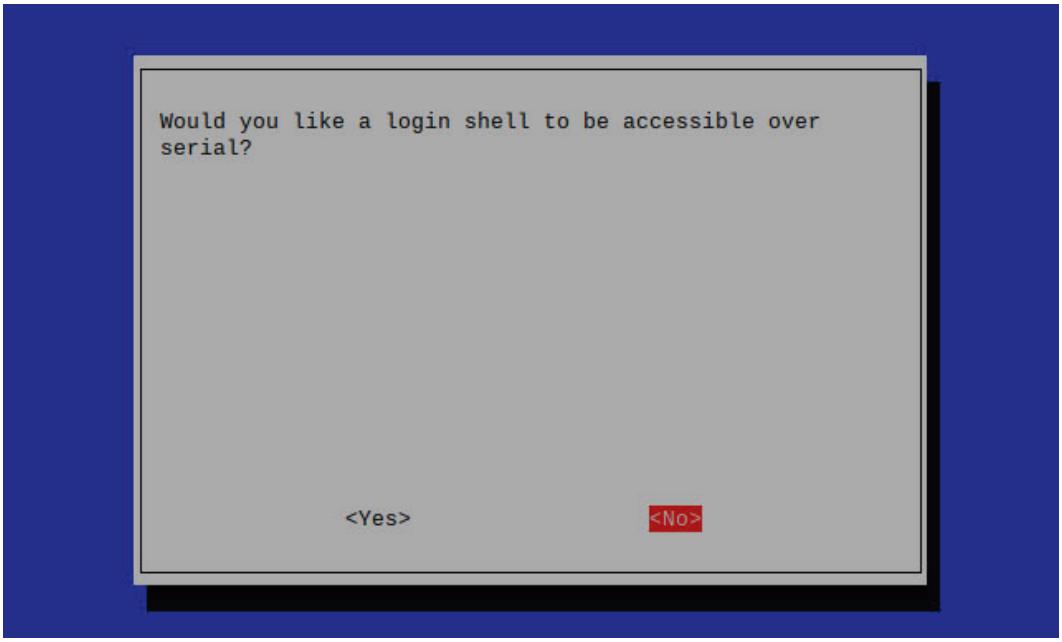
Seleccionamos la opción 3 “Interface Options”, clic en <Select> y damos “ENTER”



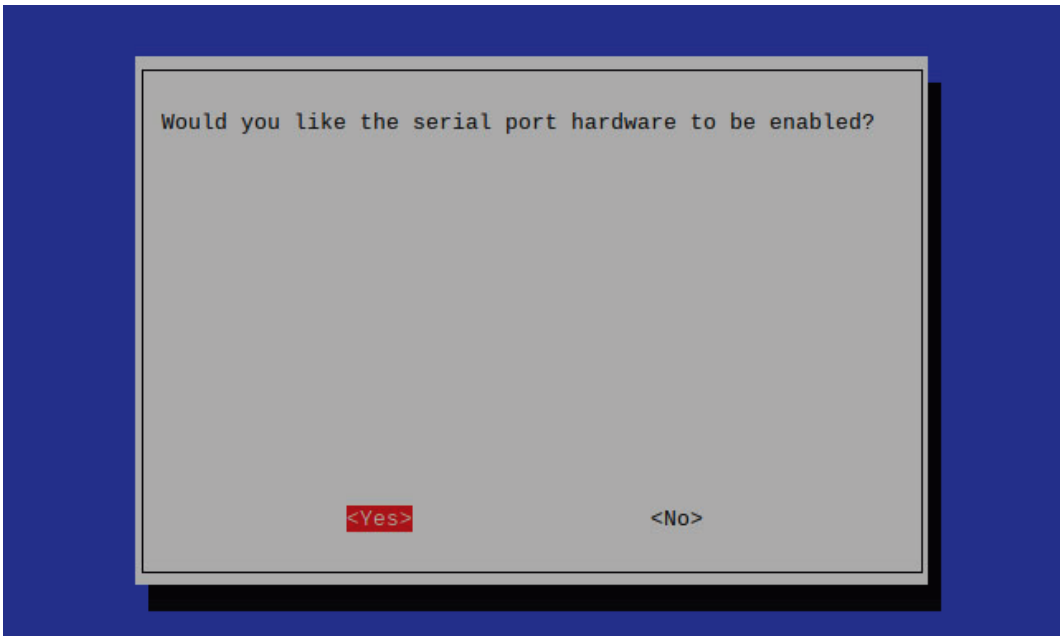
Se nos abrirá la siguiente ventana, seleccionamos P6 “Serial Port”, clic en <Select> y nuevamente damos “ENTER”



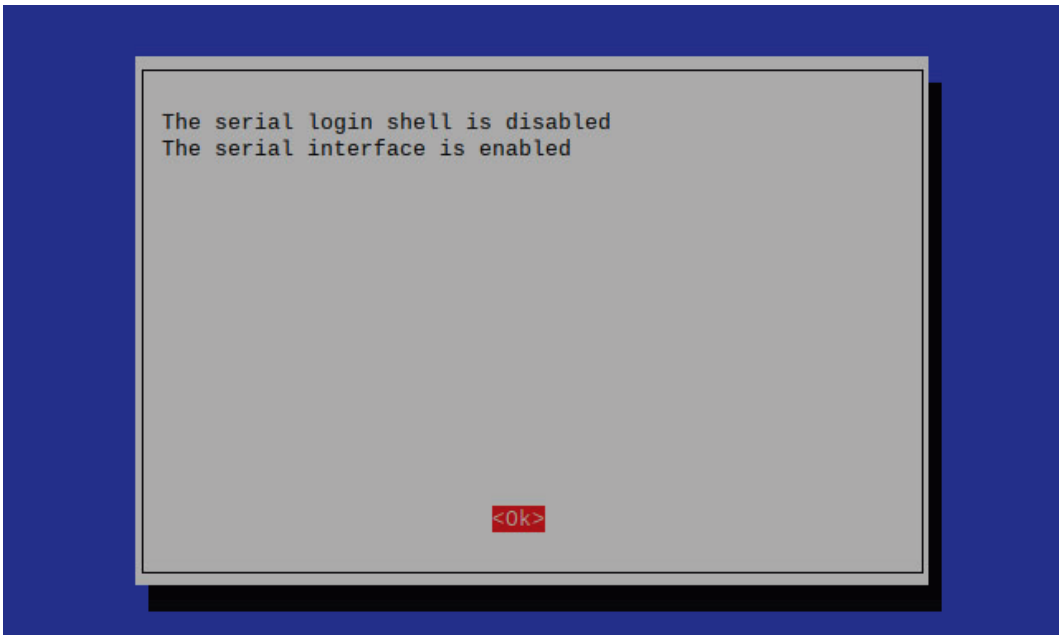
Después mostrará otra ventana donde daremos “ENTER” en “No”.



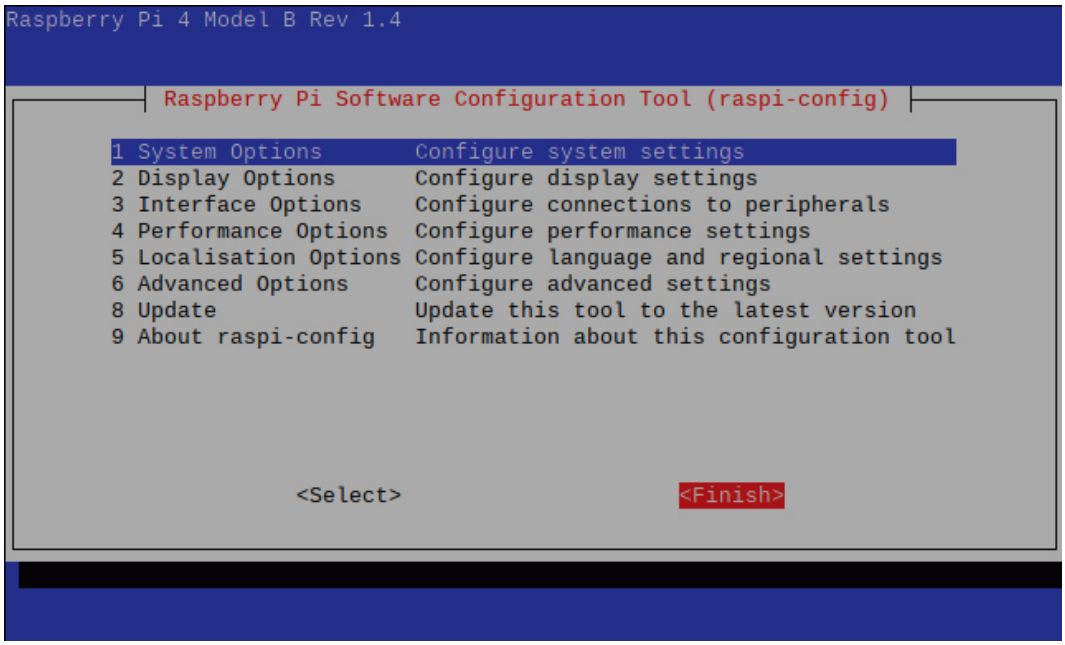
Ahora daremos “ENTER” en Yes para habilitar el puerto serial.



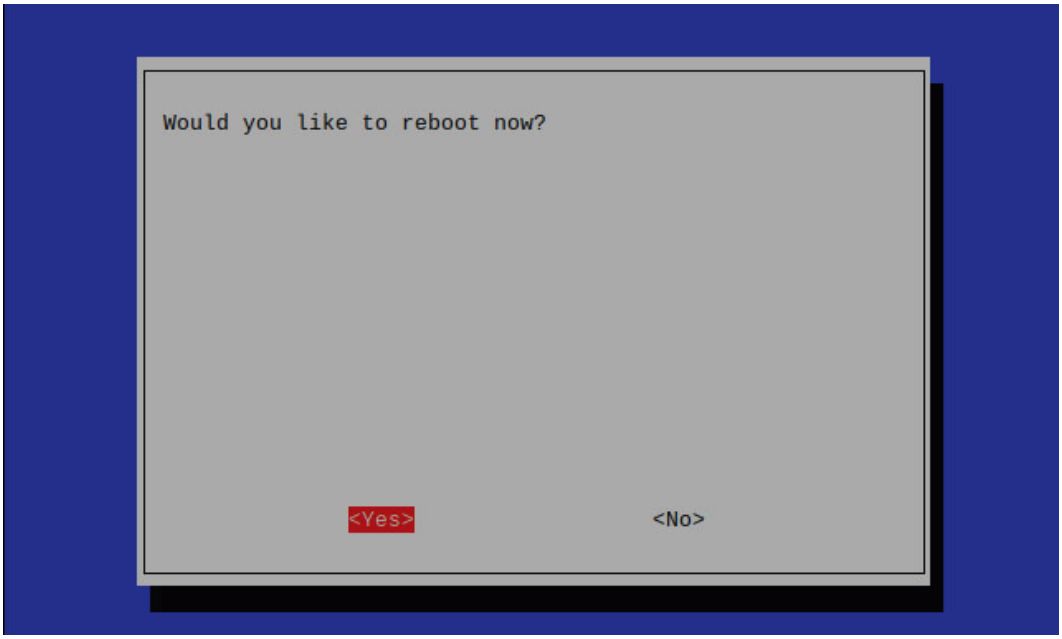
Nos saldrá la siguiente ventana, en la cual mostrará que el puerto serial se ha habilitado, daremos “ENTER” para continuar.



Esto nos regresará a la ventana principal y simplemente nos iremos a la opción <Finish> y daremos “ENTER” para finalizar el proceso.



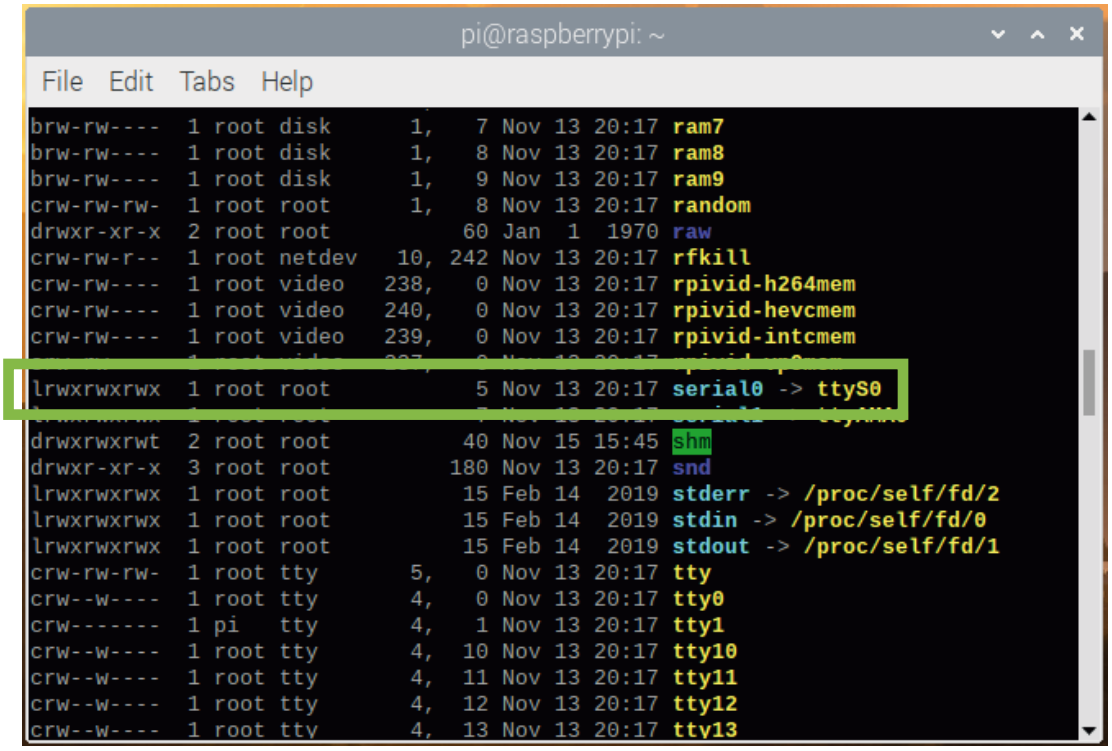
Finalmente daremos “ENTER” en Yes y para ello nos solicitará reiniciar el sistema.



Una vez que encienda nuevamente la Raspberry Pi ejecutamos el comando:

ls -l /dev

para verificar que el puerto serial (ttyS0) este habilitado.

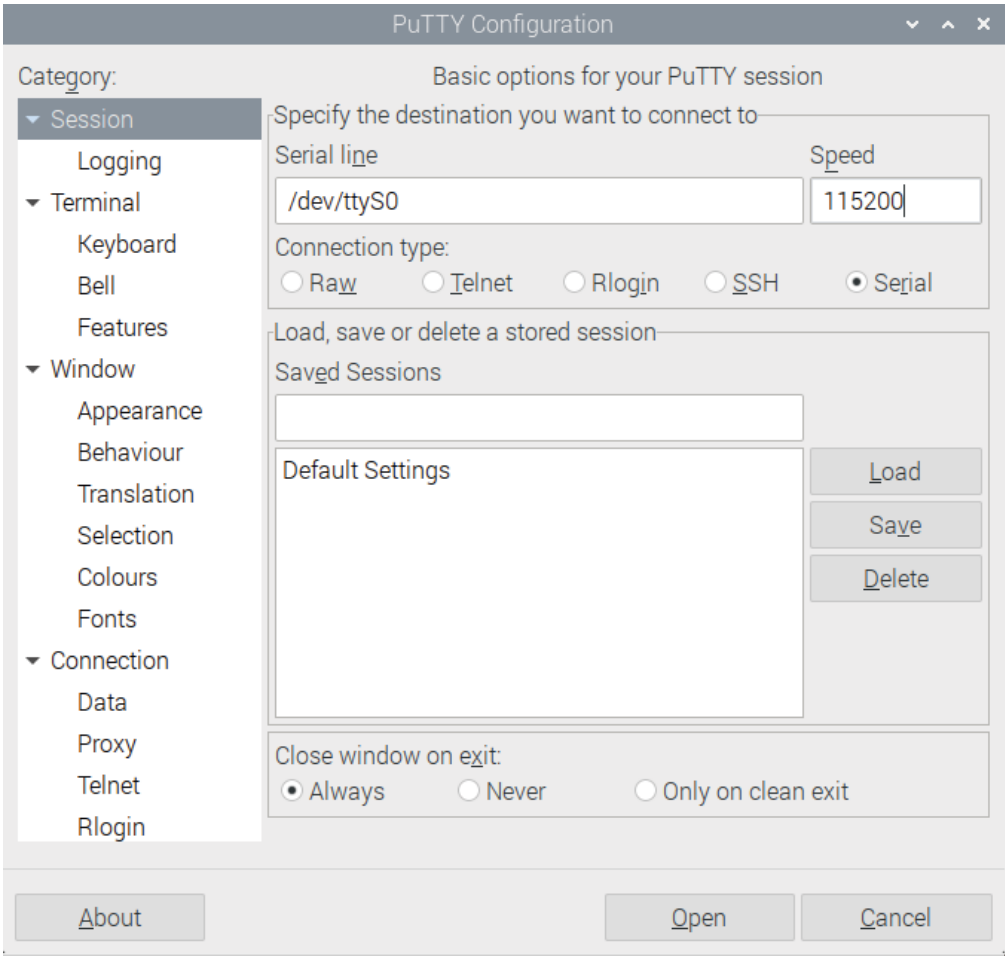


Ejemplo de uso con Terminal PuTTY

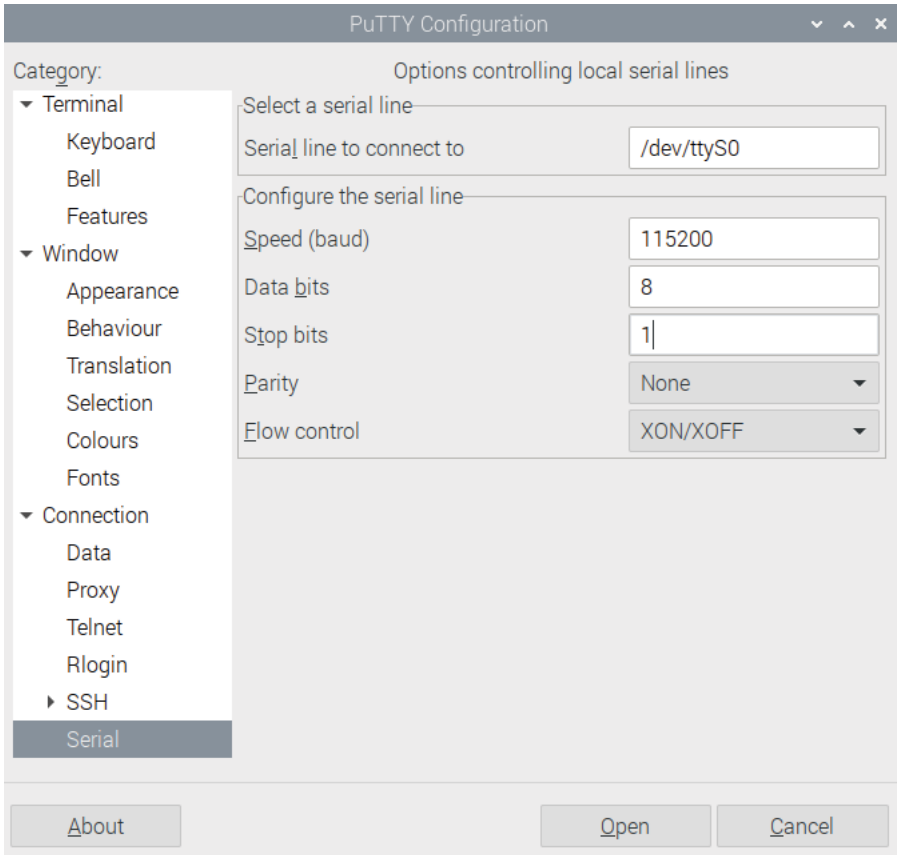
Realizamos la instalación de PuTTY a través del siguiente comando:

sudo apt install putty

Una vez instalado lo abrimos y aparecerá la siguiente ventana donde seleccionaremos “Serial” en tipo de conexión y cambiaremos el “Speed” por 115200.



Para revisar las configuraciones que actualmente se tienen en esta opción, nos desplazamos sobre la columna de la parte izquierda y damos clic sobre “Serial”, verificamos que está configurado de la siguiente manera y después damos clic en Open.



Se nos abrirá la siguiente ventana, en la cual ya podremos ingresar comandos.

Nota: Por motivos de seguridad, los comandos no se podrán visualizar, únicamente se podrán ver las respuestas que nos responda el hardware.

En este ejemplo, vamos a verificar si ya existe comunicación entre un **X-NODE 4 x 24V Input (XN046)** y la **Raspberry Pi**, para ello vamos a enviar el comando:

XN046A?

Como respuesta debemos obtener un OK, confirmando dicha comunicación entre ambos dispositivos .

Ahora ingresamos el comando:

XN046A+G

Esto con el objetivo de mostrar el estado actual de las entradas en el conector, por lo cual mostrará 1 cuando haya señal en la entrada y 0 cuando haya ausencia.

La respuesta que nos regresa es 1000, indicándonos que solo la entrada 1 tiene presencia de señal.

Ejemplo de uso con Python

También es posible enviar comandos a través de código Python, para ello debemos instalarlo con los siguientes comandos:

sudo apt install python3

sudo apt install idle3

Para utilizar los comandos en Python, es necesario instalar la librería Serial, para ello ejecutaremos el siguiente comando:

sudo apt-get install python-serial

Una vez instalado, procedemos abrir Python desde la sección Programing e ingresar el siguiente código:

```
import serial
import time
import subprocess
import os, sys

puerto = serial.Serial(port = '/dev/ttyS0', baudrate = 115200, bytesize = serial.EIGHTBITS,
parity = serial.PARITY_NONE, stopbits = serial.STOPBITS_ONE)

puerto.write('XN007A?'  + '\n\r')
port=puerto.read()
print port
```

En este caso enviaremos el comando XN046A? y como resultado nos dará un OK de confirmación.

Serial.py

```
import serial

puerto = serial.Serial('/dev/ttyS0',
                       baudrate = 115200,
                       bytesize = serial.EIGHTBITS,
                       parity = serial.PARITY_NONE,
                       stopbits = serial.STOPBITS_ONE)


puerto.write(b'XN009A?\n\r')
respuesta=puerto.readline()
print(respuesta)

```

Shell

```
>>> %Run Serial.py
  b'OK\r\n'
>>>
```

Ahoraen nuestro código ingresamos el comando:

XN046A+G

Esto con el objetivo de mostrar el estado actual de las entradas en el conector, por lo cual mostrará 1 cuando haya señal en la entrada y 0 cuando haya ausencia.

Serial.py

```
import serial

puerto = serial.Serial('/dev/ttyS0',
                       baudrate = 115200,
                       bytesize = serial.EIGHTBITS,
                       parity = serial.PARITY_NONE,
                       stopbits = serial.STOPBITS_ONE)


puerto.write(b'XN046A+G\n\r')
respuesta=puerto.readline()
print(respuesta)

```

Shell

```
>>> %Run Serial.py
  b'XN046A=1000\r\n'
>>>
```

Python 3.7.3

La respuesta que nos regresa es 1000, indicándonos que solo la entrada 1 tiene presencia de señal.

XIDE®

www.xide.pro