

Manual de usuario

X-BOARD Raspberry Pi

www.xide.pro

X-BOARD 04





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**.



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™

QW | ST

Compatible con estándar Qwiic° y STEMMA QT°



X-BOARD Raspberry Pi

I. Introducción

La X-BOARD Raspberry Pi es una tarjeta de expansión diseñada principalmente evaluar de forma rápida sensores, actuadores o módulos para 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 l²C 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:

Is -I /dev

						obe	rrypi: ~			×
File Edit	Tabs H	Help								
pi@raspbern	rypi:~ \$	ls -l	/dev							^
total O										1. J
crw-rr	1 root	root	10,	235	Nov	13	20:17	autofs		
drwxr-xr-x	2 root	root		580	Nov	13	20:17	block		
crw	1 root	root	10,	234	Nov	13	20:17	btrfs-control		
drwxr-xr-x	3 root	root		60	Jan	1	1970	bus		
crw	1 root	root	10,	62	Nov	13	20:17	cachefiles		
drwxr-xr-x	2 root	root		3260	Nov	13	20:17	char		
crw	1 root	root	5,	1	Nov	13	20:17	console		
crw	1 root	root	10,	203	Nov	13	20:17	cuse		
drwxr-xr-x	7 root	root		140	Nov	13	20:17	disk		
drwxr-xr-x	2 root	root		80	Jan	1	1970	dma_heap		
drwxr-xr-x	3 root	root		120	Nov	13	20:17	dri		
crw-rw	1 root	video	29,		Nov	13	20:17	fb0		
lrwxrwxrwx	1 root	root		13	Feb	14	2019	<pre>fd -> /proc/self/fd</pre>		
crw-rw-rw-	1 root	root	1,		Nov	13	20:17	full		
crw-rw-rw-	1 root	root	10,	229	Nov	13	20:17	fuse		
crw-rw	1 root	gpio	254,		Nov	13	20:17	gpiochip0		
crw-rw	1 root	gpio	254,	1	Nov	13	20:17	gpiochip1		
crw-rw	1 root	gpio	246,		Nov	13	20:17	gpiomem		
crw	1 root	root	243,		Nov	13	20:17	hidraw0		
crw	1 root	root	243,		Nov	13	20:17	hidraw1		
crw	1 root	root	243,	2	Nov	13	20:17	hidraw2		
crw	1 root	root	243.	3	Nov	13	20:17	hidraw3		-

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

						pi@		pbe	rrypi: ~	× ^ >	ĸ
File Edi	it T	īal	bs H	lelp							
brw-rw		1	root	disk	1,	8	Nov	15	16:19	ram8	1
brw-rw		1	root	disk	1,		Nov	15	16:19	ram9	
crw-rw-rw	N -	1	root	root	1,		Nov	15	16:19	random	
drwxr-xr-		2	root	root		60	Jan	1	1970	raw	
crw-rw-r-		1	root	netdev	10,	242	Nov	15	16:19	rfkill	
crw-rw		1	root	video	238,		Nov	15	16:19	rpivid-h264mem	
crw-rw		1	root	video	240,		Nov	15	16:19	rpivid-hevcmem	
crw-rw		1	root	video	239,		Nov	15	16:19	rpivid-intcmem	
crw-rw		1	root	video	237,		Nov	15	16:19	rpivid-vp9mem	
lrwxrwxrw	XIX	1	root	root			Nov	15	16:19	serial1 -> ttyAMA0	
drwxrwxrw	vt	2	root	root		40	Feb	14	2019	shm	
drwxr-xr-		3	root	root		180	Nov	15	16:19	snd	
lrwxrwxrw	NX N	1	root	root		15	Feb	14	2019	<pre>stderr -> /proc/self/fd/2</pre>	
lrwxrwxrw	NX N	1	root	root		15	Feb	14	2019	<pre>stdin -> /proc/self/fd/0</pre>	
lrwxrwxrw	NX N	1	root	root		15	Feb	14	2019	<pre>stdout -> /proc/self/fd/1</pre>	
crw-rw-rw	N -	1	root	tty	5,		Nov	15	16:19	tty	
crww		1	root	tty	4,		Nov	15	16:19	ttyθ	
crw		1	pi	tty	4,	1	Nov	15	16:19	tty1	
crww		1	root	tty	4,	10	Nov	15	16:19	tty10	
crww		1	root	tty	4,	11	Nov	15	16:19	tty11	
crww		1	root	tty	4,	12	Nov	15	16:19	tty12	
crww		1	root	tty	4,	13	Nov	15	16:19	tty13	
crww		1	root	tty	4,	14	Nov	15	16:19	tty14	
crww		1	root	tty	4,	15	Nov	15	16:19	tty15	-

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"

	Ras	spberı	ry Pi Software (Configuration Tool (raspi-config)
P1	Camera		Enable/disable	connection to the Raspberry Pi Camera
P2	SSH		Enable/disable	remote command line access using SSH
P3	VNC		Enable/disable	graphical remote access using RealVNC
P4	SPI		Enable/disable	automatic loading of SPI kernel module
P5	12C		Enable/disable	automatic loading of I2C kernel module
Ρ6	Serial	Port	Enable/disable	shell messages on the serial connection
P7	1-Wire		Enable/disable	one-wire interface
P8	Remote	GPIO	Enable/disable	remote access to GPIO pins
				(De alla
			<select></select>	<back></back>

Después mostrará otra ventana donde daremos "ENTER" en "No".



Ahora daremos "ENTER" en Yes para habilitar el puerto serial.

Would you l:	ike the serial	port hardware to	be enabled?
	<yes></yes>	<no></no>	

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.

Raspberry	Pi 4 Model B Rev 1.4	
1 2 3 4 5 6 8 9	Raspberry Pi Softwa System Options Display Options Interface Options Performance Options Localisation Options Advanced Options Update About raspi-config	Configure system settings Configure display settings Configure connections to peripherals Configure performance settings Configure language and regional settings Configure advanced settings Update this tool to the latest version Information about this configuration tool
	<select></select>	<finish></finish>

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

Would you like to reboot now	?	
<yes></yes>	<no></no>	

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

ls -l /dev

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

	pi@raspberrypi: ~							^	×
	File Edit	Tabs Help							
	brw-rw	1 root disk	1, 7	Nov 13	20:17	ram7			^
	brw-rw	1 root disk	1, 8	Nov 13	20:17	ram8			
	brw-rw	1 root disk	1, 9	Nov 13	20:17	ram9			
	crw-rw-rw-	1 root root	1, 8	Nov 13	20:17	random			
	drwxr-xr-x	2 root root	60	Jan 1	1970	raw			
	crw-rw-r	1 root netde	v 10, 242	Nov 13	20:17	rfkill			
	crw-rw	1 root video	238, 0	Nov 13	20:17	rpivid-h264mem			
	crw-rw	1 root video	240, 0	Nov 13	20:17	rpivid-hevcmem			
	crw-rw	1 root video	239, 0	Nov 13	20:17	rpivid-intcmem			
r		1	- 227, - 2	10	20,17				
	lrwxrwxrwx	1 root root	5	Nov 13	20:17	serial0 -> ttyS0			
	drwxrwxrwt	2 root root	40	Nov 15	15:45	shm			
	drwxr-xr-x	3 root root	180	Nov 13	20:17	snd	10		
	Lrwxrwxrwx	1 root root	15	Feb 14	2019	stderr -> /proc/self/fd	/2		
	Lrwxrwxrwx	1 root root	15	Feb 14	2019	stdin -> /proc/selt/td/	0		
	Lrwxrwxrwx	1 root root	15	Feb 14	2019	stdout -> /proc/selt/td	/1		
	crw-rw-rw-	1 root tty	5, 0	NOV 13	20:17	tty			
	crww	1 root tty	4, 0	NOV 13	20:17	ttye			
	crw		4, 1	NOV 13	20:17	ttyl			
	Crww	1 root tty	4, 10	NOV 13	20:17				
	CTWW	1 root tty	4, 11	Nov 13	20:17				
	CTWW	1 root tty	4, 12	Nov 13	20:17	tty12			_

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.

	PuTTY Configuration	~ ^ X
Category:	Basic options for your PuTTY sess	sion
- Session	Specify the destination you want to connect to—	
Logging	Serial li <u>n</u> e	S <u>p</u> eed
▼ Terminal	/dev/ttyS0	115200
Keyboard	Connection type:	
Bell	○ Ra <u>w</u> ○ <u>T</u> elnet ○ Rlog <u>i</u> n ○ <u>S</u> SH	● Se <u>r</u> ial
Features	Load, save or delete a stored session	
- Window	Sav <u>e</u> d Sessions	
Appearance		
Behaviour	Default Settings	Load
Translation		Eoud
Selection		Sa <u>v</u> e
Colours		<u>D</u> elete
Fonts		
 Connection 		
Data		
Proxy	Close window on e <u>x</u> it:	
Telnet	Always Only on clear	an exit
Rlogin		
<u>A</u> bout	<u>O</u> pen	<u>C</u> ancel

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.

	PuTTY Configuratio	n	~ ^ X			
Category: ▼ Terminal	Options controlling local serial lines					
Keyboard Bell Features Vindow Appearance Behaviour Translation Selection Colours	Serial line to connect to Configure the serial line Speed (baud) Data bits Stop bits Parity Elow control	/dev/tty 115200 8 1 None XON/X	/SO			
Fonts ▼ Connection Data Proxy Telnet Rlogin ▶ SSH Serial						
<u>A</u> bout		<u>O</u> pen	<u>C</u> ancel			

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')
```

print port

port=puerto.read()

15

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

confirmación.

```
Serial.py 🗙
  1 import serial
  2
  3 puerto = serial.Serial('/dev/ttyS0',
                             baudrate = 115200,
  4
  5
                             bytesize = serial.EIGHTBITS,
  6
                             parity = serial.PARITY NONE,
  7
                             stopbits = serial.STOPBITS ONE)
  8
  9
 10 puerto.write(b'XN009A?\n\r')
 11 respuesta=puerto.readline()
 12 print(respuesta)
 13
 п л
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 🛛
  1
    import serial
     puerto = serial.Serial('/dev/ttyS0',
  3
                             baudrate = 115200,
  4
  5
                             bytesize = serial.EIGHTBITS,
  6
                             parity = serial.PARITY NONE,
  7
                             stopbits = serial.STOPBITS ONE)
  8
  9
 10 puerto.write(b'XN046A+G\n\r')
 11 respuesta=puerto.readline()
 12 print(respuesta)
 13
 п л
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.

K





www.xide.pro