

SKU20218

MODULO DE RELEVADOR INDUSTRIAL DE 8 CANALES DE PROTECCION, P/RASPBERRY PI PICO



DESCRIPCIÓN

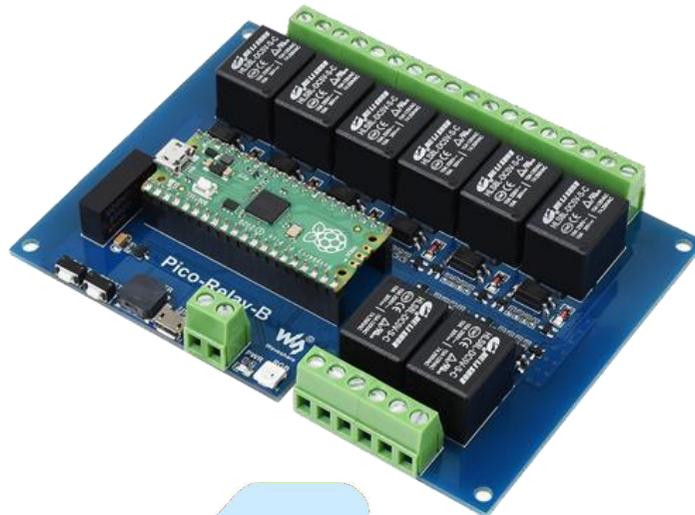
Módulo de Relevador Industrial de 8 canales para Raspberry Pi Pico.
Aislamiento de fuente de alimentación,
Aislamiento De Fotoacopladorchip de refuerzo de buck-dc dc de alta eficiencia.

CARACTERISTICAS

- Aislamiento de la fuente de alimentación unibody a bordo, proporciona un voltaje aislado estable, no necesita fuente de alimentación adicional para el terminal aislado
 - Aislamiento de fotoacoplador integrado, evita interferencias de circuitos externos de alto voltaje conectados al relé
 - Relé de alta calidad, clasificación de contacto: 10A 250V AC o $\leq 10A$ 30V DC
 - Carcasa de protección ABS con soporte de montaje en riel, fácil de instalar, segura de usar
 - El puerto USB Breakout y el pin BOOT facilitan la depuración sin desmontar la carcasa
 - Indicador PWR, LED RGB y zumbador pasivo fuera de la carcasa
 - Viene con *Recursos de desarrollo y manual (ejemplos de Raspberry Pi Pico C / C ++ y MicroPython) [Se incluyen en el link al final de este PDF]
- Voltaje de funcionamiento : 5V
 - Protocolo de Comunicación: USB
 - Interfaz de Comunicación: Micro USB
 - Canal de retransmisión: 8 Canales

 ¿Qué vamos a Innovar hoy?	AG Electrónica SAPI de CV República del Salvador N° 20 2do Piso. Teléfono: 55 5130 - 7210 http://www.agelectronica.com	Realizó	JMLM
		Revisó	JMLM
		Fecha	16/ENE/2023

- Compatibilidad Con El Cabezal Raspberry Pi Pico

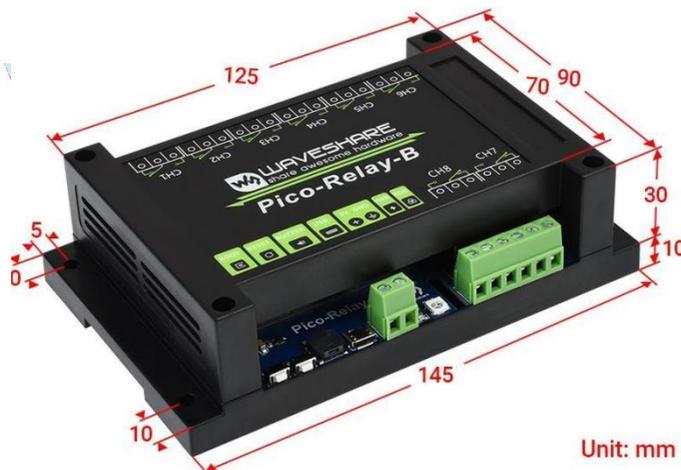


- Carcasa De Protección ABS Con Soporte De Montaje En Riel, Fácil De Instalar, Segura De Usar



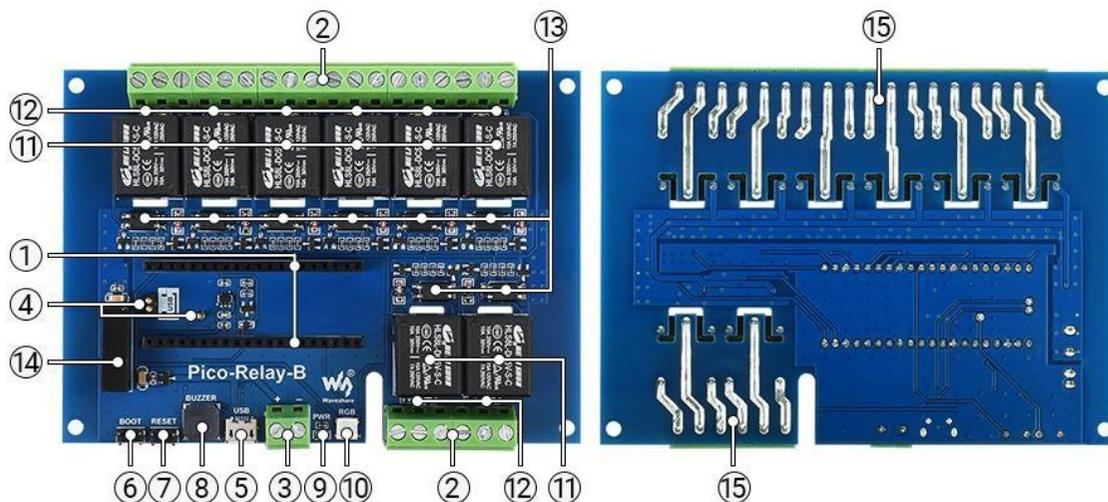
Dimensiones *Electronica*

¿Qué hoy?



 AG Electrónica ¿Qué vamos a Innovar hoy?	Realizó	JMLM
	Revisó	JMLM
	Fecha	16/ENE/2023

SECCIONES DE LA TARJETA



1. Cabecera Raspberry Pi Pico
2. Terminal de tornillo de relé para conectar dispositivos externos
3. Terminal de tornillo de fuente de alimentación los relés múltiples están conectados en paralelo
4. Pines Pogo
5. Puerto de comunicación USB conectado a través de pines pogo
6. Tecla BOOT conectada a través de pines pogo
7. Restablecer tecla Pico reset
8. Frecuencia de sonido controlable del zumbador pasivo
9. Indicador de potencia
10. WS2812 RGB LED
11. Clasificación de contacto de relés de calidad de 8 canales por canal: $\leq 10A$ 250V AC o $\leq 10A$ 30V DC
12. Indicador de funcionamiento del relé
13. El aislamiento del fotoacoplador, evita la interferencia del circuito externo de alto voltaje conectado al relé
14. El aislamiento de la fuente de alimentación, proporciona un voltaje aislado estable, no necesita una fuente de alimentación adicional para el terminal aislado
15. Las pistas engrosadas permiten grandes dispositivos de corriente

¿Qué vamos a innovar hoy?

 ¿Qué vamos a innovar hoy?	Realizó	JMLM
	Revisó	JMLM
	Fecha	16/ENE/2023
AG Electrónica SAPI de CV República del Salvador N° 20 2do Piso. Teléfono: 55 5130 - 7210 http://www.agelectronica.com		

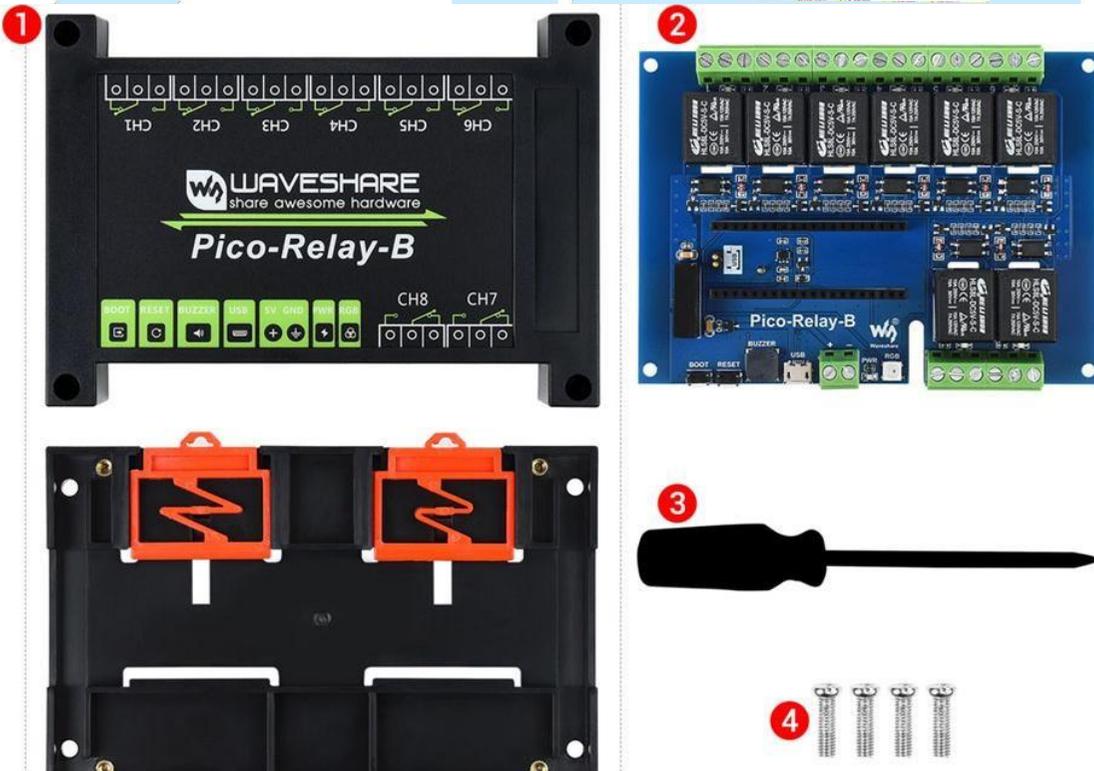
PINES

GP0	1	40	VSYS	Power supply
GP1	2	39	VSYS	
GND	3	38	GND	Ground
GP2	4	37	3V3_EN	
GP3	5	36	3V3(OUT)	
GP4	6	35	ADC_VREF	
GP5	7	34	GP28	
GND	8	33	GND	
GP6	9	32	GP27	
GP7	10	31	GP26	
GP8	11	30	RUN	
GP9	12	29	GP22	
GND	13	28	GND	
GP10	14	27	GP21	
GP11	15	26	GP20	
GP12	16	25	GP19	
GP13	17	24	GP18	
GND	18	23	GND	
GP14	19	22	GP17	
GP15	20	21	GP16	

VSYS		Power supply
GND		Ground
GP6	BUZZER	Buzzer control
GP13	RGB	RGB LED control
GP14	CH8	CH8 relay control pin
GP15	CH7	CH7 relay control pin
GP16	CH6	CH6 relay control pin
GP17	CH5	CH5 relay control pin
GP18	CH4	CH4 relay control pin
GP19	CH3	CH3 relay control pin
GP20	CH2	CH2 relay control pin
GP21	CH1	CH1 relay control pin

CONTIENE

1. Carcasa de protección ABS (superior e inferior) x1
2. Pico-Relé-B x1
3. Destornillador x1
4. Paquete de tornillos x1



*Recursos: [Pico-Relé-B](#)

 <p>AG Electrónica ¿Qué vamos a innovar hoy?</p>	Realizó	JMLM
	Revisó	JMLM
	Fecha	23/SEP/2022

EJEMPLO DE CÓDIGO

```
[ main.py ] * X
1 #Codigo para: MODULO DE RELEVADOR INDUSTRIAL DE8CANALESDEPROTECCION, P/RASPBERRYPI
2 # WAVESHARE Pico-Relay-B
3
4 from machine import Pin, PWM
5 import utime
6
7 # Crea objetos para los relevadores
8
9 ch1_relay = Pin(21, Pin.OUT) # CH_1 Relay pin 21
10 ch2_relay = Pin(20, Pin.OUT) # CH_2 Relay pin 20
11 ch3_relay = Pin(19, Pin.OUT) # CH_3 Relay pin 19
12 ch4_relay = Pin(18, Pin.OUT) # CH_4 Relay pin 18
13 ch5_relay = Pin(17, Pin.OUT) # CH_5 Relay pin 17
14 ch6_relay = Pin(16, Pin.OUT) # CH_6 Relay pin 16
15 ch7_relay = Pin(15, Pin.OUT) # CH_7 Relay pin 15
16 ch8_relay = Pin(14, Pin.OUT) # CH_8 Relay pin 14
17 buzzer = PWM(Pin(6))
18 buzzer.freq(1000)
19
20 def main():
21
22     #Prueba de Relays mediante estados lógicos ON/OFF
23
24     ch1_relay.value(1) #Enciende el Relay CH_1
25     utime.sleep(1)
26     ch1_relay.value(0) #Apaga el Relay CH_1
27     utime.sleep(1)
28
29     ch2_relay.value(1) #Enciende el Relay CH_2
30     utime.sleep(1)
31     ch2_relay.value(0) #Apaga el Relay CH_2
32     utime.sleep(1)
33
34     ch3_relay.value(1) #Enciende el Relay CH_3
35     utime.sleep(1)
36     ch3_relay.value(0) #Apaga el Relay CH_3
37     utime.sleep(1)
38
39     ch4_relay.value(1) #Enciende el Relay CH_4
40     utime.sleep(1)
41     ch4_relay.value(0) #Apaga el Relay CH_4
42     utime.sleep(1)
43
44     ch5_relay.value(1) #Enciende el Relay CH_5
45     utime.sleep(1)
46     ch5_relay.value(0) #Apaga el Relay CH_5
47     utime.sleep(1)
48
49     ch6_relay.value(1) #Enciende el Relay CH_6
50     utime.sleep(1)
51     ch6_relay.value(0) #Apaga el Relay CH_6
52     utime.sleep(1)
53
54     ch7_relay.value(1) #Enciende el Relay CH_7
55     utime.sleep(1)
56     ch7_relay.value(0) #Apaga el Relay CH_7
57     utime.sleep(1)
58
59     ch8_relay.value(1) #Enciende el Relay CH_8
60     utime.sleep(1)
61     ch8_relay.value(0) #Apaga el Relay CH_8
62     utime.sleep(1)
63
64     buzzer.duty_u16(500) #Suena Buzzer durante dos segundos al terminar la prueba de Relay's
65     utime.sleep(2)
66     buzzer.duty_u16(0)
67
68
69 if __name__ == '__main__':
70     main()
```