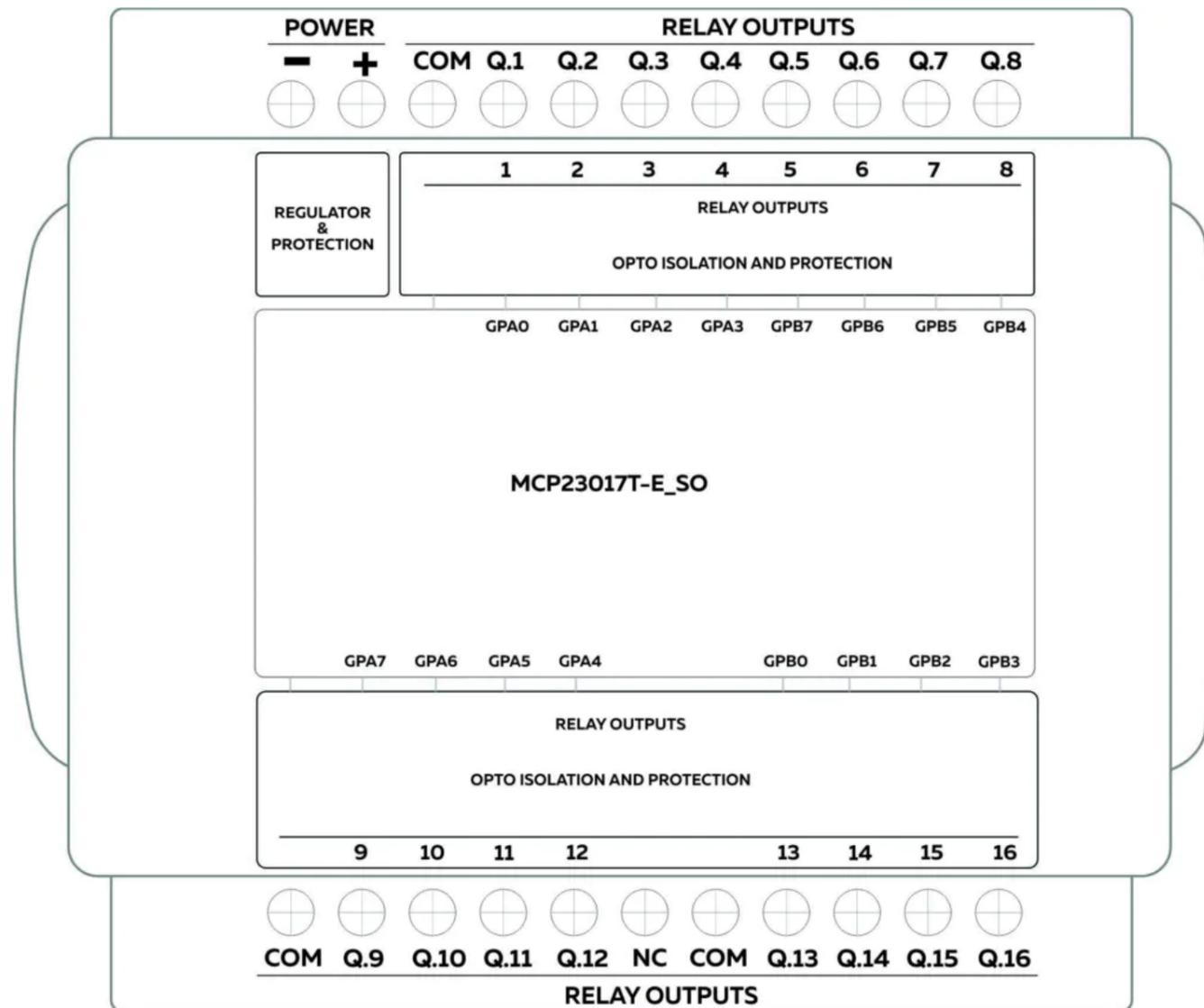


NORVI-EX-R16 – FICHA TÉCNICA

Características del Producto



- Salidas a Relé Normalmente Abiertas
- Cargas de hasta 5 A 30 VCC / 250 VCA resistivas
- Expansor de Puerto MCP23008
- Dirección configurable mediante interruptor DIP

Expansiones Admitidas

- Salidas Relé

Inicio

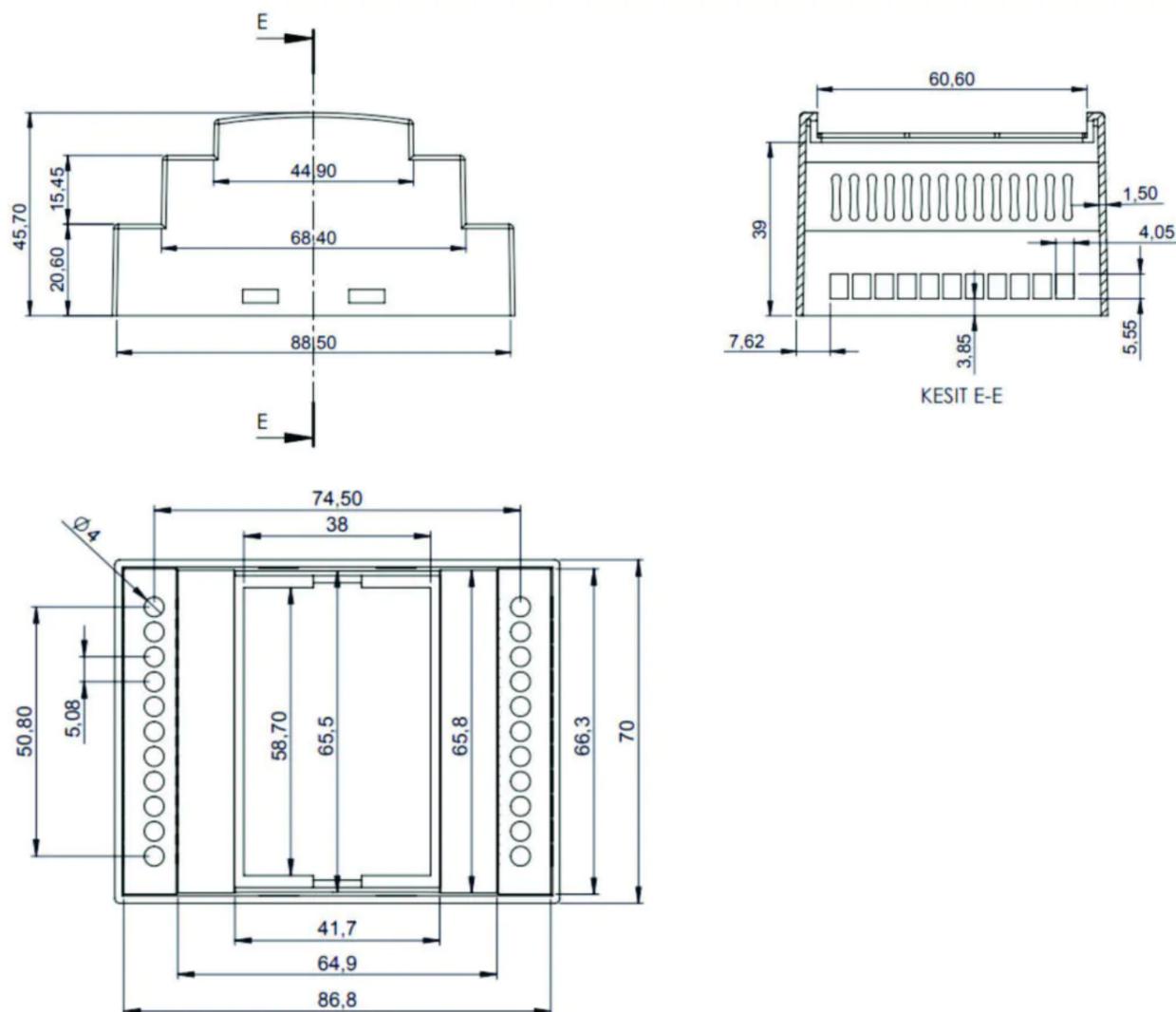
Gama de Producto	NORVI Expansion
Tipo de Producto	Modulo de Expansión
Certificaciones	EN 61131-2:2007 EN 61010-1:2010+A1:2019 EN IEC 61010-2-201:2018 2014/30/EU- Compatibilidad Electromagnética (EMC) Anexo III, Parte B, Modulo C
Tensión Nominal de Alimentación	24V DC
Comunicación	I2C
Entradas y Salidas	16 x Salidas Relé
Pantallas e Indicadores Visuales	1 LED verde

Complementarios

Producto Código Unificado	NORVI EX-R16
Números de Pieza de Producto	NORVI EX-R16

Características Mecánicas

Carcasa	NORVI 202
Montaje / Método de Instalación	RIEL DIN / PESTAÑAS DE MONTAJE
Tipo de Terminal	Carril tipo sombrero TH35-15 conforme a IEC 60715 Carril tipo sombrero TH35-7.5 conforme a IEC 60715 Placa o panel con kit de fijación
Disposición de Terminales	Arriba y abajo
Largo	59.00 mm
Alto	86.00 mm
Ancho	35.00 mm



Entorno

Grado de Protección IP	IP20
Altitud de Operatividad	0 – 2000 metros
Temperatura de Operación	– –10 ... +85° C (14...185 °F)
Altitud de Almacenaje	0 – 3000 metros
Resistencia a Golpes	15 gn para 11ms
Resistencia a Descargas Electrostáticas	4kV en contacto 8kV en aire
Resistencia a Campos Electromagnéticos	10 V/m (80 MHz 1GHz) 3 V/m (1.4 MHz 2 GHz) 1 V/m (2 MHz 3 GHz)

Características Eléctricas

Dispositivos con Alimentación de Red

Tensión de Alimentación (V)	24V DC
Consumo Corriente (mA)	400mA
Fuente de Alimentación Recomendada	1A 24V DC

Procesamiento

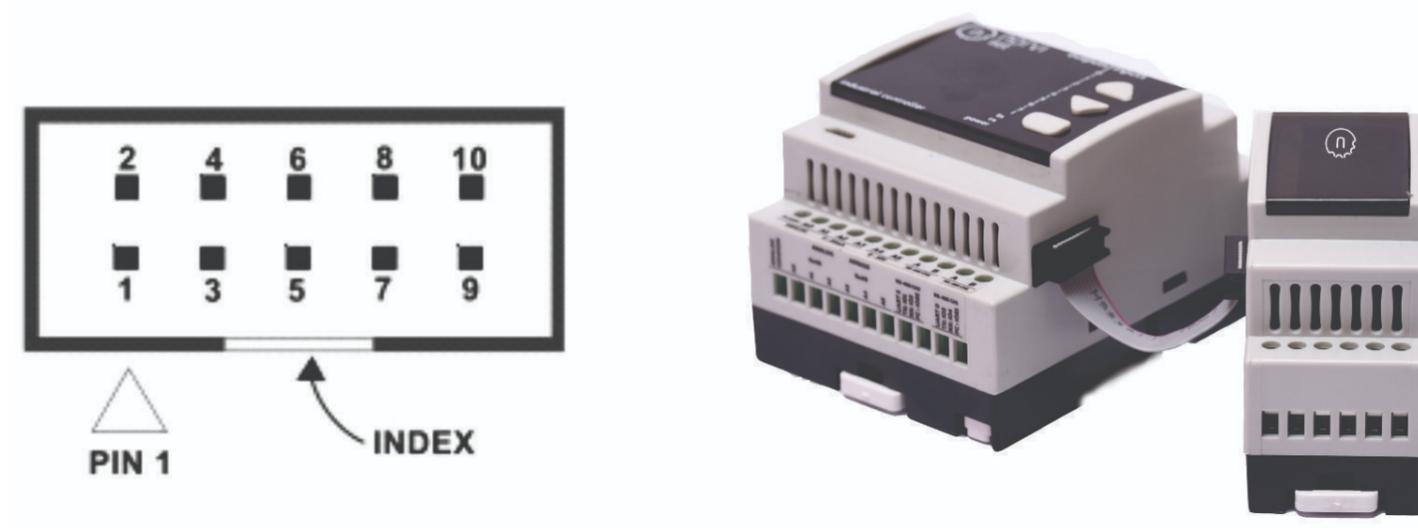
SOC / MCU	Expansor de I/O (In/Out) de 16 bits con interfaz serie
Modulo	MCP23017T-E_SO
Tensión de alimentación de funcionamiento	2.4V a 3.6V
Resolucion	16bit
I2C SDA	GPIO16
I2C SCL	GPIO17

ENTRADAS y SALIDAS

Salidas Relé

Número de Salidas Relé	16
Tipo de Salida Relé	Normally Open / SPST / Electro-mechanical
Corriente Nominal de Contacto (Resistiva)	5 A 30V DC/250V AC
Tensión Máxima de Contacto	270V AC, 125V DC
Frecuencia Máxima de Conmutación	60 Hz
Disposición de Terminales	Salida Relé 0 – GPA0 Salida Relé 1 – GPA1 Salida Relé 2 – GPA2 Salida Relé 3 – GPA3 Salida Relé 4 – GPB7 Salida Relé 5 – GPB6 Salida Relé 6 – GPB5 Salida Relé 7 – GPB4 Salida Relé 8 – GPA7 Salida Relé 9 – GPA6 Salida Relé 10 – GPA5 Salida Relé 11 – GPA4 Salida Relé 12 – GPB0 Salida Relé 13 – GPB1 Salida Relé 14 – GPB2 Salida Relé 15 – GPB3

Puerto de Expansión



PIN	Conexión ESP32
1	TXD0
2	GPIO25
3	RXD0
4	AUX1
5	GPIO15
6	NRST
7	SCL2
8	AUX2
9	SDA2
10	GND

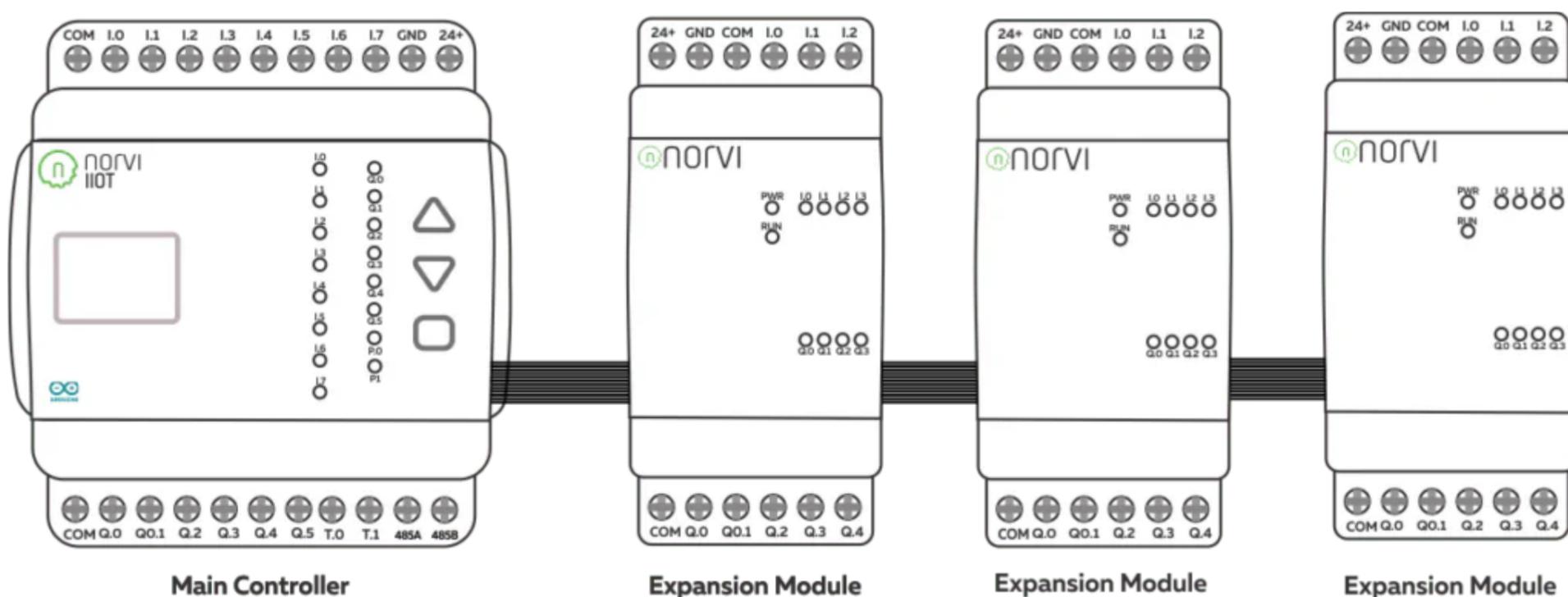
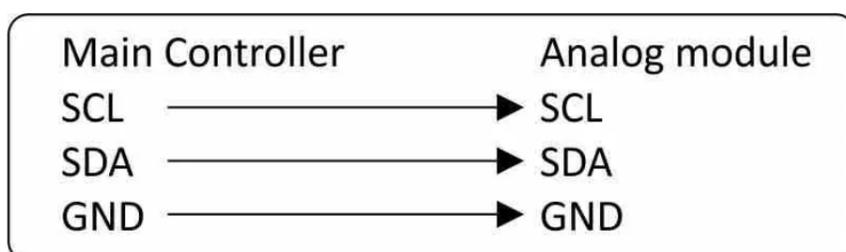
NORVI-EX-R16 – GUIA DEL USUARIO

Programación

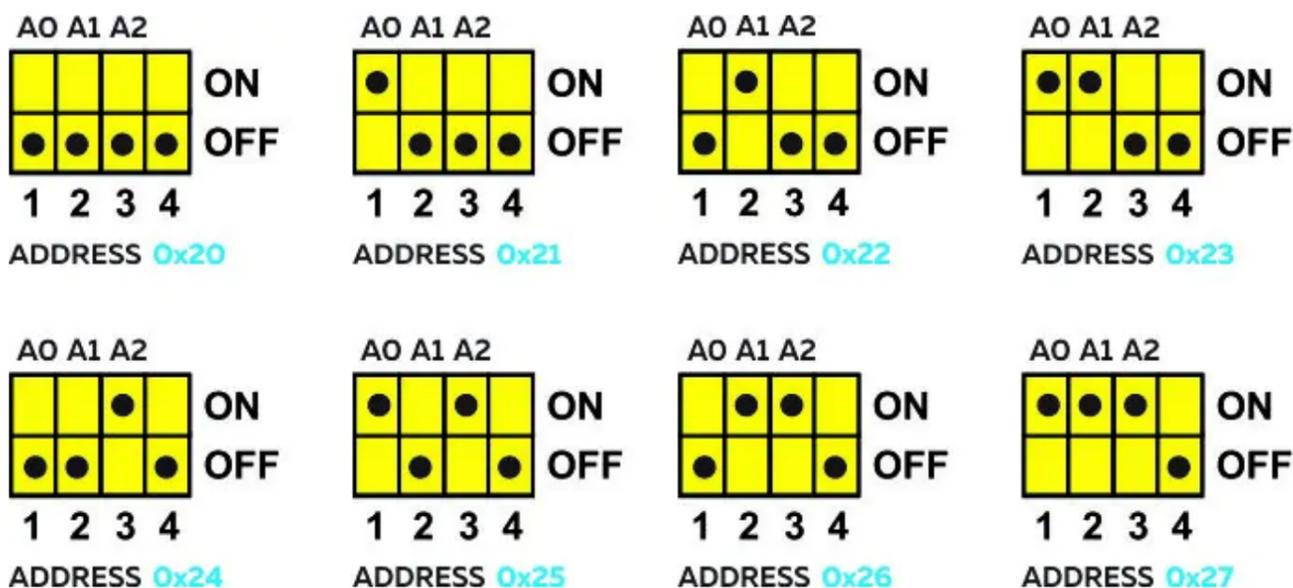
El NORVI-EX-R16 utiliza MCP23017 a través de comunicación I2C. Las direcciones I2C de los dispositivos pueden configurarse mediante los interruptores DIP en la parte inferior del controlador.

Puerto de Expansión

El puerto de expansión de los controladores NORVI IIOT puede utilizarse para conexiones de sensores externos donde se requieren conexiones GPIO directas, o pueden usarse para conectar módulos de expansión NORVI.



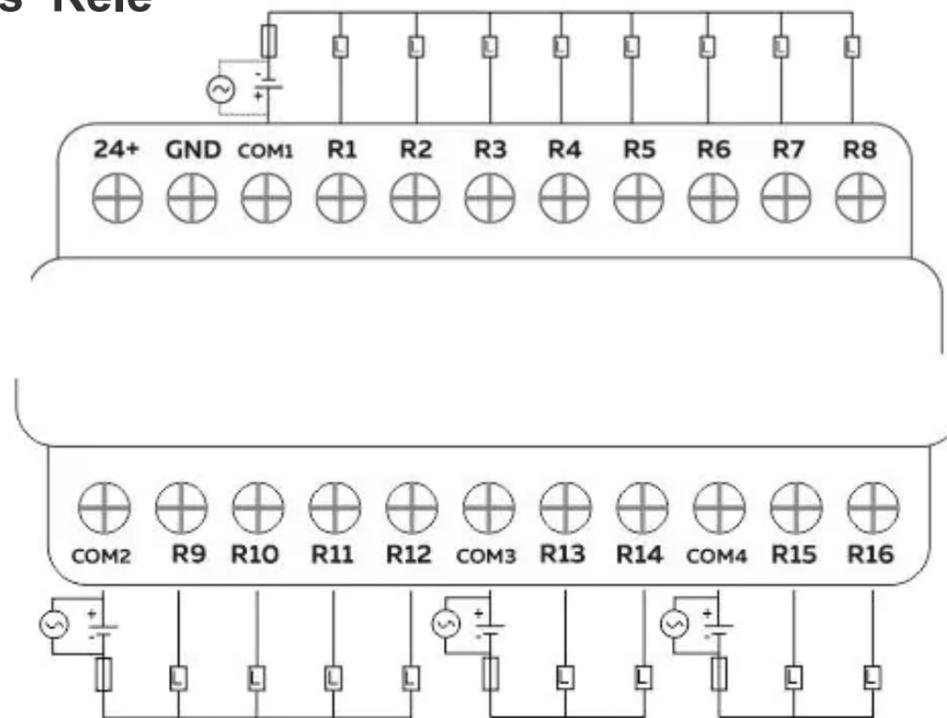
Configuración de la Dirección I2C



La dirección I2C del módulo de expansión puede configurarse cambiando los interruptores DIP en la parte inferior del módulo de expansión. El dispositivo puede configurarse a 8 direcciones I2C mediante los primeros 3 interruptores DIP.

Salidas Relé

Cableado de Salidas Relé



Cableado de salidas relé NORVI EX-R16

Programación de Salidas Relé

Escribir HIGH al GPIO relevante del MCP23017 activa la salida a relé. Consulte la Tabla de Asignación de GPIO para el GPIO de la Salida a Relé. Consulte la configuración de la dirección I2C para configurar la dirección I2C de la expansión.

```
#include <Adafruit_MCP23X17.h>
#define OUTPUT1 0
#define OUTPUT2 1
#define OUTPUT3 2
#define OUTPUT4 3
#define OUTPUT5 12
#define OUTPUT6 13
#define OUTPUT7 14
#define OUTPUT8 15
#define OUTPUT9 7
#define OUTPUT10 6
#define OUTPUT11 5
#define OUTPUT12 4
#define OUTPUT13 8
#define OUTPUT14 9
#define OUTPUT15 10
#define OUTPUT16 11
// uncomment appropriate line
//Adafruit_MCP23X08 mcp;
Adafruit_MCP23X17 mcp;
void setup() {
  Serial.begin(115200);
  Serial.println("NORVO Expansions Test");      //while (!Serial);
  Wire.begin (16, 17);                          // uncomment appropriate mcp.begin
  if (!mcp.begin_I2C(0x27)) {
    Serial.println("Error.");
    while (1);
  }
  mcp.pinMode(OUTPUT1, OUTPUT);                  // configure pin for output
  mcp.pinMode(OUTPUT2, OUTPUT);
  mcp.pinMode(OUTPUT3, OUTPUT);
  mcp.pinMode(OUTPUT4, OUTPUT);
  mcp.pinMode(OUTPUT5, OUTPUT);
```

```

mcp.pinMode(OUTPUT6, OUTPUT);
mcp.pinMode(OUTPUT7, OUTPUT);
mcp.pinMode(OUTPUT8, OUTPUT);
mcp.pinMode(OUTPUT9, OUTPUT);
mcp.pinMode(OUTPUT10, OUTPUT);
mcp.pinMode(OUTPUT11, OUTPUT);
mcp.pinMode(OUTPUT12, OUTPUT);
mcp.pinMode(OUTPUT13, OUTPUT);
mcp.pinMode(OUTPUT14, OUTPUT);
mcp.pinMode(OUTPUT15, OUTPUT);
mcp.pinMode(OUTPUT16, OUTPUT);
Serial.println("Looping...");
}
void loop() {
  mcp.digitalWrite(OUTPUT1, HIGH);
  mcp.digitalWrite(OUTPUT2, LOW);
  mcp.digitalWrite(OUTPUT3, LOW);
  mcp.digitalWrite(OUTPUT4, LOW);
  mcp.digitalWrite(OUTPUT5, LOW);
  mcp.digitalWrite(OUTPUT6, LOW);
  mcp.digitalWrite(OUTPUT7, LOW);
  mcp.digitalWrite(OUTPUT8, LOW);
  delay(500);
  mcp.digitalWrite(OUTPUT1, LOW);
  mcp.digitalWrite(OUTPUT2, HIGH);
  mcp.digitalWrite(OUTPUT3, LOW);
  mcp.digitalWrite(OUTPUT4, LOW);
  mcp.digitalWrite(OUTPUT5, LOW);
  mcp.digitalWrite(OUTPUT6, LOW);
  mcp.digitalWrite(OUTPUT7, LOW);
  mcp.digitalWrite(OUTPUT8, LOW);
  delay(500);
  mcp.digitalWrite(OUTPUT1, LOW);
  mcp.digitalWrite(OUTPUT2, LOW);
  mcp.digitalWrite(OUTPUT3, HIGH);
  mcp.digitalWrite(OUTPUT4, LOW);
  mcp.digitalWrite(OUTPUT5, LOW);
  mcp.digitalWrite(OUTPUT6, LOW);
  mcp.digitalWrite(OUTPUT7, LOW);
  mcp.digitalWrite(OUTPUT8, LOW);
  delay(500);
  mcp.digitalWrite(OUTPUT1, LOW);
  mcp.digitalWrite(OUTPUT2, LOW);
  mcp.digitalWrite(OUTPUT3, LOW);
  mcp.digitalWrite(OUTPUT4, HIGH);
  mcp.digitalWrite(OUTPUT5, LOW);
  mcp.digitalWrite(OUTPUT6, LOW);
  mcp.digitalWrite(OUTPUT7, LOW);
  mcp.digitalWrite(OUTPUT8, LOW);
  delay(500);
  mcp.digitalWrite(OUTPUT1, LOW);
  mcp.digitalWrite(OUTPUT2, LOW);
  mcp.digitalWrite(OUTPUT3, LOW);
  mcp.digitalWrite(OUTPUT4, LOW);
  mcp.digitalWrite(OUTPUT5, HIGH);
  mcp.digitalWrite(OUTPUT6, LOW);
  mcp.digitalWrite(OUTPUT7, LOW);
  mcp.digitalWrite(OUTPUT8, LOW);
}

```

```

delay(500);
mcp.digitalWrite(OUTPUT1, LOW);
mcp.digitalWrite(OUTPUT2, LOW);
mcp.digitalWrite(OUTPUT3, LOW);
mcp.digitalWrite(OUTPUT4, LOW);
mcp.digitalWrite(OUTPUT5, LOW);
mcp.digitalWrite(OUTPUT6, HIGH);
mcp.digitalWrite(OUTPUT7, LOW);
mcp.digitalWrite(OUTPUT8, LOW);
delay(500);
mcp.digitalWrite(OUTPUT1, LOW);
mcp.digitalWrite(OUTPUT2, LOW);
mcp.digitalWrite(OUTPUT3, LOW);
mcp.digitalWrite(OUTPUT4, LOW);
mcp.digitalWrite(OUTPUT5, LOW);
mcp.digitalWrite(OUTPUT6, LOW);
mcp.digitalWrite(OUTPUT7, HIGH);
mcp.digitalWrite(OUTPUT8, LOW);
delay(500);
mcp.digitalWrite(OUTPUT1, LOW);
mcp.digitalWrite(OUTPUT2, LOW);
mcp.digitalWrite(OUTPUT3, LOW);
mcp.digitalWrite(OUTPUT4, LOW);
mcp.digitalWrite(OUTPUT5, LOW);
mcp.digitalWrite(OUTPUT6, LOW);
mcp.digitalWrite(OUTPUT7, LOW);
mcp.digitalWrite(OUTPUT8, HIGH);
delay(500);
mcp.digitalWrite(OUTPUT1, LOW);
mcp.digitalWrite(OUTPUT2, LOW);
mcp.digitalWrite(OUTPUT3, LOW);
mcp.digitalWrite(OUTPUT4, LOW);
mcp.digitalWrite(OUTPUT5, LOW);
mcp.digitalWrite(OUTPUT6, LOW);
mcp.digitalWrite(OUTPUT7, LOW);
mcp.digitalWrite(OUTPUT8, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, HIGH);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, HIGH);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, HIGH);

```

```

mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, HIGH);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, HIGH);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, HIGH);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, HIGH);
mcp.digitalWrite(OUTPUT16, LOW);
delay(500);
mcp.digitalWrite(OUTPUT9, LOW);
mcp.digitalWrite(OUTPUT10, LOW);
mcp.digitalWrite(OUTPUT11, LOW);
mcp.digitalWrite(OUTPUT12, LOW);
mcp.digitalWrite(OUTPUT13, LOW);
mcp.digitalWrite(OUTPUT14, LOW);
mcp.digitalWrite(OUTPUT15, LOW);
mcp.digitalWrite(OUTPUT16, HIGH);
delay(500);
Serial.println("Looping...");
}

```

