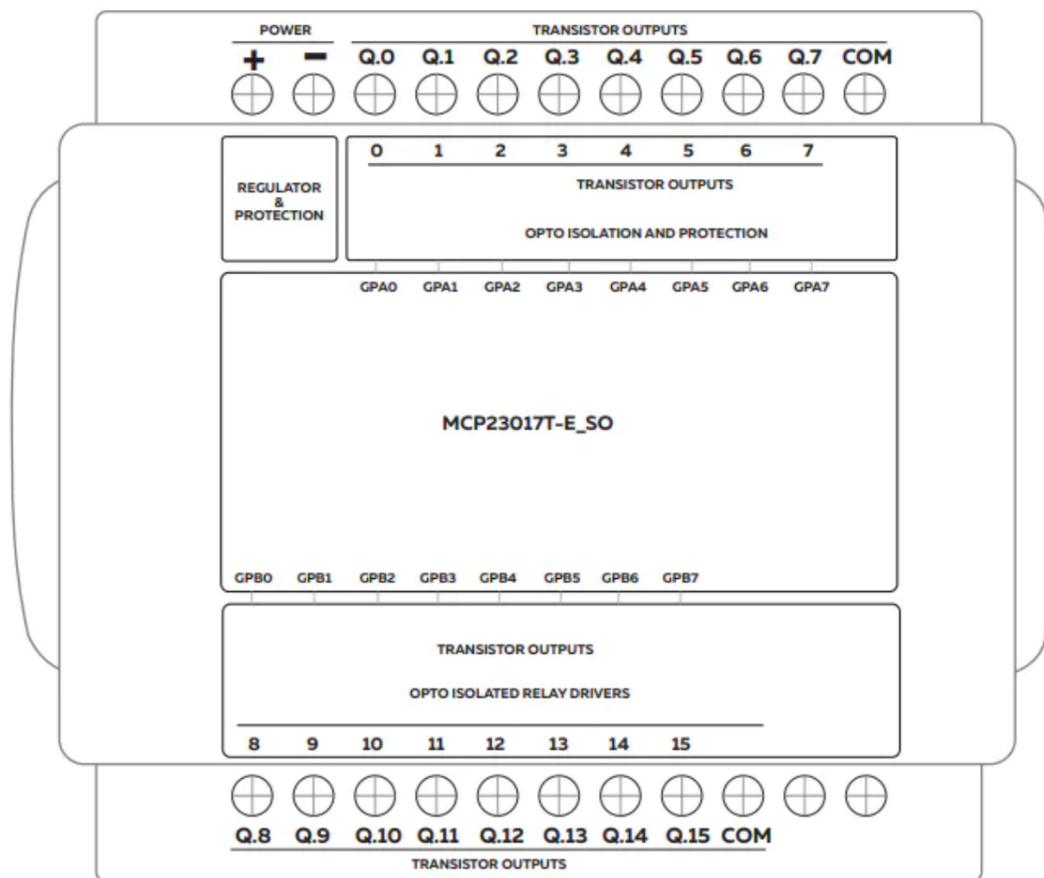


NORVI-EX-Q16 – FICHA TÉCNICA

Características del Producto



- Salidas a Transistor Ópticamente Aisladas
- Soporta cargas de hasta 36V CC
- Disipación de Potencia de 300mW
- Expansor de Puerto MCP23008
- Dirección configurable mediante interruptor DIP

Expansiones Admitidas

- Salidas a Transistor de Colector Abierto



Inicio

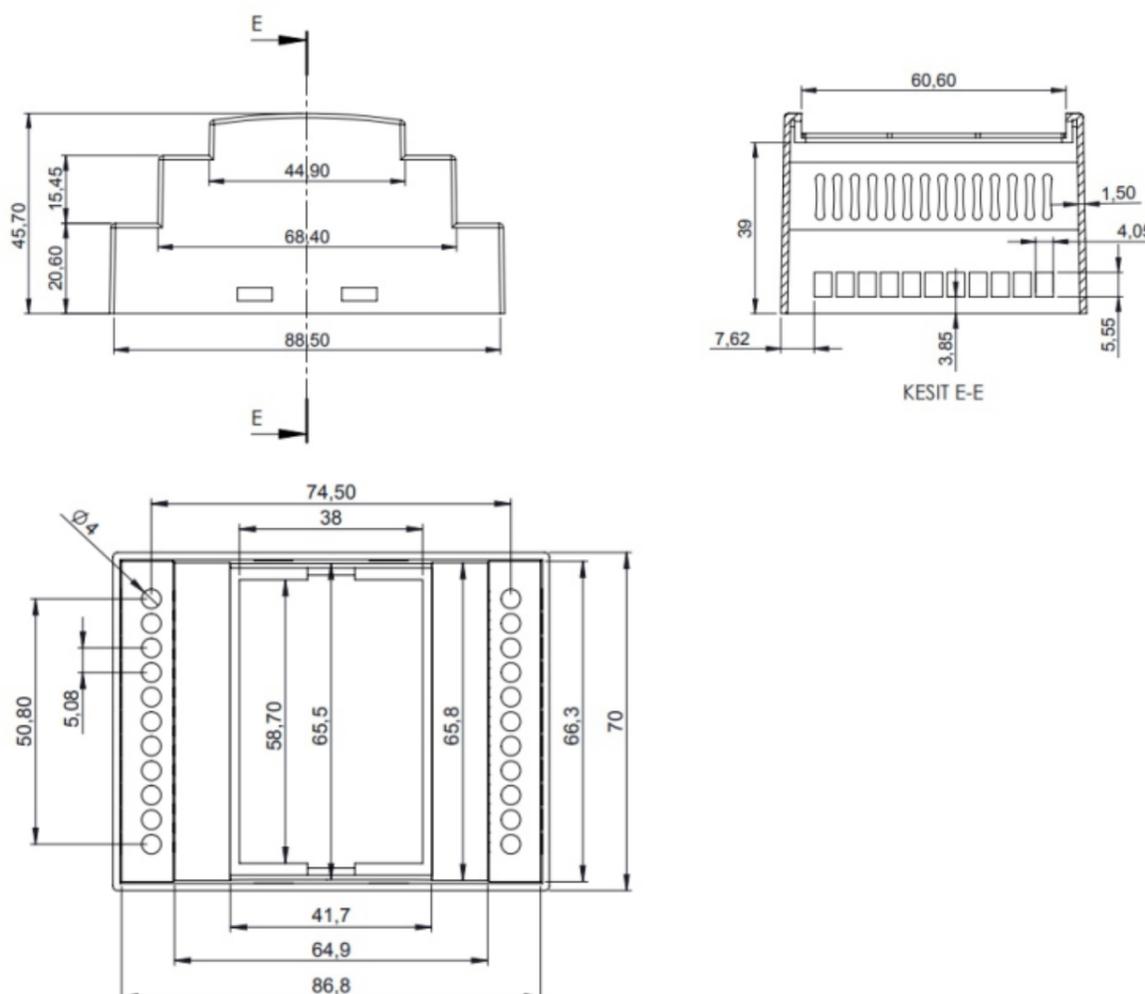
Gama de Producto	NORVI Expansion
Tipo de Producto	Modulo de Expansi3n
Certificaciones	EN 61131-2:2007 EN 61010-1:2010+A1:2019 EN IEC 61010-2-201:2018 2014/30/EU- Compatibilidad Electromagn3tica (EMC) Anexo III, Parte B, Modulo C
Tensi3n Nominal de Alimentaci3n	24V DC
Comunicaci3n	I2C
Entradas y Salidas	16 x Salidas a Transistor de Colector Abierto
Pantallas e Indicadores Visuales	1 LED verde

Complementarios

Producto C3digo Unificado	NORVI EX-Q16
N3meros de Pieza de Producto	NORVI EX-Q16

Características Mecánicas

Carcasa	NORVI 204
Montaje / M3todo de Instalaci3n	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 fijaci3n
Disposici3n de Terminales	Arriba y Abajo
Largo	90.50 mm
Alto	56.60 mm
Ancho	60.60 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 Electroestáticas	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 16-bit-I/O (In/Out) con interfaz serie
Modulo	MCP23017T-E_SO
Tensión de Alimentación	2.4V a 3.6V
I2C SDA	GPIO16
I2C SCL	GPIO17

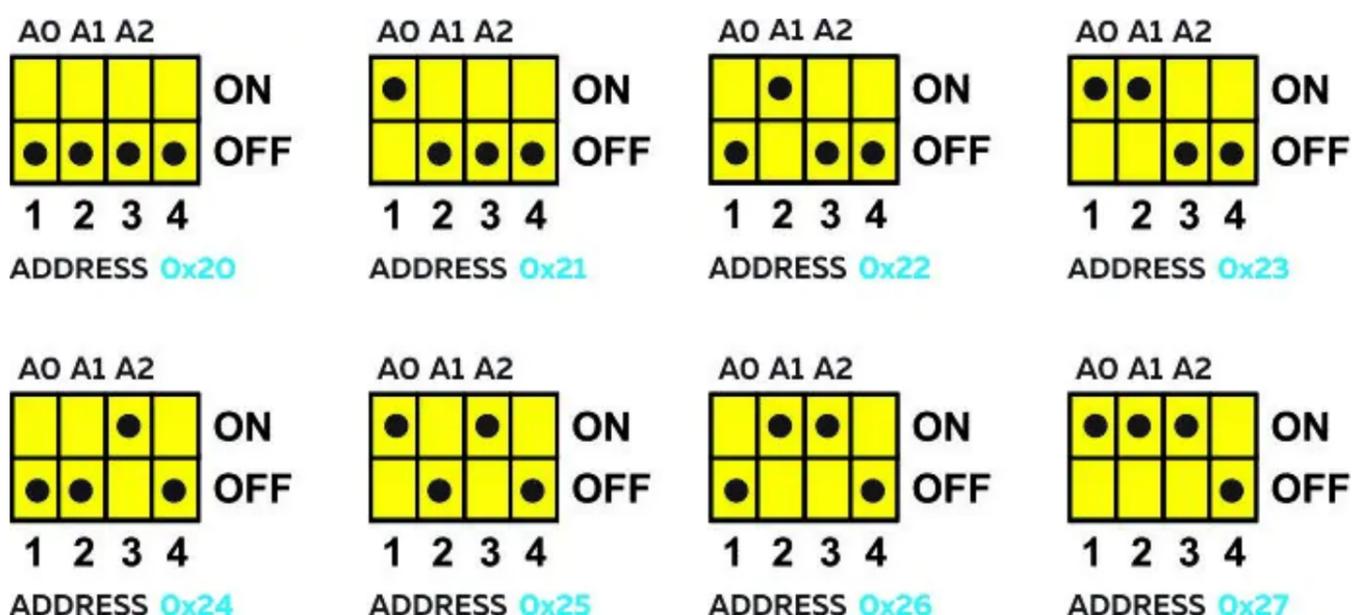
ENTRADAS y SALIDAS

Salidas Transistor

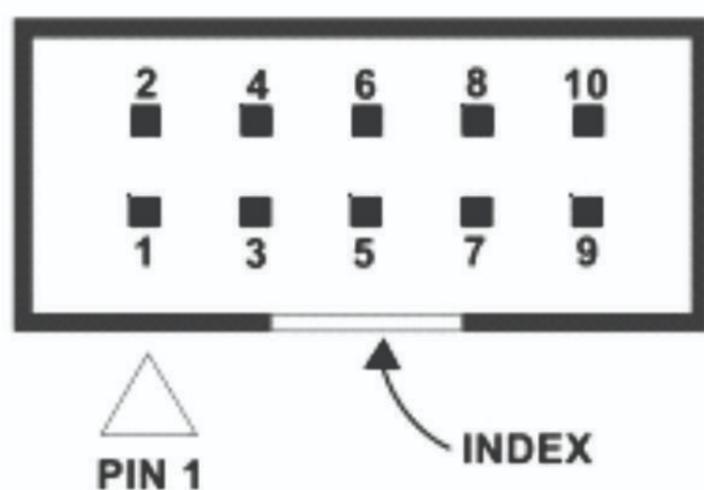
Numero de Salidas Transistor	16
Tipo de Salida Transistor	COLECTOR ABIERTO
Corriente Máxima Sink/Source (mA)	100mA
Tensión Máxima Aplicable	36V DC
Frecuencia Máxima de Conmutación	1 kHz
Disposición de Terminales	T1 – GPA0 T2 – GPA1 T3 – GPA2 T4 – GPA3 T5 – GPA4 T6 – GPA5 T7 – GPA6 T8 – GPA7 T9 – GPB0 T10 – GPB1 T11 – GPB2 T12 – GPB3 T13 – GPB4 T14 – GPB5 T15 – GPB6 T16 – GPB7

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.



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

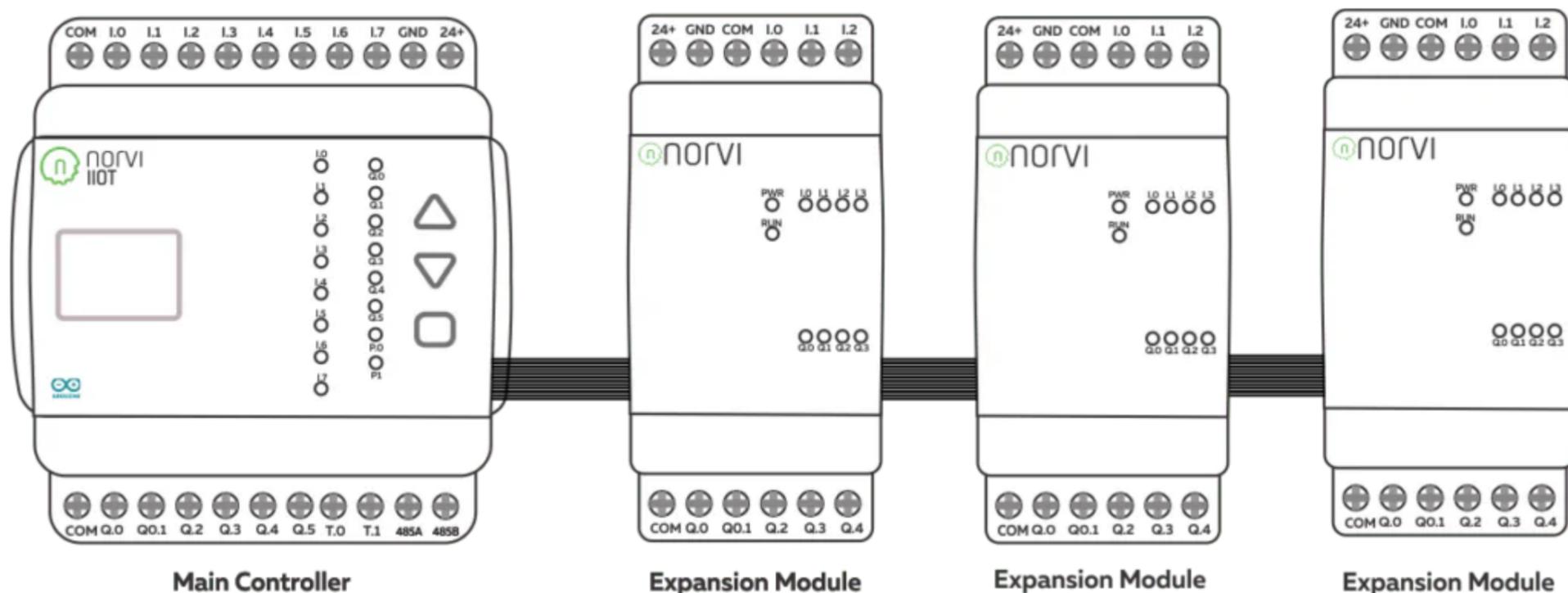
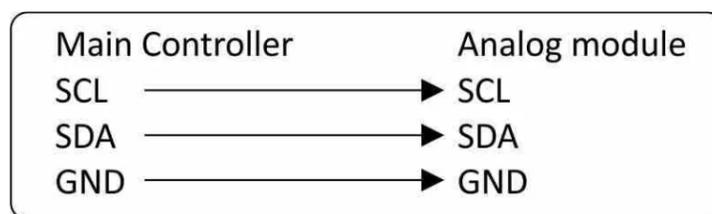
NORVI-EX-Q16 – GUIA DEL USUARIO

Programación

El NORVI-EX-Q16 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. Puede conectar en cadena hasta ocho módulos de expansión.

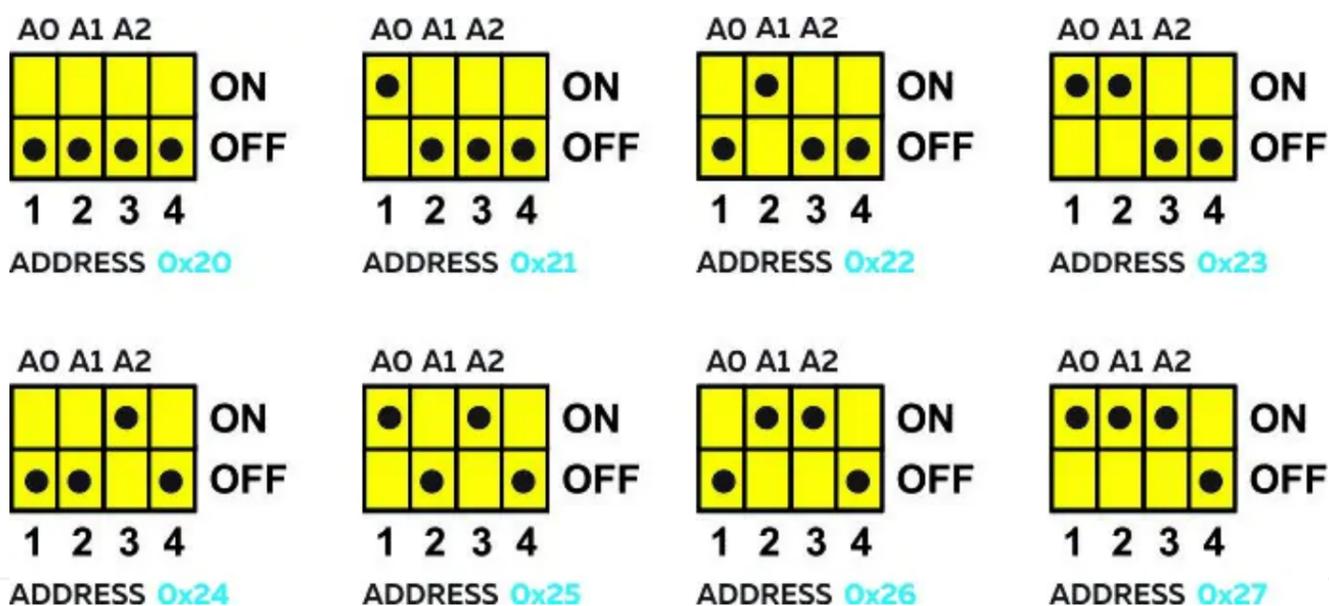
Puerto de Expansión

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.



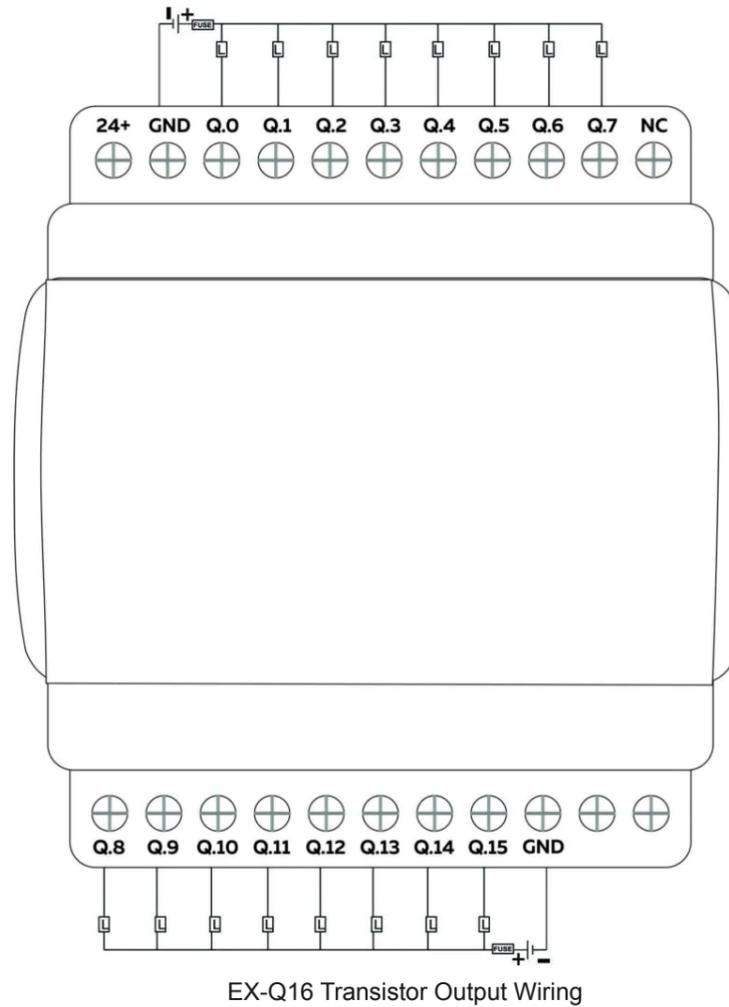
Configuración de la Dirección I2C

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 Transistor

Cableado de Salidas Transistor



Programación de Salidas Transistor

Escribir **HIGH** al **GPIO** relevante del MCP23008 activa la salida a transistor.

Consulte la Tabla de Asignación de GPIO para el GPIO de la Salida a Transistor.

Consulte la configuración de la dirección I2C para configurar la dirección I2C de la Expansión.

```
// Blinks an LED attached to a MCP23XXX pin.  
// ok to include only the one needed  
// both included here to make things simple for example  
//#include <Adafruit_MCP23X08.h>  
#include <Adafruit_MCP23X17.h>  
#define OUTPUT1 7  
#define OUTPUT2 6  
#define OUTPUT3 5  
#define OUTPUT4 4  
#define OUTPUT5 8  
#define OUTPUT6 9  
#define OUTPUT7 10  
#define OUTPUT8 11  
  
#define OUTPUT9 12  
#define OUTPUT10 13  
#define OUTPUT11 14  
#define OUTPUT12 15  
#define OUTPUT13 0  
#define OUTPUT14 1  
#define OUTPUT15 2  
#define OUTPUT16 3
```

```
// uncomment appropriate line  
//Adafruit_MCP23X08 mcp;  
Adafruit_MCP23X17 mcp;
```

```

void setup() {
  Serial.begin(115200);
  //while (!Serial);
  Serial.println("NORVO Expansions Test");

  // uncomment appropriate mcp.begin
  Wire.begin (16, 17);
  if (!mcp.begin_I2C(0x27)) {
    Serial.println("Error.");
    while (1);
  }

  // configure pin for output
  mcp.pinMode(OUTPUT1, 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...");
}
```