

Tarjeta de relevadores controlada por I2C 16CH-5V

La tarjeta de relevadores puede ser controlada por un puerto I2C de 5V o individualmente por líneas de control individuales. La tarjeta está formada por 16 relevadores SRD-05VDC-SL-C que incluyen entradas ópticamente aisladas. También se cuenta con una fuente de alimentación mediante un eliminador de 12V-1.5A y un módulo STEP-DOWN LM2596S-3A.

Especificaciones

- Modelo: RTRJ0016.
- Voltaje de alimentación: 12V/1.5A.
- Voltaje de operación lógico: 5VDC.
- Controlador: 2 Controladores PCF8574T por cada 8 canales. Control de direcciones mediante hardware.
- 16 Relevadores: SRD-05VDC-SL-C DE HASTA 100,000 ciclos de operación.
- Dimensiones: 212mm x 80mm.
- Corriente de operación máxima: 1.6A.

Incluye:

- Tarjeta de relevadores controlada por I2C de 16 canales a 5VDC.
- Eliminador 12V/1.5A jack 2.1mm.

Diseño esquemático

<https://gist.github.com/esmarr58/1b54747690a751ce5e8bdc8fa4cb2d64>

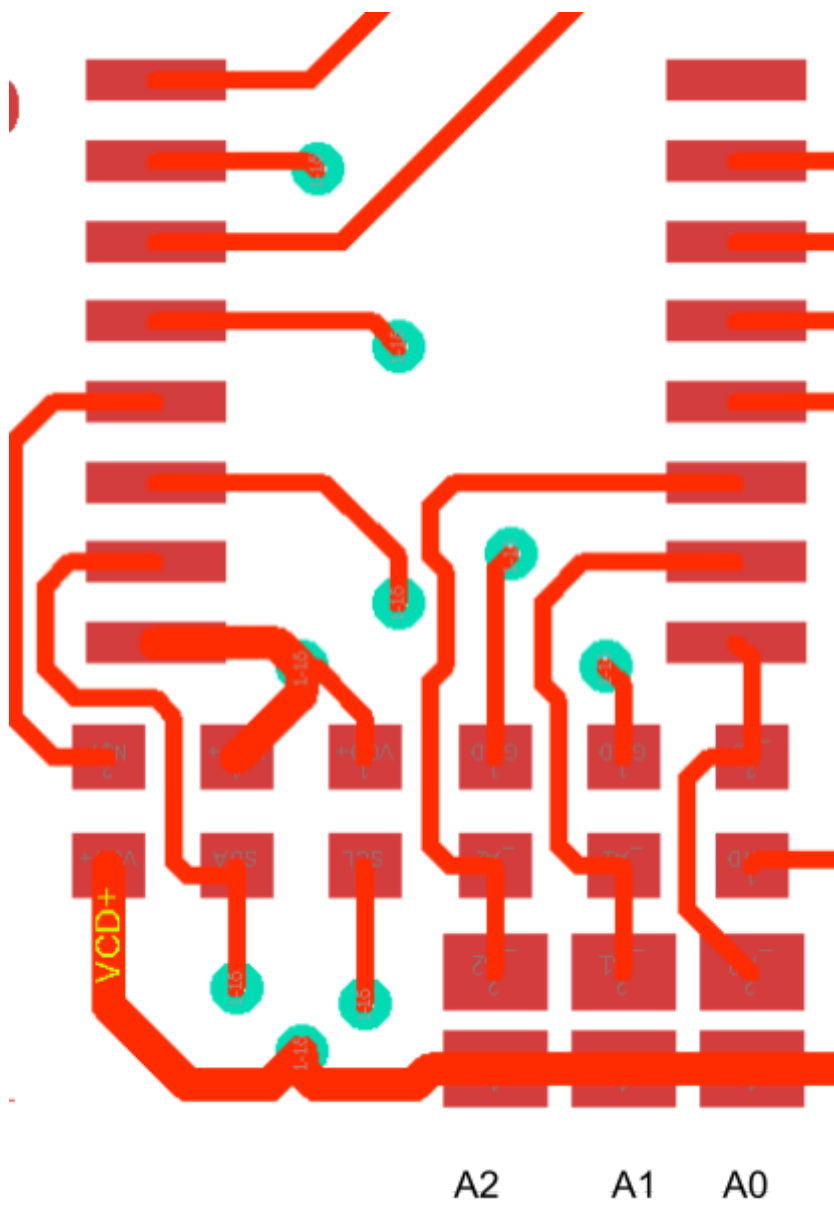


Figura 2. Jumpers selectores de direcciones. El rango de direcciones va desde 0x20 a 0x27

Código de Arduino para la tarjeta de relevadores controlada por I2C

```
/*HETPRO*/
/* Link de descarga de la libreria PCF8574:
 * https://github.com/skywodd/pcf8574\_arduino\_library
 * Programa de prueba para activar 16 relevadores de manera secuencial
 */

#include "Arduino.h"
#include "PCF8574.h"

// Crear dos objetos uno para cada expansor de E/S
PCF8574 pcf8(0x20); //Para activar los reles 1-8
PCF8574 pcf16(0x24); //Para activar los reles 9-16
int i;
int y;
void setup()
{
  // Declaramos ambos expansores como salidas
  for(i=0;i<8;i++){
    pcf8.pinMode(i, OUTPUT);
  }
  for(i=0;i<8;i++){
    pcf16.pinMode(i, OUTPUT);
  }
  //inicializamo cada uno de los Expansores de E/S
  pcf8.begin();
  pcf16.begin();
}

void loop()
{
  for(y=0; y<8; y++){
    pcf8.digitalWrite(y, LOW);
    delay(1000);
  }
  for(y=0; y<8; y++){
```

```
pcf16.digitalWrite(y, LOW);  
delay(1000);  
}
```

```
for(y=0; y<8; y++){  
pcf8.digitalWrite(y, HIGH);  
delay(1000);  
}
```

```
for(y=0; y<8; y++){  
pcf16.digitalWrite(y, HIGH);  
delay(1000);  
}
```

```
}
```