



KAmoD I2C Hub8 (PL)



Rev. 20250917053930

Źródło: [https://wiki.kamamilabs.com/index.php?title=KAmoD_I2C_Hub8_\(PL\)](https://wiki.kamamilabs.com/index.php?title=KAmoD_I2C_Hub8_(PL))

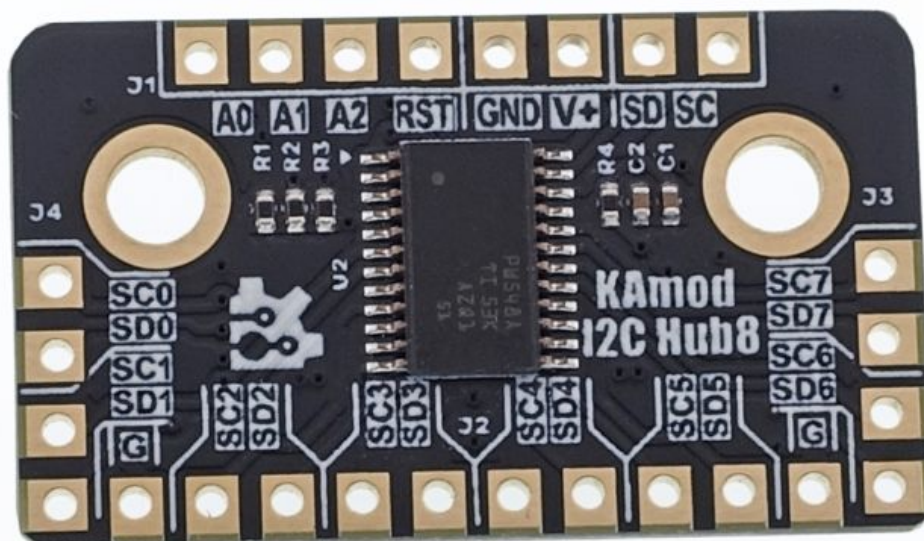
Spis treści

Opis	1
Podstawowe cechy i parametry	2
W wyposażenie standardowe	3
Schemat elektryczny	4
Rozmieszczenie linii I/O	5
Wyjściowe interfejsy I2C	6
Sterowanie wyjściami I2C	8
Zasilanie	9
Wymiary	10
Program testowy	11
Linki	14

Opis

KAmo*d* I2C Hub8 - Przełącznik/rozdzielacz interfejsu I2C z funkcją translacji napięć

Moduł KAmo*d* I2C Hub8 umożliwia przekierowanie sygnałów SCL i SDA wejściowego interfejsu I2C do ośmiu niezależnie aktywowanych wyjściowych interfejsów I2C (8 par SCL i SDA). Zastosowany w dowolnej aplikacji pozwala na usunięcie konfliktu zduplikowanych adresów układów na magistrali I2C i zapewnia dwukierunkową komunikację. Dodatkowo KAmo*d* I2C Hub8 może działać jako translator napięć magistrali I2C - interfejs wejściowy toleruje napięcie do 5,5 V i może być ono wyższe od napięcia zasilającego moduł.

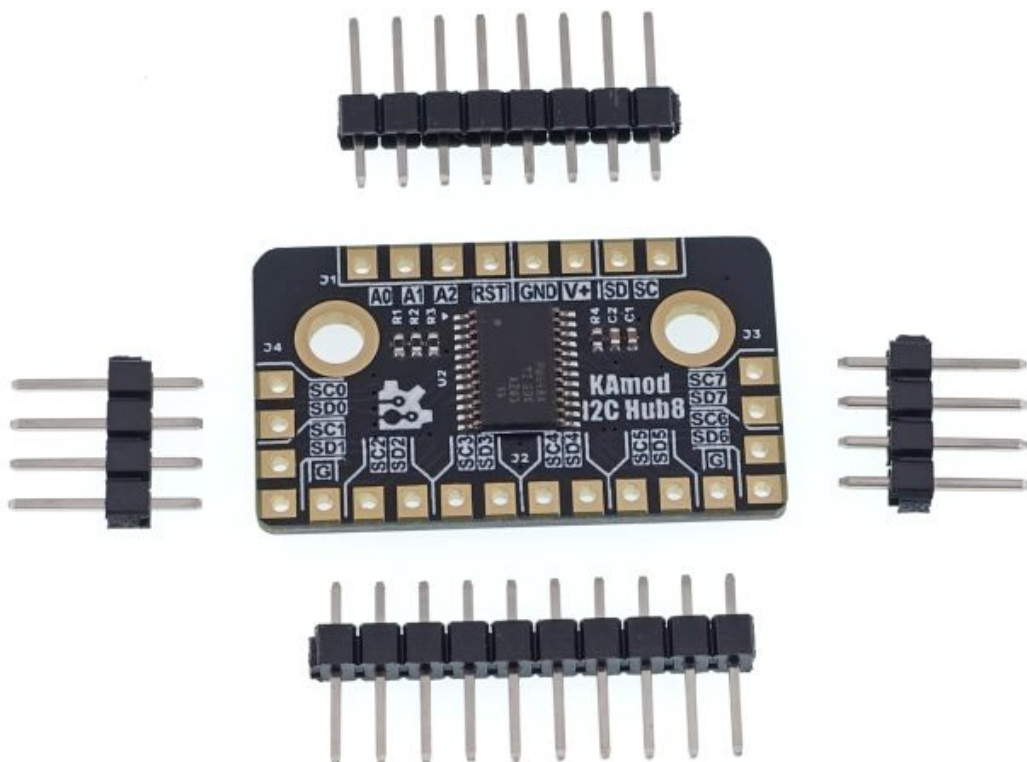


Podstawowe cechy i parametry

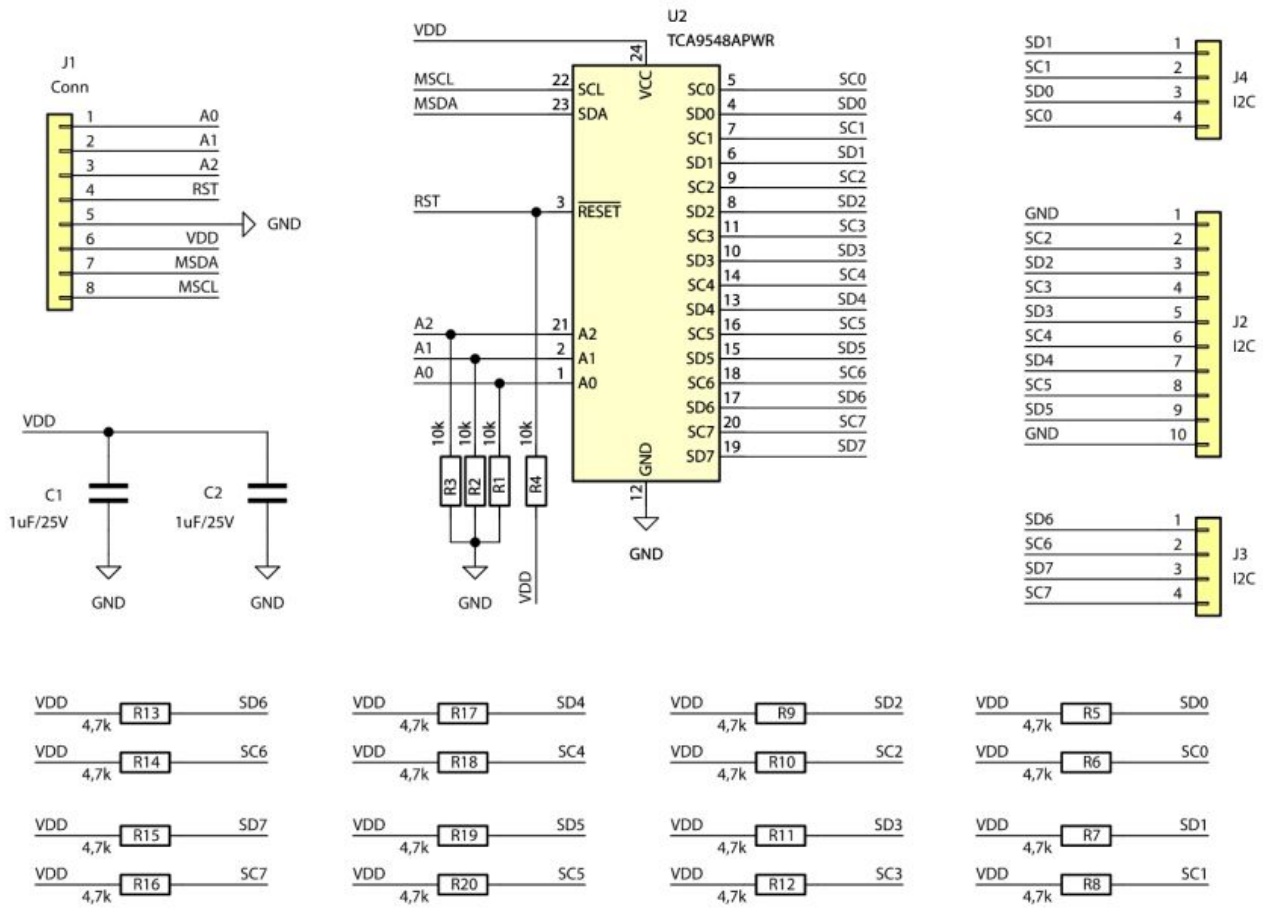
- Dwukierunkowy przełącznik/rozdzielacz interfejsu I2C z ośmioma wyjściami
- Dowolna kombinacja przekierowania wejściowego interfejsu I2C do wyjściowych
- Zawiera układ TCA9548A
- Trzy wejścia adresowe – możliwe dołączenie 8 modułów do jednego interfejsu
- Działa z taktowaniem interfejsu I2C do 400 kHz
- Działa z napięciem 1,8...5,5 V na liniach interfejsu I2C
- Wymaga zasilania napięciem 1,8...5,5 V
- Zawiera rezystory podciągające (pull-up, 4,7 kΩ) do zasilania V+ na wszystkich liniach interfejsów wyjściowych I2C (wszystkie dołączone układy muszą być kompatybilne z tym napięciem)
- Umożliwia translację napięć – interfejs wejściowy I2C toleruje napięcie wyższe (do 5,5 V) od napięcia zasilającego moduł

Wyposażenie standardowe

Kod	Opis
KAmod I2C Hub8	• Zmontowany i uruchomiony moduł • 1 x prosta listwa goldpin 10-pin raster 2,54 mm • 1 x prosta listwa goldpin 8-pin raster 2,54 mm • 2 x prosta listwa goldpin 4-pin raster 2,54 mm



Schemat elektryczny



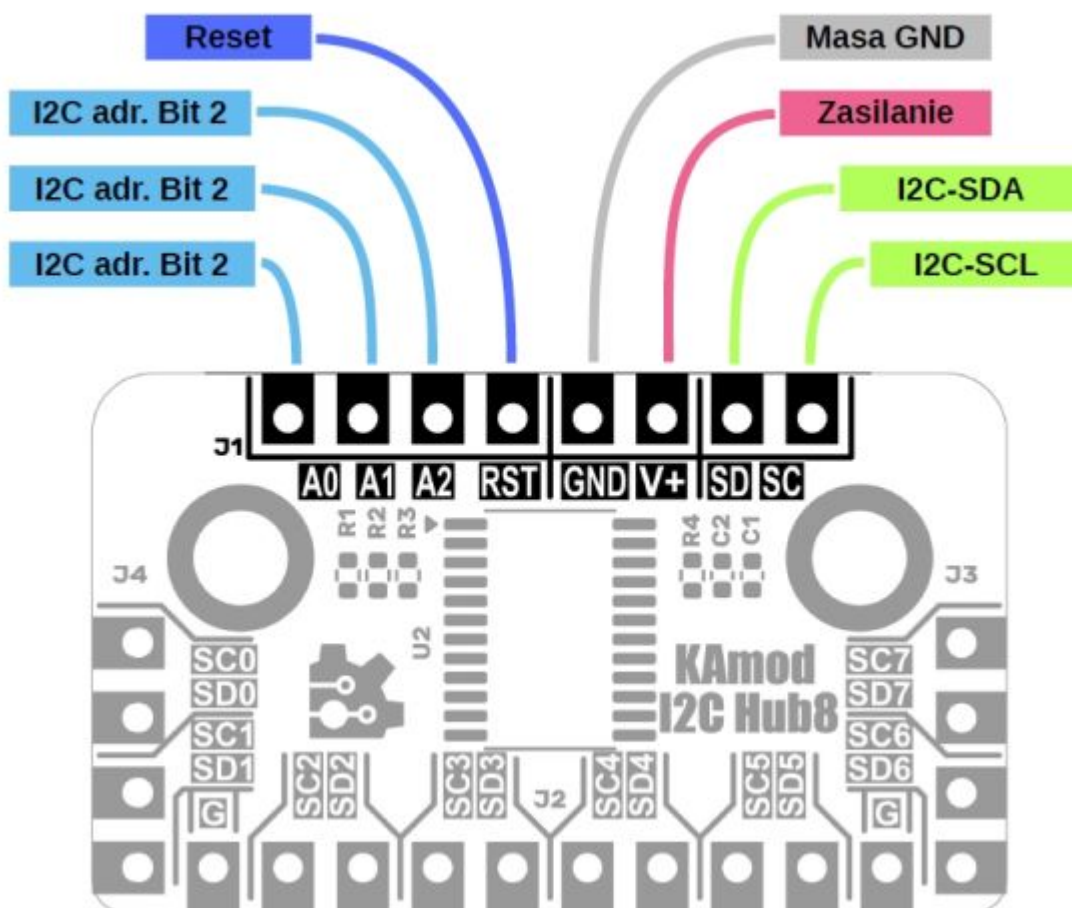
Rozmieszczenie linii I/O

Oznaczenie (J1)	Funkcja
SD	• Sygnał SDA (dane) interfejsu I2C
SC	• Sygnał SCL (zegar) interfejsu I2C
RST	• Wejście zerujące układ, aktywne w stanie L
A0, A1, A2	• Wejścia umożliwiające ustawienie adresu interfejsu I2C (bity 2, 1, 0). Domyślnie ustawione na logiczne 0.

Sterowanie modulem KAmoD I2C Hub8 odbywa się poprzez interfejs I2C wyprowadzony na złączu J1. Linie SD i SC wymagają zewnętrznych rezystorów podciągających pull-up. Komunikacja może być taktowana sygnałem zegarowym o częstotliwości do 400 kHz.

Wejścia A0, A1, A2 pozwalają ustawić adres (bity nr 2, 1 i 0), jakim będzie dysponował moduł dołączony do interfejsu I2C. Domyślnie wejścia te są połączone do masy poprzez rezystory (pull-down) i przyjmują wartości logiczne 0. Dołączenie dodatniego bieguna zasilania do wejść A0/A1/A2 ustawia logiczne 1 dla najmłodszych bitów adresu. Wszystkie dostępne adresy interfejsu I2C zawierają się w zakresie 0x70...0x77.

Sygnał zerujący (RST) służy do wyzerowania układu TCA9548A na płytce KAmoD I2C Hub8. Podanie logicznego stanu niskiego (L) przywraca wszystkie ustawienia do stanu początkowego – wyjścia I2C będą nieaktywne.

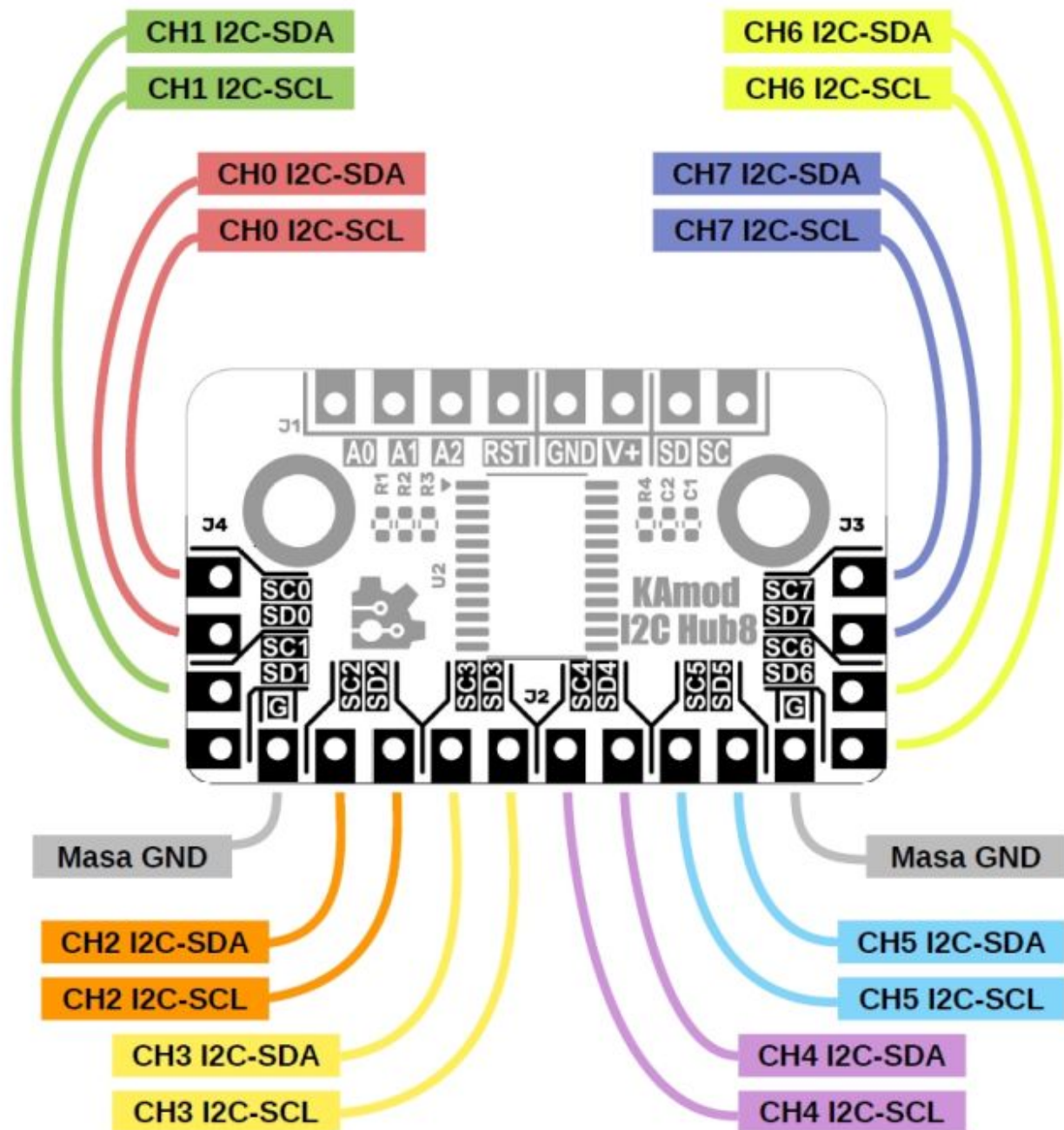


Wyjściowe interfejsy I2C

Oznaczenie	Funkcja
SD0, SC0 (J4)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 0
SD1, SC1 (J4)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 1
SD2, SC2 (J2)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 2
SD3, SC3 (J2)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 3
SD4, SC4 (J2)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 4
SD5, SC5 (J2)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 5
SD6, SC6 (J3)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 6
SD7, SC7 (J3)	Sygnały SDA (dane) i SCL (zegar) interfejsu I2C 7

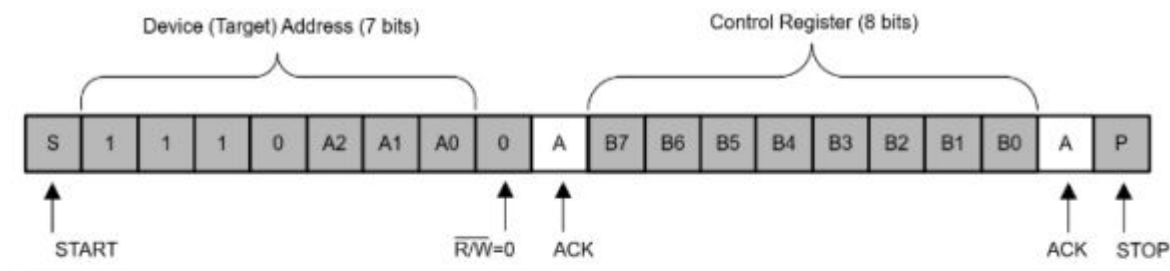
Moduł KAmoD I2C Hub8 zawiera 8 wyjściowych interfejsów I2C oznaczonych jako I2C0...I2C7. Komunikacja pomiędzy Interfejsem wejściowym, a interfejsami wyjściowymi jest w pełni dwukierunkowa - nazwy „wejściowy” i „wyjściowy” są tylko umowne.

Interfejsy I2C0...I2C7 są wyposażone w rezystory podciągające pull-up, o wartości 4,7 kΩ dołączone do zasilania V+. Wszystkie układy dołączone do interfejsów wyjściowych muszą być kompatybilne z tym napięciem. Rozmieszczenie interfejsów oraz poszczególnych linii SDA i SCL zostało dokładnie oznaczone na płytce PCB modułu oraz na rysunku:

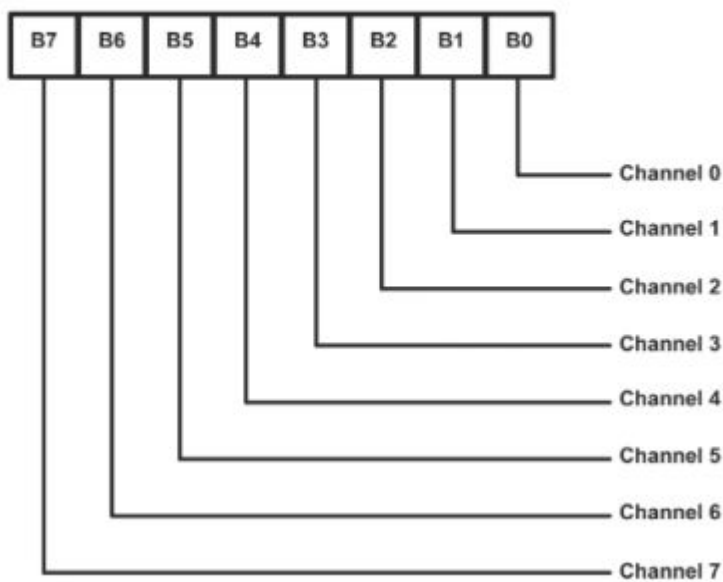


Sterowanie wyjściami I2C

Sterowanie modulem KAmoD I2C Hub8 jest możliwe poprzez interfejs I2C na złączu J1 (linie SD i SC). Zastosowany w module układ TCA9548A wymaga wysłania 1 bajtu danych, który ustawia wyjścia I2C0...I2C7 w stan aktywny lub nieaktywny. Podstawowy adres układu to 0x70 i może być zmieniany w zakresie 0x70...0x77 w zależności od stanu wejść A2, A1, A0 na złączu J1. Format ramki sterującej I2C pokazują rysunki.



INPUTS			I ² C BUS TARGETADDRESS
A2	A1	A0	
L	L	L	112 (decimal), 70 (hexadecimal)
L	L	H	113 (decimal), 71 (hexadecimal)
L	H	L	114 (decimal), 72 (hexadecimal)
L	H	H	115 (decimal), 73 (hexadecimal)
H	L	L	116 (decimal), 74 (hexadecimal)
H	L	H	117 (decimal), 75 (hexadecimal)
H	H	L	118 (decimal), 76 (hexadecimal)
H	H	H	119 (decimal), 77 (hexadecimal)

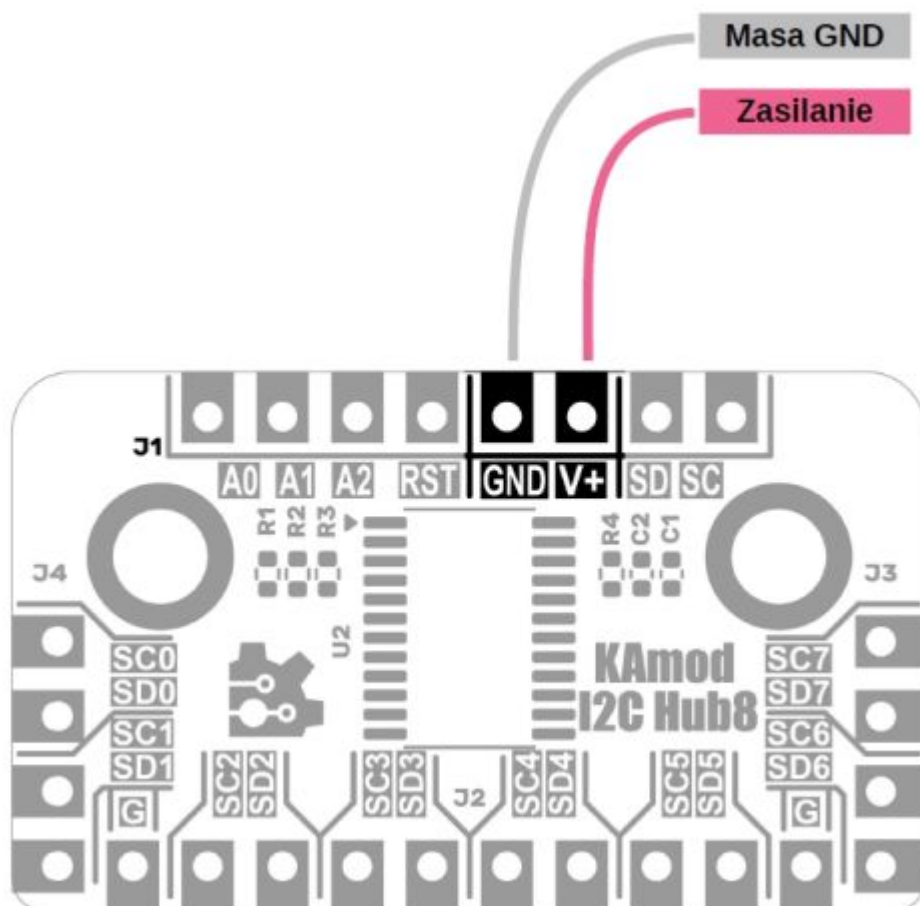


Szczegółowe informacje dotyczące komunikacji i konfiguracji układu TCA9548A są dostępne w dokumentacji technicznej tego układu.

Zasilanie

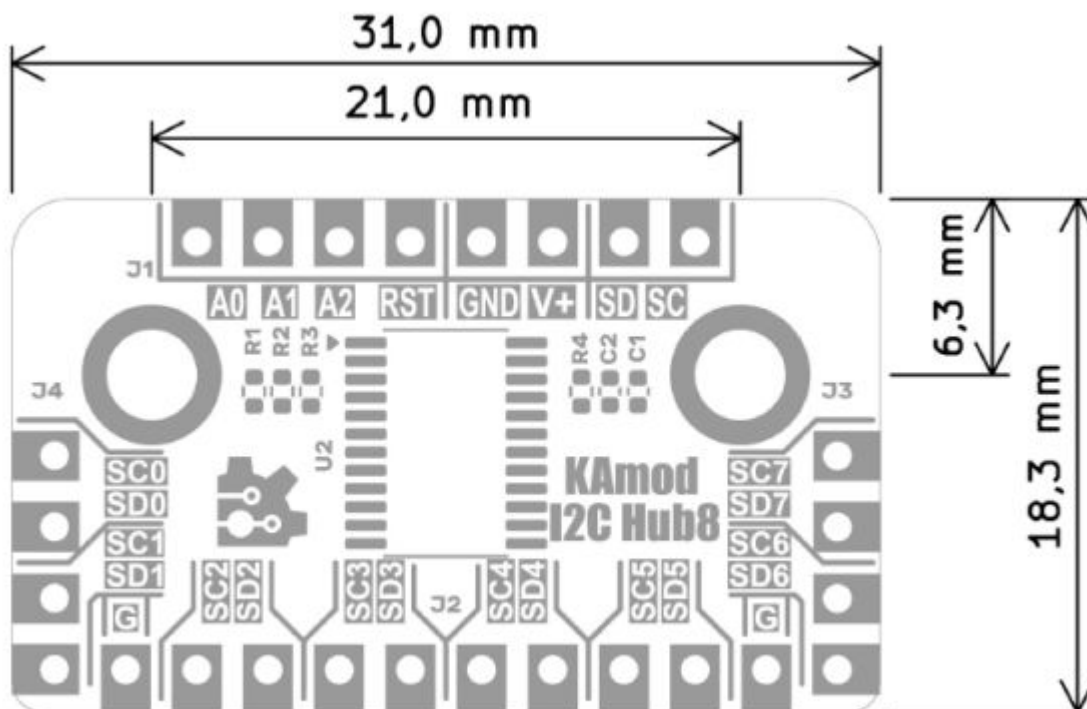
Oznaczenie (J1)	Funkcja
V+	wejście zasilania, biegun dodatni
GND	wejście zasilania, biegun ujemny (GND)

Moduł KAmođ I2C Hub8 może być zasilany napięciem stałym o wartości 1.8...5,5V, dołączonym do styków V+ oraz GND na złączu J1.



Wymiary

Wymiary płytki KAmo*d* I2C Hub8 to 31x18,3 mm. Na płytce znajdują się 2 otwory montażowe o średnicy 3,2 mm.



Program testowy

Program testowy został napisany w środowisku Arduino dla płytki KAmoESP32 POW RS485.

```
//example code for KAmo I2C Hub8
//control 8x MCP23008 with the same I2C address
//ino board: ESP32-WR00M-DA Module

#include <Wire.h>
//#include <Adafruit_MCP23X17.h>
#include <Adafruit_MCP23X08.h>

#define DEBUG False

//LED
#define LED_PIN      2
#define MY_DELAY     2000
#define TEST_LED_ON  1
#define TEST_LED_OFF 0

//I2C
#define I2C_SDA      33
#define I2C_SCL      32
#define I2C_FREQ     400000
#define MCP23008_ARD 0x20

#define TCA9548_ARD   0x70
#define TCA9548_CH0   (1<<0)
#define TCA9548_CH1   (1<<1)
#define TCA9548_CH2   (1<<2)
#define TCA9548_CH3   (1<<3)
#define TCA9548_CH4   (1<<4)
#define TCA9548_CH5   (1<<5)
#define TCA9548_CH6   (1<<6)
#define TCA9548_CH7   (1<<7)

TwoWire I2Cbus = TwoWire(0);
Adafruit_MCP23X08 MCP10;
int i;
int sel;
int success;

//-----
bool i2cHub(uint8_t ch_mask){
    I2Cbus.beginTransmission(TCA9548_ARD);
    I2Cbus.write(&ch_mask, 1);
    return I2Cbus.endTransmission(true);
}

//-----
bool i2cCheck(uint8_t adr){
    I2Cbus.beginTransmission(adr);
    return I2Cbus.endTransmission(true);
}

//-----
void setup() {
```

```

Serial.begin(115200);
while (!Serial) {}
delay(1000); // wait for native usb
Serial.println(); Serial.println(); Serial.println();
Serial.println("Hello. KAmode I2C Hub8 test");

pinMode(LED_PIN, OUTPUT);
digitalWrite(LED_PIN, HIGH);

delay(500);
while(!I2Cbus.begin(I2C_SDA, I2C_SCL, I2C_FREQ)){
  Serial.println("I2C Bus init failed...");
}

success = 0;
sel = 7;

while(1){
  sel++;
  if (sel > 7) {
    sel = 0;
    success = 0;
    delay(500);
  }
  i2cHub(1<<sel);
  delay(20);

  if (!i2cCheck(MCP23008_ARD)){
    delay(20);
    if (MCPI0.begin_I2C(MCP23008_ARD, &I2Cbus)) {
      Serial.print("MCP init OK at channel ");
      success++;
    } else {
      Serial.print("MCP init FAIL at channel ");
    }
  } else {
    Serial.print("MCP Detect Failed at channel ");
  }
  Serial.print(sel);
  Serial.println();

  if ((sel == 7) && (success > 0)) break;
}

Serial.println("Setup finished");
sel = 7;
i = 8;
success = 0;
}

//-----
void loop() {
  i++;
  if (i >= 8){
    i = 0;
    Serial.println(); Serial.println();
    while(1){
      sel++;
      if (sel > 7) {

```

```

    sel = 0;
    delay(500);
}

i2cHub(1<<sel);
if (!i2cCheck(MCP23008_ARD)){
    Serial.println(); Serial.println("#####");
    Serial.print("Select I2C channel ");
    Serial.print(sel); Serial.println();
    Serial.print("Out: ");
    success++;
    break;
} else {
    Serial.print("MCP Detect Failed at channel ");
    Serial.print(sel); Serial.println();
    success = 0;
}
delay(50);
}
}

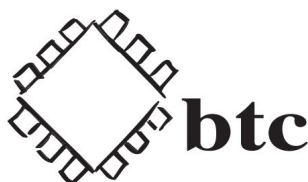
if (i & 1){
    digitalWrite(LED_PIN, TEST_LED_ON);
} else {
    digitalWrite(LED_PIN, TEST_LED_OFF);
}
Serial.print(i); Serial.print(" ");
MCPIO.pinMode(i, OUTPUT);
MCPIO.digitalWrite(i, TEST_LED_ON);
delay(200);
MCPIO.pinMode(i, OUTPUT);
MCPIO.digitalWrite(i, TEST_LED_OFF);
delay(100);

if (success >= 16) {
    Serial.println(); Serial.println(); Serial.println();
    Serial.println("### Verify OK 2 times ###");
    Serial.println("### THATS ALL ###");
    while(1){};
}
}
}

```

Linki

- [Karta katalogowa TCA9548A](#)
- [Przykładowe programy](#)
- [Model CAD \(STEP\)](#)



BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail:
sprzedaz@kamami.pl
<https://kamami.pl>

Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.

Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.

BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.

BTC Korporacja nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.

BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.