



KAmo SPI Flash (PL)



Rev. 20250728054228

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

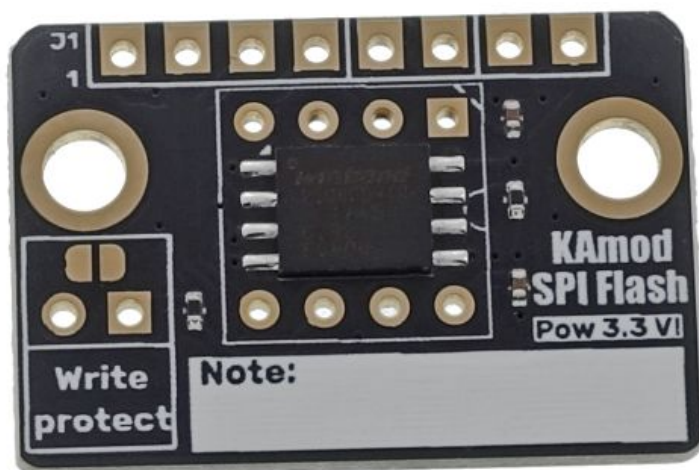
Spis treści

Opis	1
Podstawowe cechy i parametry	2
W wyposażenie standardowe	3
Schemat elektryczny	4
Sygnały sterujące	5
Oznaczenie pojemności pamięci	7
Zabezpieczenie przed zapisem „Write protect”	9
Zasilanie	10
Wymiary	11
Program testowy	12
Linki	15

Opis

KAmođ SPI Flash - Moduł z nieulotną pamięcią typu Flash z interfejsem SPI

Moduł KAmođ SPI Flash zawiera pamięć Flash z interfejsem SPI. Wszystkie sygnały sterujące zostały wyprowadzone jako punkty lutownicze z otworami, rozmieszczone ze standardowym rastrem 2,54 mm, tuż przy krawędzi płytki oraz w sposób odpowiadający obudowie do montażu przewlekanego DIL8. Moduł dostępny jest z pamięciami o pojemności 1 MB (8Mb) lub 16 MB (128 Mb).

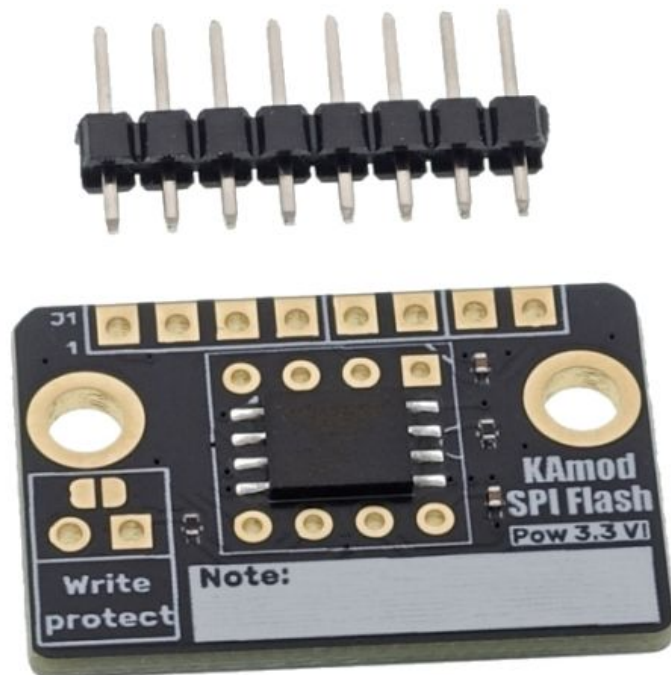


Podstawowe cechy i parametry

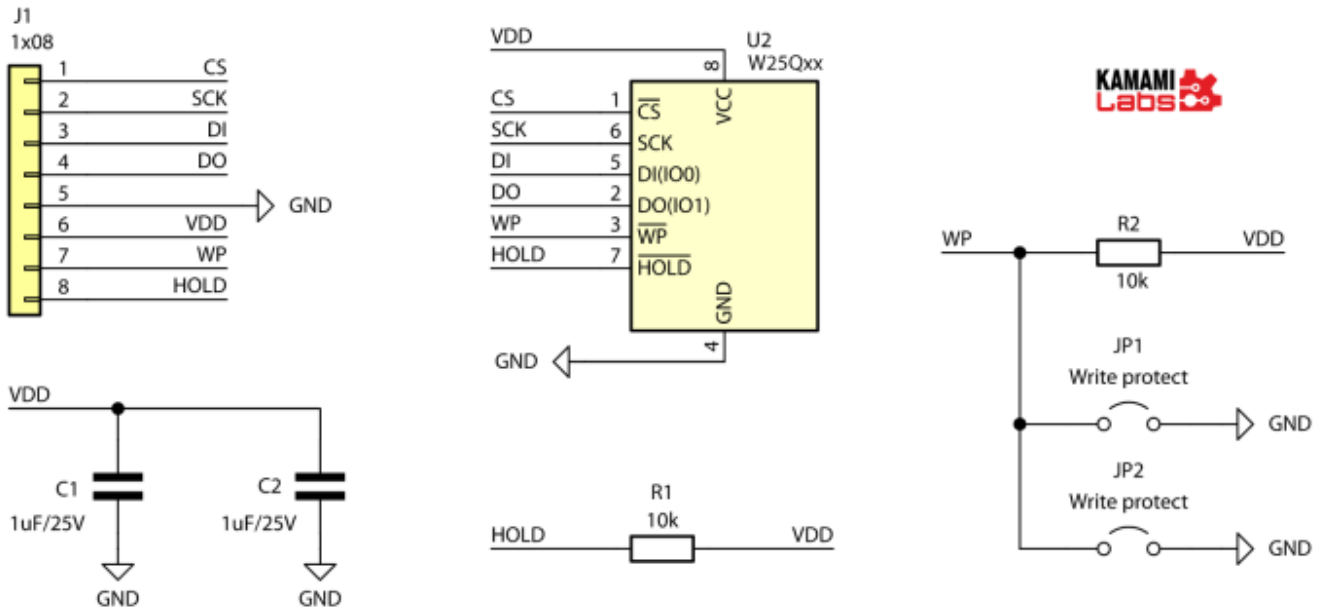
- Pamięć nieulotna Flash o pojemności 1 MB (W25Q80) lub 16 MB (W25Q128)
- Interfejs SPI o taktowaniu do 50 MHz (praca przy wyższej częstotliwości jest możliwa, ale zależy od jakości/długości połączeń sygnałów interfejsu SPI)
- Zawartość pamięci można zabezpieczyć przed skasowaniem montując zworkę „Write protect”
- Sygnały sterujące wyprowadzone jako punkty lutownicze z otworami, rozmieszczone ze standardowym rastrem 2,54 mm, tuż przy krawędzi płytki
- Sygnały sterujące dodatkowo zostały wyprowadzone jako punkty lutownicze z otworami, rozmieszczone ze standardowym rastrem 2,54 mm w sposób odpowiadający obudowie do montażu przewlekane DIL8
- Zasilanie napięciem 2,7...3,6 V
- Wymiary płytki: 27 x 18,3 mm

Wposażenie standardowe

Kod	Opis
KAmo SPI Flash	- Zmontowany i uruchomiony moduł 1 x prosta listwa goldpin 8-pin raster 2,54 mm



Schemat elektryczny



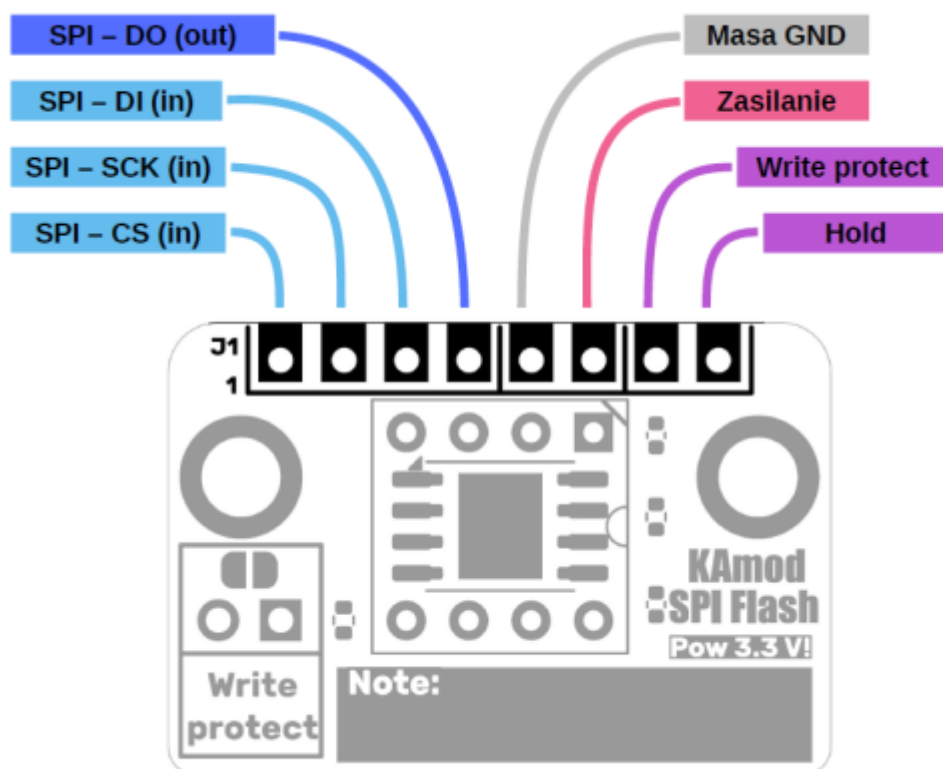
Sygnały sterujące

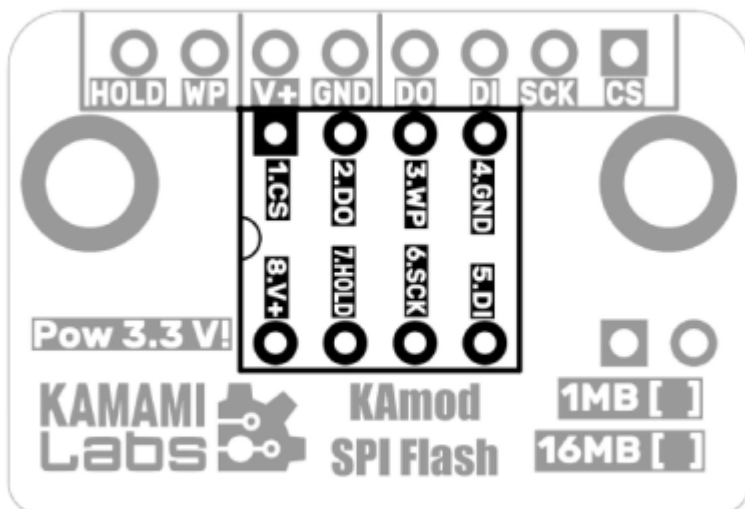
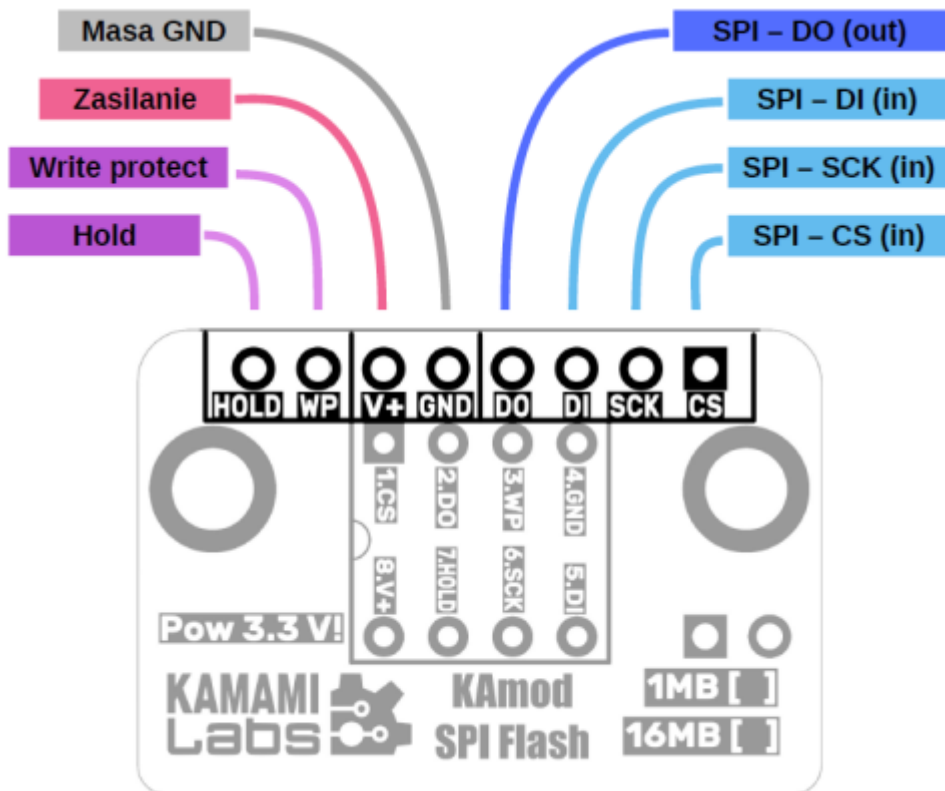
Oznaczenie (J1)	Funkcja
CS	• Wejście sygnału CS (Chip Select - aktywowanie interfejsu SPI)
SCK	• Wejście sygnału zegarowego/taktującego
DI	• Wejście szeregowe danych (Data In)
DO	• Wyjście szeregowe danych (Data Out)
GND	• Zasilanie - biegun ujemny (masa/GND)
V+	• Zasilanie - biegun dodatni, napięcie 2,7...3,6 V
WP	• Wejście blokady zapisu (Write Protect) aktywne w stanie niskim, połączone ze zworką Write protect
HOLD	• Wejście sygnału zatrzymania komunikacji

Komunikacja z pamięciami W25Q80/W25Q128 stosowanymi w KAmoD SPI Flash jest możliwa poprzez interfejs SPI. Niektóre sygnały pełnią podwójną rolę w trybie komunikacji Dual/Quad SPI. Dokładny opis wszystkich sygnałów, sposobów komunikacji oraz procedur i komend jest zawarty w dokumentacji technicznej układów.

Interfejs SPI układów pamięci typu W25Qxxx może działać z bardzo wysokim taktowaniem (sygnał na wejściu SCK) - nawet ponad 100 MHz, jednak w przypadku modułu połączonego przewodami nie zaleca się komunikacji z taktowaniem większym niż 50 MHz. W razie problemów z komunikacją należy zmniejszyć częstotliwość taktowania SPI.

Rozmieszczenie sygnałów sterujących zostało pokazane na poniższych rysunkach. Oznaczenia sygnałów znajdują się na dolnej stronie płytki (bottom), zarówno dla złącza na krawędzi płytki jak i dla obudowy DIL8.

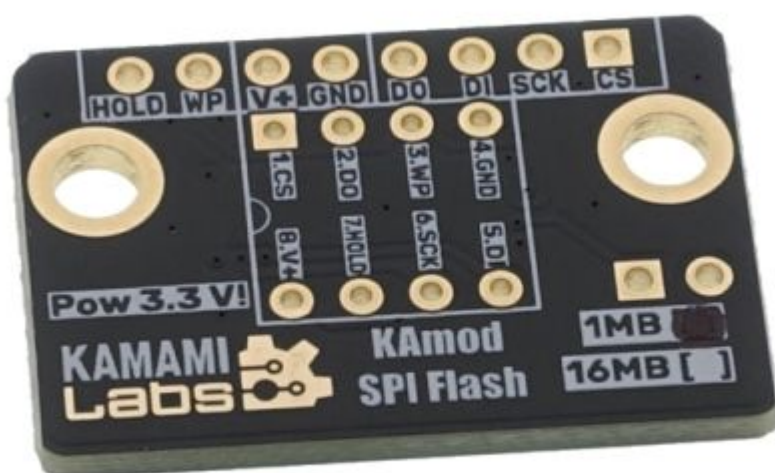
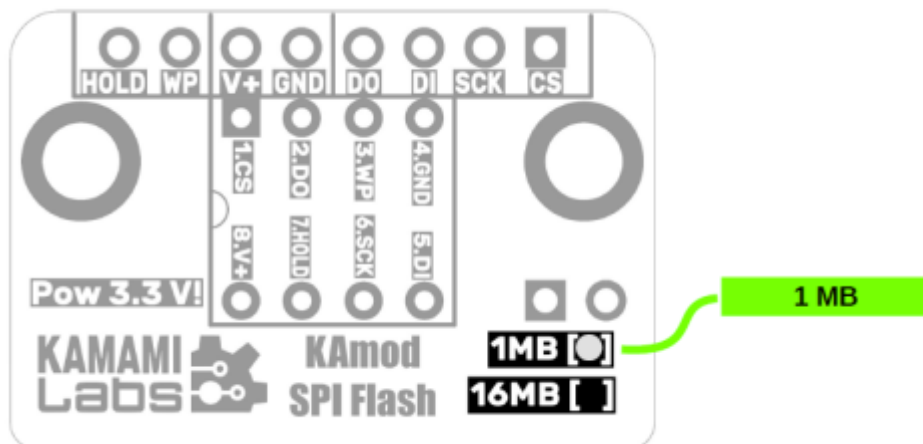


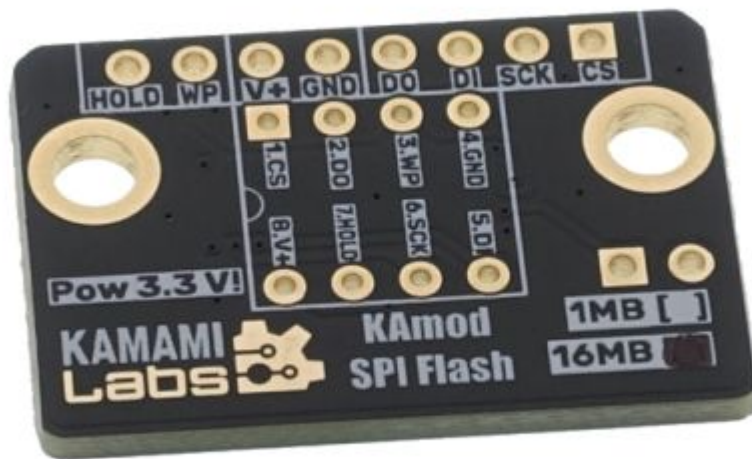
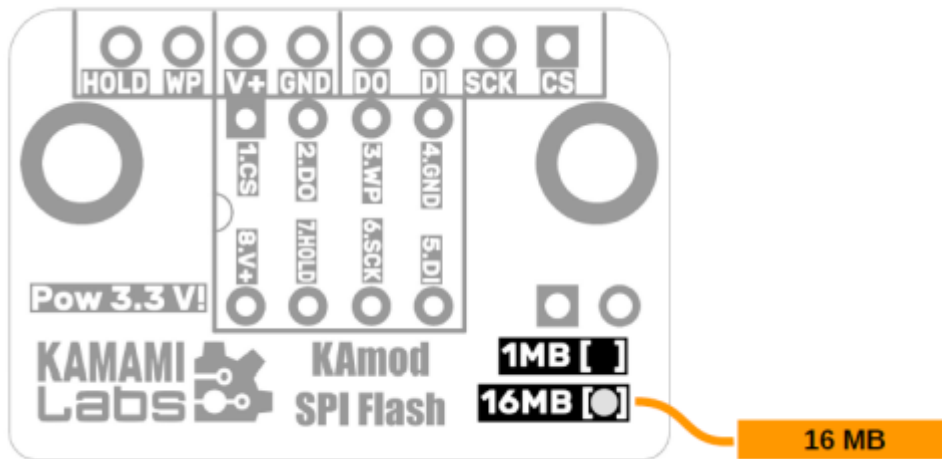


Oznaczenie pojemności pamięci

Moduł KAmođ SPI Flash dostępny jest z pamięciami o pojemności 1 MB (8Mb) lub 16 MB (128 Mb). W zależności od wersji na płytce zamontowane są układy W25Q80 (1 MB) lub W25Q128 (16 MB).

Wersja pamięci jest zaznaczona na dolnej stronie płytki (bottom) tak, jak pokazano poniżej.



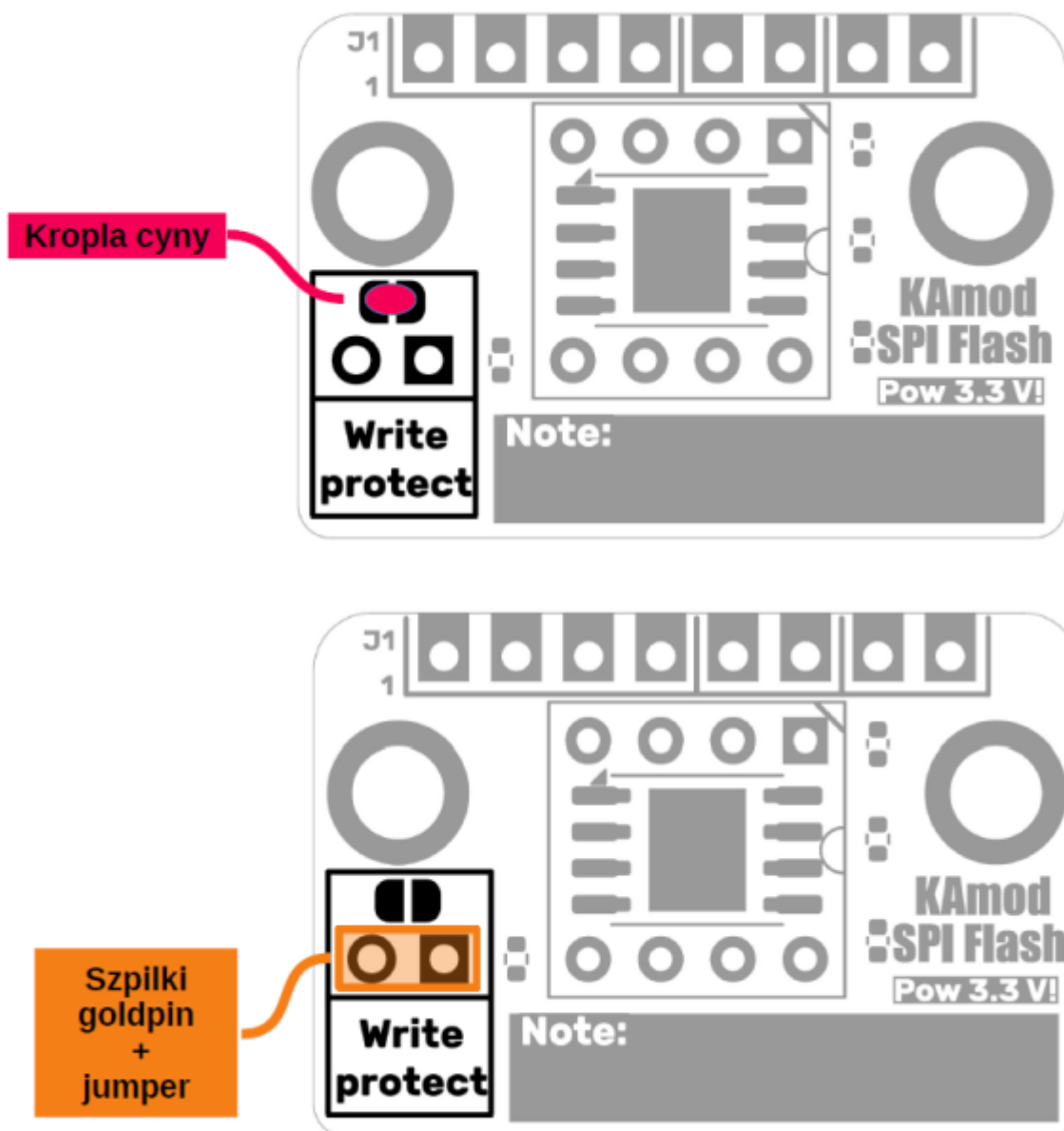


Zabezpieczenie przed zapisem „Write protect”

Oznaczenie (J1)	Funkcja
WP (J1)	• Podanie stanu niskiego uaktywnia funkcję zabezpieczenia przed zapisem
Write protect	• Zamontowanie zworki/szpilek goldpin uaktywnia funkcję zabezpieczenia przed zapisem

Funkcja Write protect zabezpiecza pamięć przez skasowaniem lub zmianą jej zawartości.

Aby uaktywnić funkcję zabezpieczającą należy podać stan niski (logiczne 0) na wyprowadzenie WP. Innym sposobem jest przylutowanie kropki cyny w miejscu jumpers SMD lub przylutowanie szpilek goldpin z zamontowanym jumperem, tak jak pokazano na rysunku. Zamontowanie jumpers również wymusza stan niski na wyprowadzeniu WP.

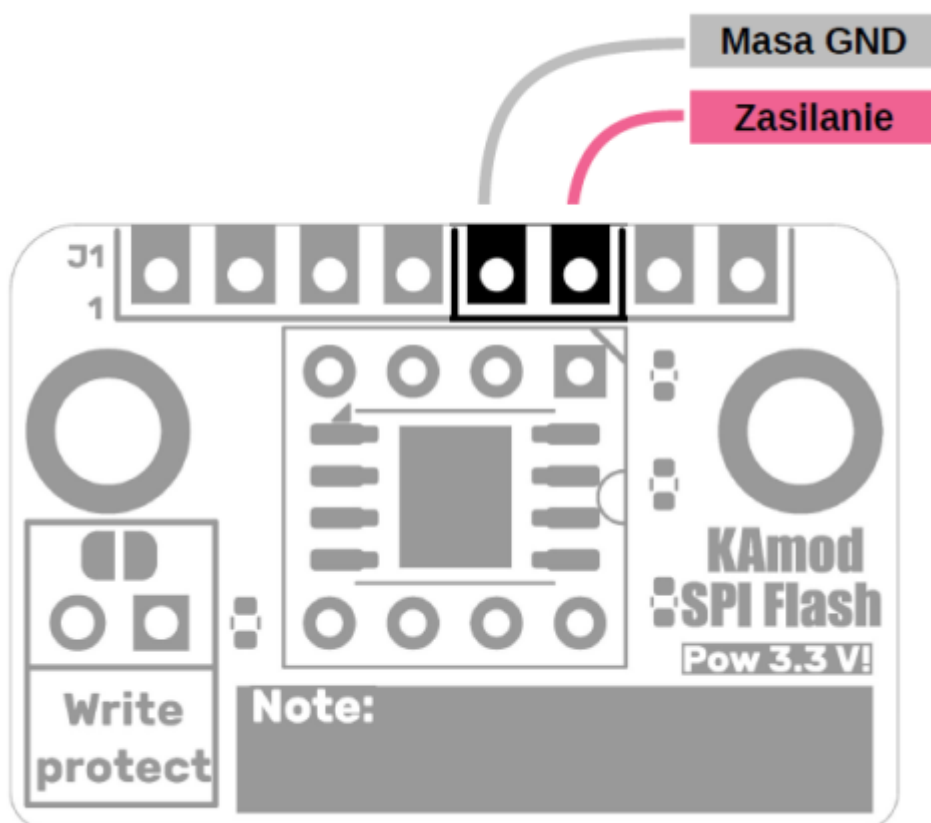


Zasilanie

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

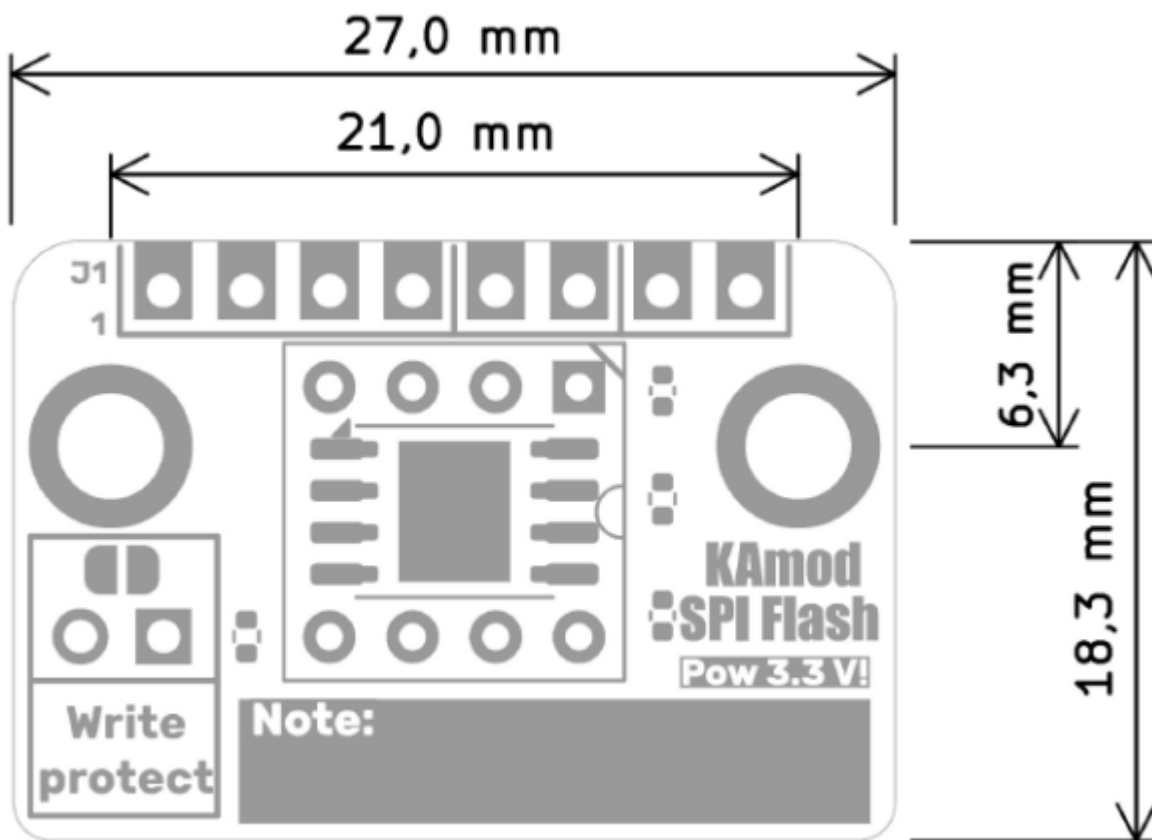
Moduł KAmo SPI Flash może być zasilany napięciem stałym o wartości 2,7...3,6V, dołączonym do styków V+ oraz GND na złączu J1.

Nie należy dołączać zasilania o napięciu 5V!



Wymiary

Wymiary płytki KAmo SPI Flash to 27x18,3 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 SPI Flash 1MB (W25Q80)
//ino board: ESP32-WR00M-DA Module

#include <SPI.h>
#include "Kamod_SPIFlashBase.h"

//LED
#define LED_PIN      2
#define MY_DELAY     2000

//SPI
#define SPI_MOSI      13
#define SPI_MISO      12
#define SPI_SCK       14
#define SPI_CS        15
#define SPI_CLOCK     100000000

//INIT
SPIClass spiFlashBus(HSPI);
Kamod_SPIFlashBase spiFlashBase(&spiFlashBus, SPI_CS, SPI_SCK, SPI_MOSI, SPI_MISO);
Kamod_SPIFlash_Device_t MY_W25Q80DV;
uint8_t *data_buff;

int ledi;
int loops;
bool retstat;
bool verify;
int result;

//-----
void setup() {
  Serial.begin(115200);
  while (!Serial) {}
  delay(1000); // wait for native usb
  Serial.println();
  Serial.println();
  Serial.println();
  Serial.println("Hello. KAmo SPI Flash test start");

  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, HIGH);
  ledi = 0;

  retstat = false;
  while(retstat == false){
    spiFlashBus.begin(SPI_SCK, SPI_MISO, SPI_MOSI, SPI_CS);
    spiFlashBase.setClockSpeed(SPI_CLOCK, SPI_CLOCK);
    retstat = spiFlashBase.begin(&MY_W25Q80DV);
    if (!retstat){
      Serial.println("Init fail ");
      delay(1000);
    } else {
      Serial.print("JEDEC: ");
    }
  }
}
```

```

        Serial.println(spiFlashBase.getJEDECID(), HEX);
    }
}

Serial.println("Flash init...OK");
Serial.println("--- First, erase sector ---");
spiFlashBase.eraseSector(0);
delay(500);
loops = 0;
result = 0;

Serial.println();
}

//-----
void loop() {
    Serial.print("### START at page: ");
    Serial.print(loops);
    Serial.println(" ###");

    Serial.println("--- Clear buff ---");
    data_buff = spiFlashBase.buff();
    for(int n = 0; n<KAMOD_SFLASH80_PAGE_SIZE; n++){
        *data_buff = 0;
        data_buff++;
    }

    Serial.println("--- Read mem ---");
    spiFlashBase.readPageBuff(loops * KAMOD_SFLASH80_PAGE_SIZE);
    data_buff = spiFlashBase.buff();
    for(int n = 0; n<20; n++){
        Serial.print(*data_buff);
        Serial.print(",");
        data_buff++;
    }
    Serial.println("");

    Serial.println("--- Prepare new buff ---");
    data_buff = spiFlashBase.buff();
    for(int n = 0; n<20; n++){
        *data_buff = (uint8_t)(n + (loops*10));
        Serial.print(*data_buff);
        Serial.print(",");
        data_buff++;
    }
    Serial.println("");

    Serial.println("--- Write mem ---");
    //spiFlashBase.eraseSector(0);
    spiFlashBase.writePageBuff(loops * KAMOD_SFLASH80_PAGE_SIZE);
    Serial.println("--- Clear buff ---");
    data_buff = spiFlashBase.buff();
    for(int n = 0; n<KAMOD_SFLASH80_PAGE_SIZE; n++){
        *data_buff = 0;
        data_buff++;
    }

    Serial.println("--- Read mem for check content ---");
    spiFlashBase.readPageBuff(loops * KAMOD_SFLASH80_PAGE_SIZE);

```

```

data_buff = spiFlashBase.buff();
for(int n = 0; n<20; n++){
    Serial.print(*data_buff);
    Serial.print(",");
    data_buff++;
}
Serial.println("");

Serial.print("### Verify ");
verify = true;
data_buff = spiFlashBase.buff();
for(int n = 0; n<20; n++){
    if (*data_buff != (uint8_t)(n + (loops*10))) verify = false;
    data_buff++;
}
if (verify) {
    result++;
    Serial.println("OK ###");
}
else Serial.println("FAIL !!! ###");

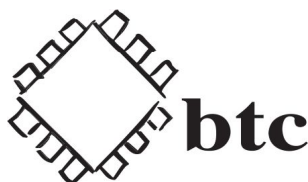
ledi++;
digitalWrite(LED_PIN, (ledi&1));
Serial.println();
Serial.println();
Serial.println();

delay(500);
loops++;
if (loops >= 3) {
    if (result == 3) Serial.println("### Verify OK 3 times ###");
    else Serial.println("!!! Some things are wrong, some verifications fail !!!!");
    Serial.println("### THATS ALL ###");
    while(1){};
}
}
}

```

Linki

- [Karta katalogowa W25Q80](#)
- [Karta katalogowa W25Q128](#)
- [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.