

KAmoRPi CAN-FD (PL)

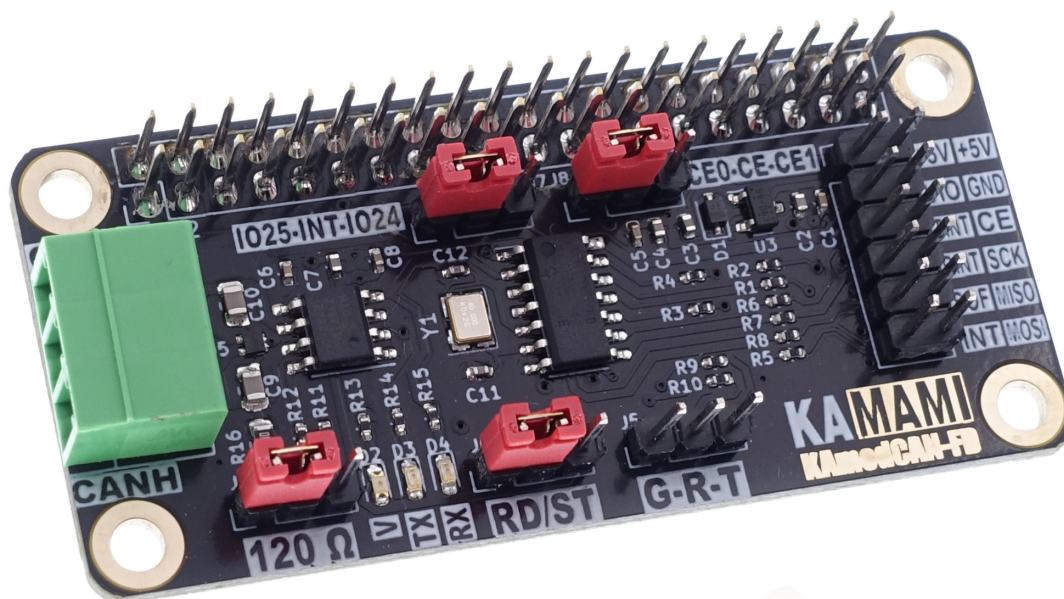


Spis treści

Opis	1
Podstawowe parametry	2
Wyposażenie standardowe	3
Schemat elektryczny	4
Złącze sterujące	5
Złącze GPIO w standardzie Raspberry Pi	6
Podłączenie dwóch modułów interfejsu CAN	7
Złącze magistrali CAN	8
Dołączenie rezystora terminującego	9
Wyłączenie transceivera magistrali CAN	10
Monitorowanie strumieni danych TX i RX	11
Zasilanie i napięcie sygnałów sterujących VIO	12
Diody sygnalizacyjne	13
Wymiary	14
Linki	15
Uruchomienie modułu z komputerkiem Raspberry Pi 5	15
Odblokowanie interfejsu SPI	16
Instalowanie pakietu narzędzi dla interfejsu CAN	17
Konfigurowanie interfejsu CAN	18
Restartowanie systemu	19
Sprawdzenie konfiguracji	20
Test komunikacji	21
Test komunikacji w trybie CAN FD	23

Opis

KAmoRPi CAN-FD jest kompletnym interfejsem magistrali CAN bazującym na zaawansowanym kontrolerze CAN typu MCP2518FD. Zastosowany układ odpowiada za formowanie prawidłowych ramek wysyłanych na magistralę, oraz buforowanie danych przesłanych do, oraz z magistrali CAN. Dzięki tym cechom może współpracować niemal z dowolnym komputerkiem SBC lub mikrokontrolerem. Komunikacja po stronie CAN może odbywać się z prędkością do 1 Mbps w trybie CAN 2.0 lub do 8 Mbps w trybie CAN FD. Komunikacja z kontrolerem odbywa się poprzez 4-przewodowy interfejs SPI (MISO, MOSI, SCK, CE) oraz dodatkowy sygnał przerwania (INT). Wszystkie sygnały są doprowadzone do 40 stykowego złącza w taki sposób, że 2 moduły mogą działać jednocześnie z jednym komputerkiem Raspberry Pi 5.



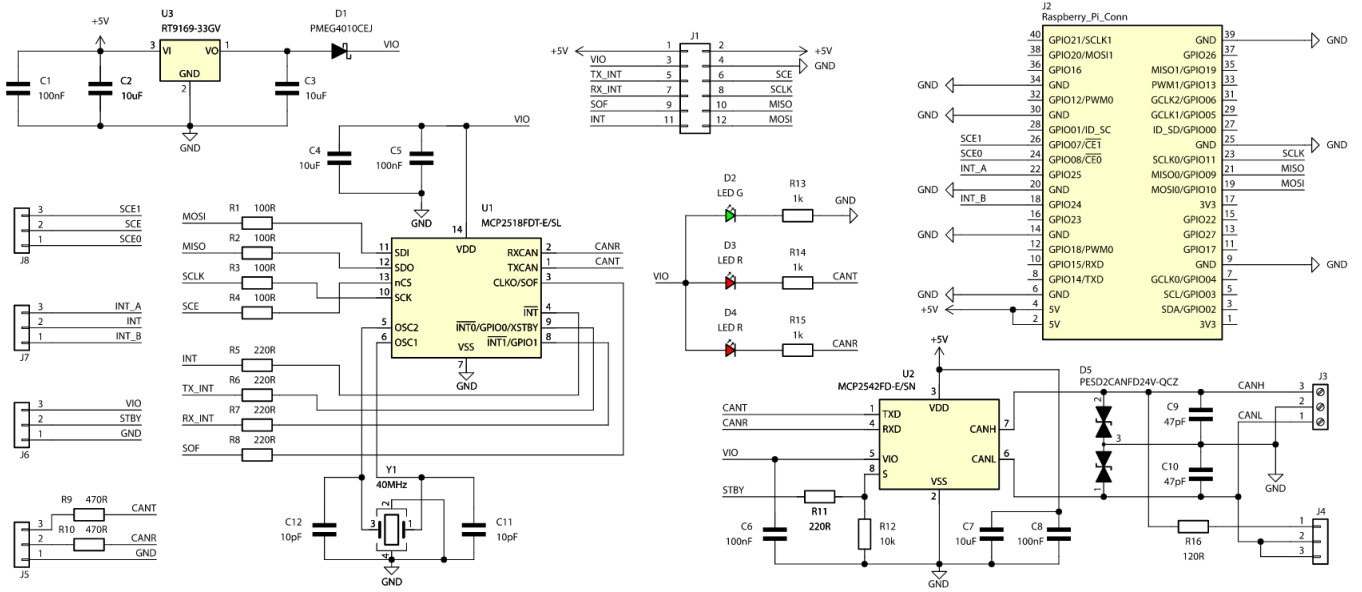
Podstawowe parametry

- Interfejs magistrali CAN, kompatybilny z CAN 2.0 oraz CAN FD
- Bazuje na kontrolerze CAN typu MCP2518FD
- Komunikacja poprzez interfejs SPI z maksymalnym taktowaniem 20 MHz
- Interfejs SPI działający z napięciem 3,3 V lub 5 V
- Prędkość komunikacji (bit rate): 125 kbps...1 Mbps (500 kbps...8 Mbps w trybie FD)
- Dołączany rezystor terminujący 120 Ω
- Diody LED sygnalizujące poprawne zasilanie oraz komunikację
- Kompatybilny z płytkami Raspberry Pi 5 oraz innymi z rodziny Raspberry Pi
- Możliwość dołączenia dwóch modułów do jednego komputerka (komunikacja poprzez jeden interfejs SPI z oddzielnymi liniami CE0/CE1)
- Linie magistrali CAN dołączane poprzez złącze Phoenix MC 3,81 mm
- Zabezpieczenie przed przepięciami na liniach magistrali CAN
- Zasilanie 5 V, 100 mA
- Wymiary płytki 65x30 mm, wysokość ok. 25 mm

Wyposażenie standardowe

Kod	Opis
KAmoD CAN-FD	Zmontowany i uruchomiony moduł

Schemat elektryczny



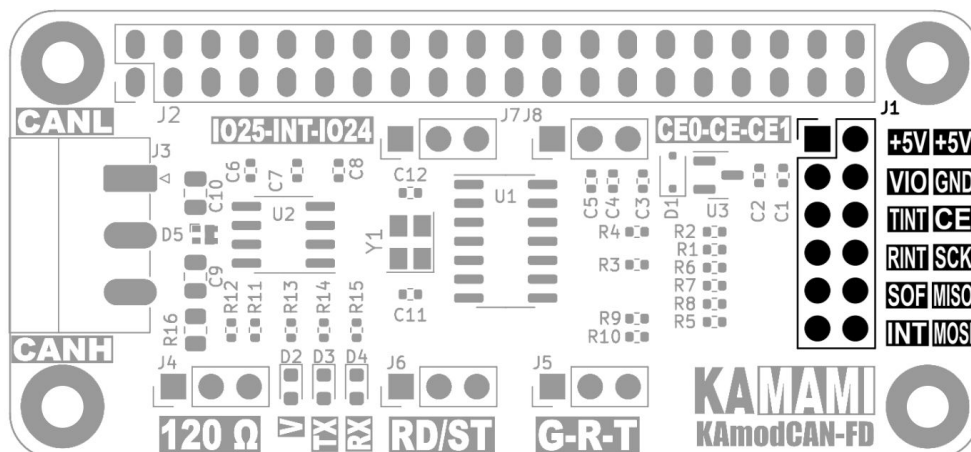
Złącze sterujące

Złącze	Funkcja
J1 Szpilki goldpin 1x6, 2,54 mm	<ul style="list-style-type: none"> Wyprowadzone sygnały sterujące kontrolera CAN Wejście zasilania

Kontroler CAN typu MCP2518FD jest sterowany poprzez interfejs SPI. Poprzez SPI odbywa się konfigurowanie parametrów pracy oraz przesyłanie danych do i z kontrolera. Dodatkowo kontroler udostępnia kilka dodatkowych sygnałów - przerw, które pozwalają szybko reagować na określone zdarzenia. Wszystkie linie działają z napięciem 3,3 V, ale opcjonalnie można je skonfigurować do pracy z napięciem 5 V.

Funkcje sygnałów na złączu **J1** są następujące:

- szpilki nr 1 i 2: **+5V** - wejście zasilania o napięciu 4,5...5,5 V;
- szpilka nr 3: **VIO** - opcjonalne wejście zasilania dla napięcia 3,3...5 V, jeśli sygnały sterujące mają działać z napięciem wyższym niż 3,3 V;
- szpilka nr 4: **GND** - masa zasilania i sygnałowa;
- szpilka nr 5: **TINT** - wyjście przerwania INTO (TX Interrupt output), aktywne w stanie niskim;
- szpilka nr 6: **CE** - wejście wyboru układu interfejsu SPI (SPI chip select input), aktywne w stanie niskim;
- szpilka nr 7: **RINT** - wyjście przerwania INT1 (RX Interrupt output), aktywne w stanie niskim;
- szpilka nr 8: **SCK** - wejście sygnału zegarowego interfejsu SPI kontrolera CAN (SPI clock input);
- szpilka nr 9: **SOF** - wyjście sygnalizujące start ramki (Start of Frame output);
- szpilka nr 10: **MISO** - wyjście danych SPI kontrolera CAN (SPI data output);
- szpilka nr 11: **INT** - główne wyjście przerwania, aktywne w stanie niskim;
- szpilka nr 12: **MOSI** - wejście danych SPI kontrolera CAN (SPI data input);

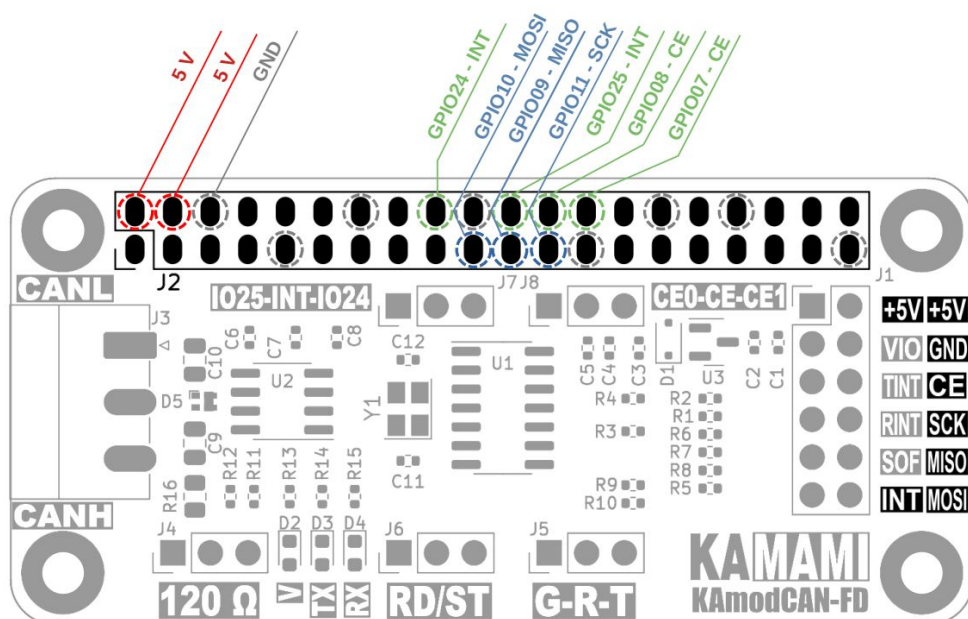


Złącze GPIO w standardzie Raspberry Pi

Złącze	Funkcja
J2 Złącze goldpin 2x20, 2,54 mm	<ul style="list-style-type: none"> Wyprowadzone sygnały sterujące kontrolera CAN, dołączone do odpowiednich styków złącza GPIO Wejście zasilania, dołączone do odpowiednich styków złącza GPIO

Sygnały sterujące kontrolerem CAN zostały doprowadzone do odpowiednich styków złącza **J2** (oraz równolegle do złącza J1), które jest kompatybilne ze standardem złącza GPIO płytek Raspberry Pi. Dzięki temu moduł KAmoD CAN-FD można łatwo dołączyć do SBC Raspberry Pi. Rozmieszczenie sygnałów sterujących na złączu GPIO jest następujące:

- pozycja nr 2 oznaczona „**3V3**” - zwarcie szpilek 3-4 oznacza, że szeregowy interfejs sterujący jest dostosowany do napięcia 3,3 V, które jest dostarczane ze zintegrowanego na płycie stabilizatora napięcia;
- styki 2 i 4: **+5V** - zasilanie 5 V doprowadzone ze złącza GPIO;
- styki 6, 9, 14, 20, 25, 30, 34, 39 - masa **GND** doprowadzona ze złącza GPIO;
- styk 18: **GPIO24** - może być połączony z sygnałem **INT** (zależy od ustawienia zworki J7);
- styk 22: **GPIO25** - może być połączony z sygnałem **INT** (zależy od ustawienia zworki J7);
- styk 24: **GPIO08** - może być połączony z sygnałem **CE** (zależy od ustawienia zworki J8);
- styk 26: **GPIO07** - może być połączony z sygnałem **CE** (zależy od ustawienia zworki J8);
- styk 19: **GPIO10** - połączony z sygnałem MOSI - wejście danych SPI kontrolera CAN;
- styk 19: **GPIO09** - połączony z sygnałem MISO - wyjście danych SPI kontrolera CAN;
- styk 19: **GPIO11** - połączony z sygnałem SCK - wejście zegarowe SPI kontrolera CAN;



Podłączenie dwóch modułów interfejsu CAN

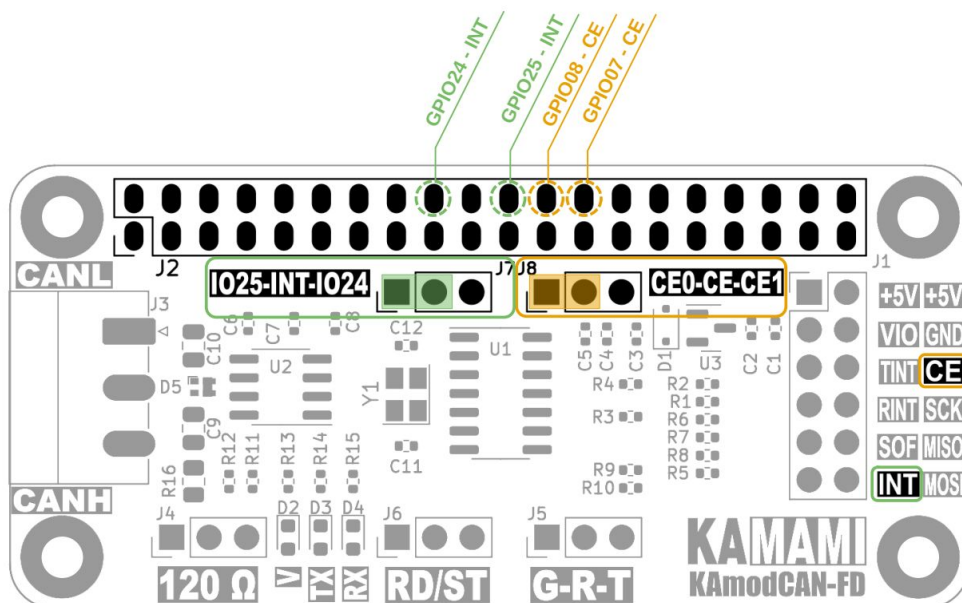
Złącze	Funkcja
J7, J8 Szpilki goldpin + jumper 1x3, 2,54 mm	<ul style="list-style-type: none"> Wybór kanału interfejsu SPI dla kontrolera CAN (sygnały CE0/CE1 na złączu GPIO) Wybór wejścia dla przerwania INT z kontrolera CAN (sygnały GPIO24/GPIO25 na złączu GPIO)

Moduł KAmoD CAN-FD jest sterowany poprzez interfejs szeregowy SPI, który może obsługiwać kilka kontrolerów w jednej aplikacji. Każdy kontroler powinien mieć przypisaną dla siebie linię wyboru **CE** (Chip Enable). Na złączu GPIO Raspberry Pi są dostępne dwie linie CE - CE0 oraz CE1. Za pomocą zworki **J8** można wybrać, z której linii CE będzie korzystał kontroler CAN na płycie modułu KAmoD CAN-FD:

- zwarcie szpilek 1-2 J8: ustawia linię CE0, jako linię wyboru kontrolera (jak na rysunku),
- zwarcie szpilek 2-3 J8: ustawia linię CE1, jako linię wyboru kontrolera.

Kontroler CAN typu MCP2518FD, do prawidłowego działania wymaga obsługi przerwania ustawianego stanem na linii **INT**. Aby umożliwić działanie dwóch kontrolerów CAN, sygnał INT można dołączyć do jednego z dwóch styków złącza GPIO, za pomocą zworki **J7**:

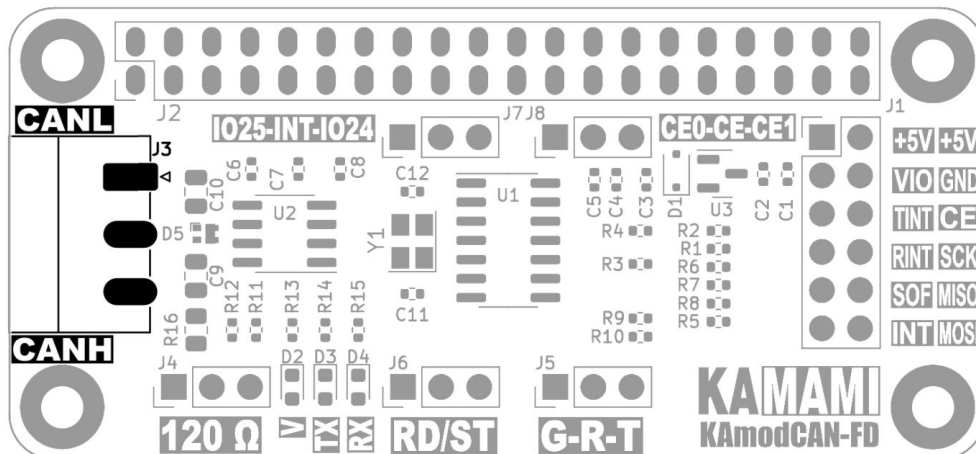
- zwarcie szpilek 1-2 J7: sygnał INT połączony z linią GPIO25 (jak na rysunku),
- zwarcie szpilek 2-3 J7: sygnał INT połączony z linią GPIO24.



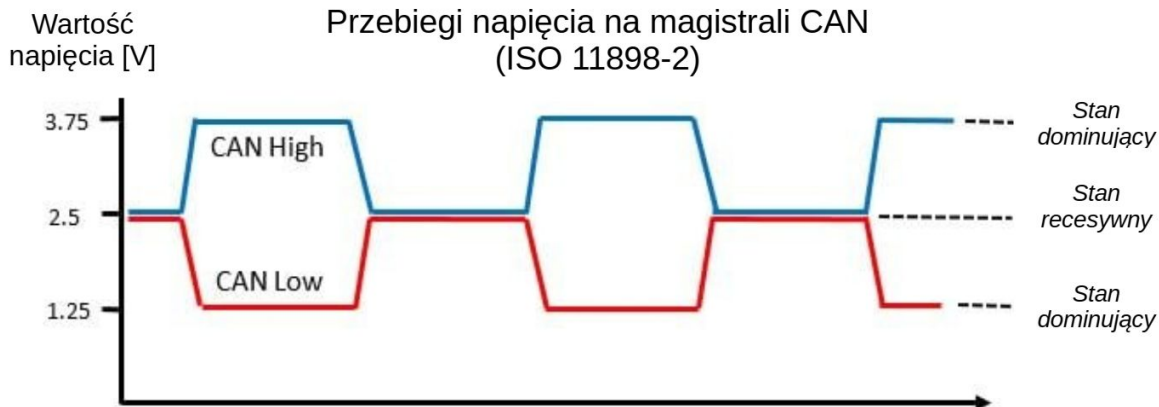
Złącze magistrali CAN

Złącze	Funkcja
J3 - CAN Phoenix MC 3,81 mm	• Złącze magistrali CAN

Złącze magistrali CAN zawiera 3 styki: **CANL**, **CANH** oraz **GND**. Ich rozmieszczenie jest opisane na płytce modułu, styk **GND to styk środkowy**. Każdy styk należy dołączyć do magistrali CAN zgodnie z oznaczeniem. Sygnał GND powinien być dołączony do wspólnej masy zasilania urządzeń połączonych magistralą CAN.



Przebiegi napięć na magistrali CAN pokazuje poniższy rysunek:

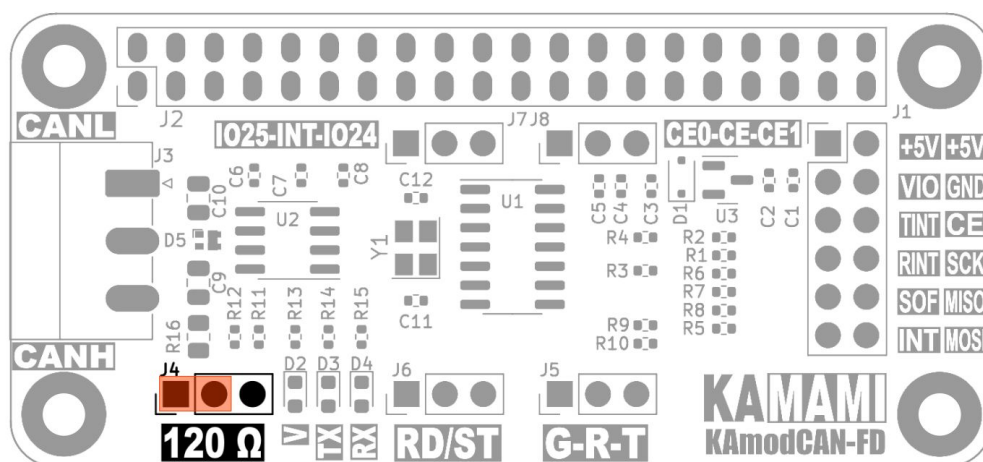


Dołączenie rezystora terminującego

Złącze	Funkcja
J4 Szpilki goldpin + jumper 1x3, 2,54 mm	• Dołączenie rezystora terminującego 120 Ω

Urządzenia w magistrali CAN, jak sama nazwa wskazuje, są połączone w topologii magistrali. Jest to jedna szyna, bez rozgałęzień, w której można wskazać dwa końce. Każdy koniec magistrali powinien być wyposażony w terminator magistrali - w przypadku magistrali CAN jest to rezystor o wartości **120 Ω**. Na płycie modułu znajduje się odpowiedni rezystor, który można dołączyć poprzez odpowiednie ustawienie zworki **J4**:

- zwarcie szpilek 1-2, oznacza, że rezystor terminujący 120 Ω jest dołączony pomiędzy liniami CANL i CANH (jak na rysunku);
- zwarcie szpilek 2-3, lub usunięcie zworki powoduje odłączenie rezystora terminującego.

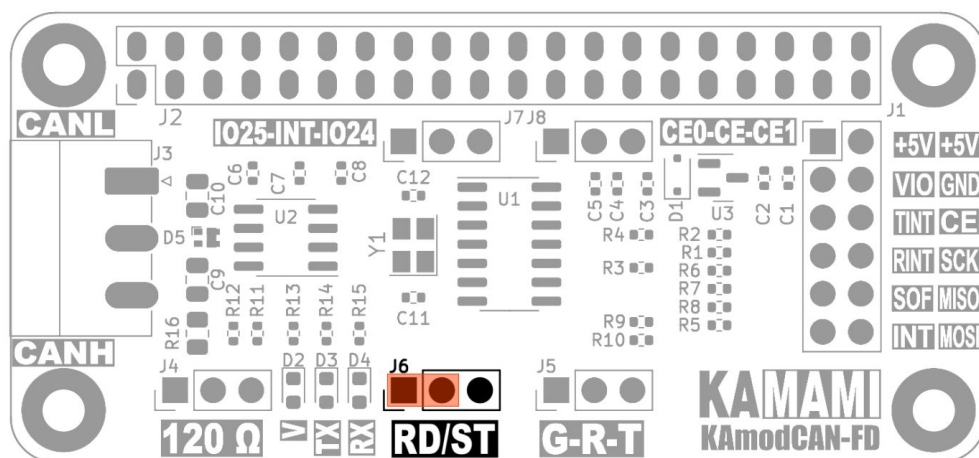


Wyłączenie transceivera magistrali CAN

Złącze	Funkcja
J6 Szpilki goldpin + jumper 1x3, 2,54 mm	• Włączenie/wyłączenie transceivera magistrali CAN

W trakcie prób i testów komunikacji poprzez CAN, może istnieć potrzeba odcinania dostępu do magistrali, aby testowana aplikacja nie zakłócała komunikacji innych urządzeń. Moduł KAMod CAN-FD zawiera transceiver typu **MCP2542FD**, który dostosowuje strumień danych z kontrolera CAN do parametrów elektrycznych magistrali. Szybkim sposobem odcięcia dostępu do magistrali, bez modyfikacji okablowania, jest wyłączenie transceivera - ustawienie trybu uśpienia *standby*. Za pomocą szpilek **J6** można w łatwy sposób włączyć/wyłączyć transceiver:

- zwarcie szpilek 1-2: transceiver aktywny (jak na rysunku),
- zwarcie szpilek 2-3: transceiver w trybie uśpienia *standby*.

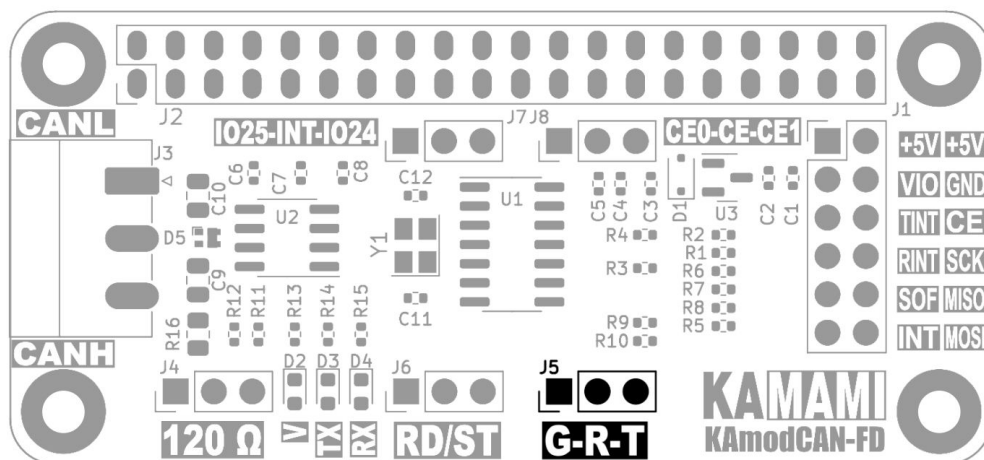


Monitorowanie strumieni danych TX i RX

Złącze	Funkcja
J5 Szpilki goldpin + jumper 1x3, 2,54 mm	• Monitorowanie strumieni danych nadawanych i odbieranych z magistrali CAN

Na złączu **J5** modułu KAMod CAN-FD, wyprowadzone są sygnały strumieni danych wysyłanych z kontrolera i do kontrolera CAN. Są to zwykłe sygnały logiczne, dlatego można je łatwo monitorować i rejestrować np. za pomocą analizatora logicznego. Poziom napięcia sygnałów odpowiada napięciu **VIO**. Monitorowanie tych sygnałów może być przydatne na etapie uruchamiania aplikacji i wyszukiwania błędów. Rozmieszczenie sygnałów jest następujące:

- szpilka nr 1: **G** - masa sygnałów,
- szpilka nr 2: **R** - strumień danych odebranych z magistrali CAN,
- szpilka nr 3: **T** - strumień danych wysyłanych do magistrali CAN z kontrolera.



Zasilanie i napięcie sygnałów sterujących VIO

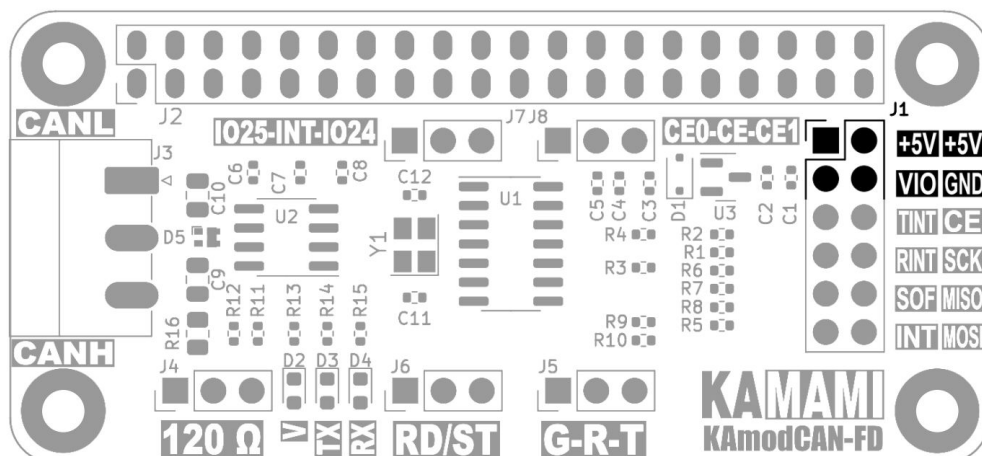
Złącze	Funkcja
J1 Szpilki goldpin 2x6, 2,54 mm	<ul style="list-style-type: none"> Wejście zasilania Ustawienie napięcia sygnałów sterujących VIO

Zastosowany w module KAmoD CAN-FD transceiver CAN (MCP2542FD) wymaga zasilania o napięciu z zakresu 4,5...5,5 V, które należy dołączyć do złącza J1, do styków +5V i GND. Natomiast zasilanie kontrolera CAN (MCP2518FD), może być niższe - musi zawierać się w zakresie 3...5,5 V. Napięcie sygnałów sterujących na złączu J1 będzie takie samo, jak napięcie zasilania kontrolera.

Kontroler CAN jest zasilany napięciem 3,3 V z wbudowanego w moduł stabilizatora napięcia. Napięcie to jest wyprowadzone na styku VIO złącza J1. Podając na styk VIO napięcie wyższe niż 3 V, np. 5 V, można uzyskać wyższe napięcie sygnałów sterujących - np. 5V.

Funkcje styków zasilania na złączu **J1** są następujące:

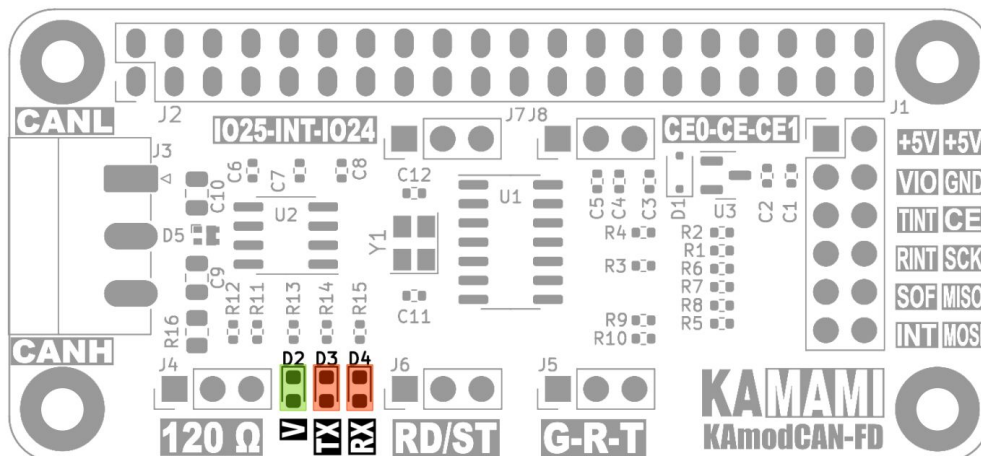
- szpilki nr 1 i 2: **+5V** - wejście zasilania o napięciu 4,5...5,5 V;
- szpilka nr 3: **VIO** - opcjonalne wejście zasilania dla napięcia 3,3...5 V, jeśli sygnały sterujące mają działać z napięciem wyższym niż 3,3 V;
- szpilka nr 4: **GND** - masa zasilania i sygnałowa;



Diody sygnalizacyjne

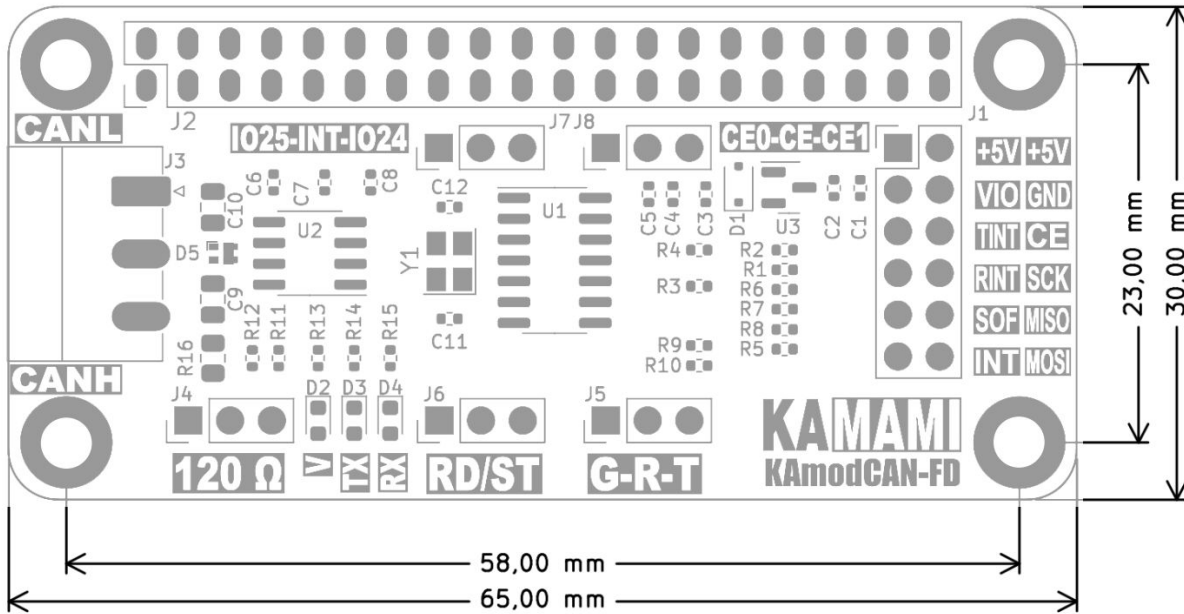
Typ	Funkcja
RX	• RX - sygnalizacja odczytu danych z magistrali CAN
TX	• TX - sygnalizacja nadawania danych na magistralę
V	• V - sygnalizacja napięcia kontrolera CAN

Diody LED **RX** i **TX** sygnalizują stan aktywny - czyli niski stan logiczny (stan dominujący na magistrali CAN), odpowiednio na wyjściu i wejściu transceivera CAN. W przypadku przesyłania danych na magistralę, będą migały obie diody - TX oraz RX, ponieważ dane są jednocześnie odczytywane przez transceiver. Dioda **V** sygnalizuje obecność napięcia kontrolera CAN.



Wymiary

Wymiary modułu interfejsu KAmoD CAN-FD to 65x30 mm, i wysokość ok. 25 mm. Moduł jest kompatybilny z płytkami Raspberry Pi 5 oraz Raspberry Pi Zero.



Linki

- [Karta katalogowa układu MCP2518](#)
- [Karta katalogowa układu MCP2542FD](#)
- [Artykuł na portalu MIKROKONTROLER.pl "Interfejs komunikacyjny CAN: podstawy"](#)
- [Repozytorium narzędzi Linux CAN](#)

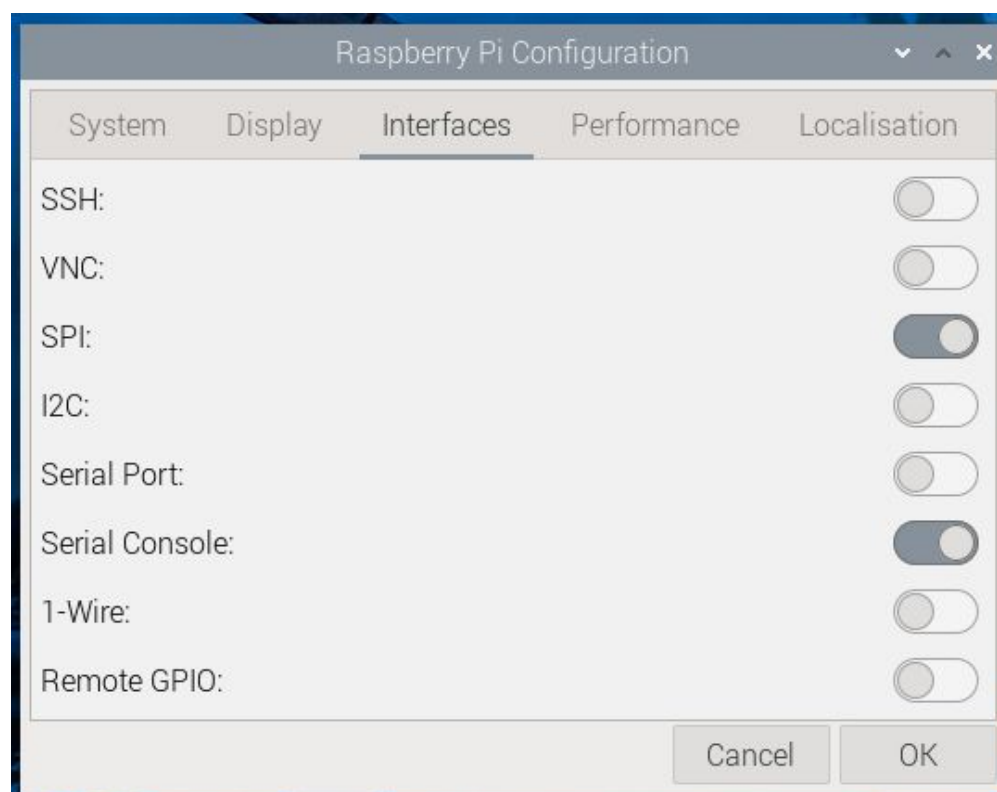
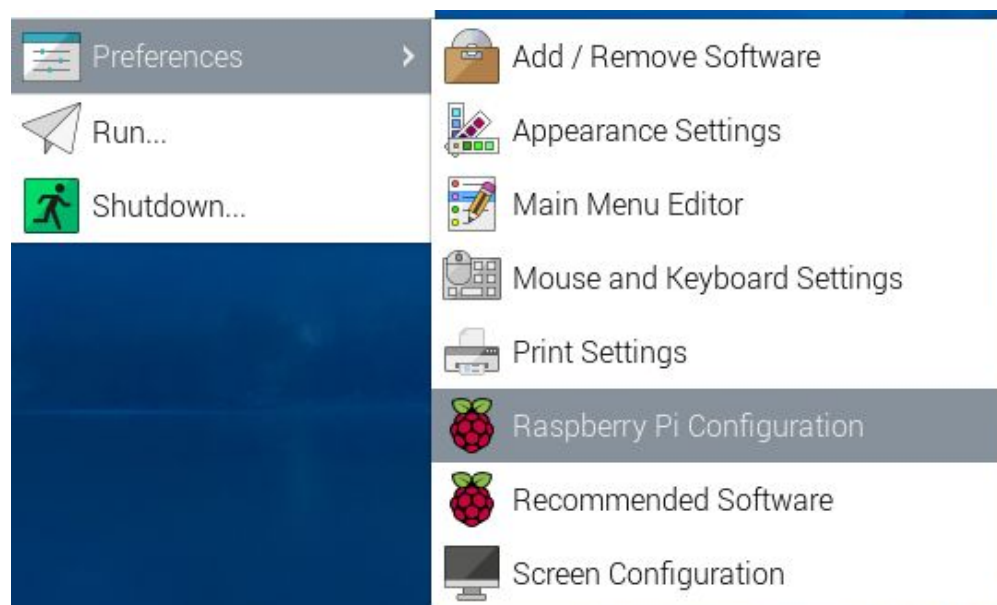
Uruchomienie modułu z komputerkiem Raspberry Pi 5

Uruchomienie interfejsu CAN z połączeniu z komputerkiem Raspberry Pi 5 wymaga wykonania następujących kroków.

Odblokowanie interfejsu SPI

Uruchamiamy program do konfiguracji Raspberry Pi - *Raspberry Pi Configuration*

Wybieramy zakładkę *Interfaces*, a tam przełączamy w pozycję aktywną przełącznik przy *SPI*. Zapisujemy i zamykamy program.



Instalowanie pakietu narzędzi dla interfejsu CAN

Uruchamiamy okno terminala, np. za pomocą klawiszy *Ctrl+Alt+T*, a następnie wpisujemy komendę:

```
sudo apt-get install can-utils
```

Czekamy do zakończenia i potwierdzamy wszystkie zapytania.

Konfigurowanie interfejsu CAN

Uruchamiamy okno terminala, np. za pomocą klawiszy *Ctrl+Alt+T*. W programie do edycji plików tekstowych nano otwieramy plik `config.txt`, wpisując komendę:

```
sudo nano /boot/firmware/config.txt
```

(we wcześniejszych wersjach systemu Raspbian, plik `config.txt` jest tu: `/boot/config.txt`)

Sprawdzamy, czy linia o treści:

```
dtparam=spi=on
```

nie ma znaku `#` jeśli tak, to usuwamy znak `#`

Na końcu pliku dopisujemy:

```
dtoverlay=mcp251xfd,spi0-0,oscillator=40000000,interrupt=25
```

Wpis ten odnosi się do ustawienia zworek na płytce modułu (J7: 1-2; J8: 1-2)

Jeśli zamierzamy dołączyć dwa moduły interfejsu CAN, to należy dopisać kolejną linię:

```
dtoverlay=mcp251xfd,spi0-1,oscillator=40000000,interrupt=24
```

Wpis ten odnosi się do ustawienia zworek na płytce modułu (J7: 2-3; J8: 2-3)

```
# Run as fast as firmware / board allows
arm_boost=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

dtoverlay=mcp251xfd,spi0-0,oscillator=40000000,interrupt=25
dtoverlay=mcp251xfd,spi0-1,oscillator=40000000,interrupt=24

^G Help          ^O Write Out    ^W Where Is     ^K Cut
^X Exit          ^R Read File    ^\ Replace      ^U Paste
```

Zapisujemy zmiany naciskając *Ctrl+O* i potwierdzając klawiszem *Enter*, a następnie zamykamy edytor naciskając *Ctrl+X*

Restartowanie systemu

Restartujemy system, np. za pomocą polecenia:

```
sudo reboot
```

Sprawdzenie konfiguracji

Uruchamiamy okno terminala, np. za pomocą klawiszy *Ctrl+Alt+T*. Wpisujemy polecenie:

```
dmesg | grep spi
```

które wyświetli informacje dotyczące interfejsu SPI zebrane w czasie uruchamiania systemu:

```
pi@raspberrypi:~$ dmesg | grep spi
[ 2.810691] spi_master spi0: will run message pump with realtime priority
[ 2.848478] mcp251xfd spi0.1 can0: MCP2518FD rev0.0 (-RX_INT -PLL -MAB_NO_WARN +CRC_REG +CRC_RX +CRC_TX +ECC -HD o:40.00MHz c:40.00MHz m:20.00MHz rs:17.00MHz es:16.66MHz rf:17.00MHz ef:16.66MHz) successfully initialized.
[ 2.900261] mcp251xfd spi0.0 can1: MCP2518FD rev0.0 (-RX_INT -PLL -MAB_NO_WARN +CRC_REG +CRC_RX +CRC_TX +ECC -HD o:40.00MHz c:40.00MHz m:20.00MHz rs:17.00MHz es:16.66MHz rf:17.00MHz ef:16.66MHz) successfully initialized.
pi@raspberrypi:~$
```

Jeśli wszystko zostało prawidłowo skonfigurowane powinniśmy zobaczyć dwa wpisy (lub jeden, jeśli zamontowano tylko jeden moduł): *mcp251xfd spi0.0* oraz *mcp251xfd spi0.1*. co oznacza, że kontrolery CAN zostały prawidłowo zainicjowane - *successfully initialized*.

Teraz należy wpisać polecenie:

```
ifconfig -a
```

które wyświetli listę dostępnych interfejsów sieciowych - Ethernet, Wi-Fi oraz właśnie CAN:

```
pi@raspberrypi:~$ ifconfig -a
can0: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 190

can1: flags=193<UP,RUNNING,NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 193
```

Test komunikacji

Najprostszy test komunikacji można wykonać mając dwa moduły interfejsu CAN KAmoD CAN-FD podłączone do jednego komputera Raspberry Pi 5. Jednak analogicznie będzie przebiegał test z dwoma komputerkami wyposażonymi w interfejs KAmoD CAN-FD.



Uruchamiamy okno terminana, np. za pomocą klawiszy *Ctrl+Alt+T*. Aktywujemy magistralę CAN poleceniem:

```
sudo ip link set can0 up type can bitrate 1000000
```

gdzie: *can0* - to numer interfejsu, dla drugiego modułu będzie *can1*
bitrate 1000000 - oznacza prędkość komunikacji, w tym przypadku ustawiamy 1 Mbps

Dla drugiego modułu należy wpisać analogiczne polecenie:

```
sudo ip link set can1 up type can bitrate 1000000
```

Teraz otwieramy drugi terminal - w jednym oknie będziemy obserwować dane odebrane za pomocą jednego interfejsu CAN (np. *can0*), a w drugim oknie będziemy wysyłać ramki danych za pomocą drugiego interfejsu CAN (np. *can1*). W pierwszym oknie wpisujemy polecenie:

```
candump can0
```

które sprawia, że interfejs *can0* wyświetla w terminalu wszystkie odebrane dane.

W drugim oknie wpisujemy polecenie:

```
cansend can1 456#214234
```

Efekt widać na poniższych zrzutach ekranów:

```
pi@raspberrypi:~ $ sudo ip link set can1 up type can bitrate 1000000
pi@raspberrypi:~ $ sudo ip link set can0 up type can bitrate 1000000
pi@raspberrypi:~ $ candump can0
can0 456 [3] 21 42 34
can0 456 [3] 21 42 34
```

```
pi@raspberrypi:~ $ cansend can1 456#214234
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cansend can1 456#214234
pi@raspberrypi:~ $
```

Polecenie `cansend can1 456#214234` powoduje wysłanie ramki CAN o identyfikatorze 456 z 3 bajtami danych: 21, 42 i 34. W standardzie CAN 2.0B identyfikator ma długość 29 bitów, a ramka może zawierać do 8 bajtów danych. Polecenie `candump can0` monitoruje stan magistrali i wyświetla wszystkie odebrane ramki danych. Aby przerwać działanie tego procesu należy nacisnąć `Ctrl+C`.

Aby zmienić parametry komunikacji należy najpierw dezaktywować magistrale CAN poleceniami

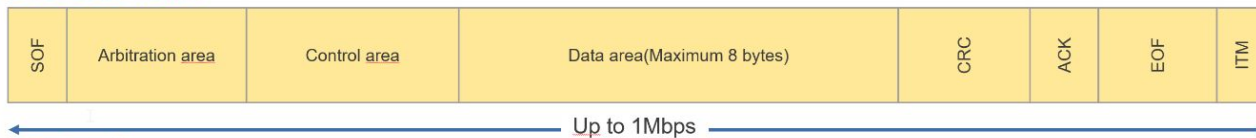
```
sudo ifconfig can0 down
```

```
sudo ifconfig can1 down
```

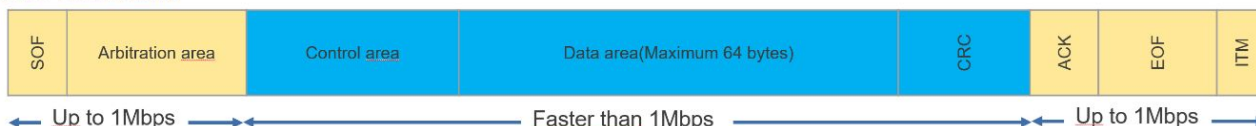
Test komunikacji w trybie CAN FD

Standard CAN2.0 pozwala na przesłanie do 8 bajtów danych z prędkością do 1 Mbps. Standard CAN FD (Controller Area Network Flexible Data-Rate) pozwala na przesyłanie do 64 bajtów danych w jednej ramce, poprzez zwiększenie szybkości komunikacji wewnątrz ramki - w czasie przesyłania danych, do 8 Mbps:

▪ CAN 2.0B data frame



▪ CAN FD data frame



Moduł KAMod CAN-FD jest kompatybilny ze standardem CAN FD. Aby przetestować moduł w tym trybie należy wykonać następujące czynności.

Uruchamiamy okno terminana, np. za pomocą klawiszy Ctrl+Alt+T. Aktywujemy magistralę CAN poleceniem:

```
sudo ip link set can0 up type can bitrate 1000000 dbitrate 8000000 restart-ms 1000 berr-reporting on fd on
```

Analogicznie postępujemy z drugą magistralą:

```
sudo ip link set can1 up type can bitrate 1000000 dbitrate 8000000 restart-ms 1000 berr-reporting on fd on
```

Teraz użyjemy narzędzia do generowania losowych ramek CAN - cangen. W oknie pierwszego terminala wpisujemy polecenie:

```
cangen can0 -mv
```

W oknie drugiego terminala wpisujemy polecenie:

```
candump can1
```

Efekt jest widoczny na poniższym zrzucie ekranu:

Wśród wygenerowanych ramek widać takie, które nie niosą żadnych danych, tylko sam identyfikator oraz takie, które niosą 20 bajtów danych - zgodnie z możliwościami CAN FD.



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.