



KAmoD STM32H503 (PL)



Rev. 20260820104800

Źródło: [https://wiki.kamamilabs.com/index.php?title=KAmoD_STM32H503_\(PL\)](https://wiki.kamamilabs.com/index.php?title=KAmoD_STM32H503_(PL))

Spis treści

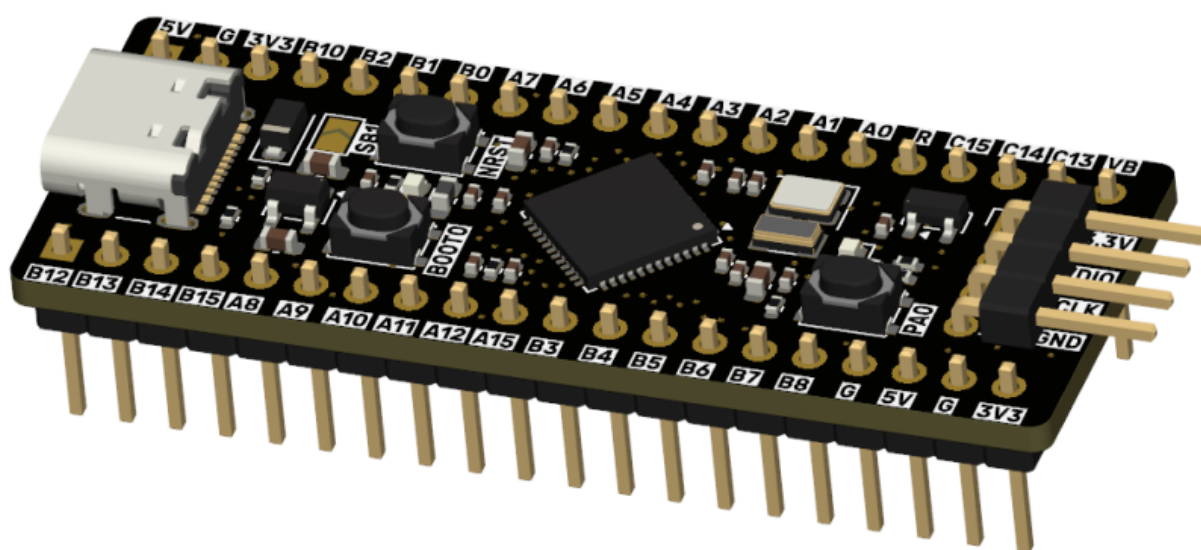
Opis	1
Podstawowe parametry	1
Wyposażenie standardowe	2
Schemat elektryczny	2
Zasilanie	2
Zasilanie RTC	3
Zworka SB1	3
Interfejs USB	3
Interfejs programowania i debugowania SWD	4
Programowanie mikrokontrolera	4
Programowanie przez SWD	4
Programowanie przez USB DFU	4
Dodatkowe elementy - przyciski i diody LED	5
Pamięć SPI Flash	5
Złącza GPIO	6
Złącze P1	7
Złącze P2	8
Zworki SB2 (PC14) i SB3 (PC15)	8
Wymiary	9
Program testowy	9
Szczegółowy opis protokołu i odpowiedzi	10
USB CDC CLI — dokumentacja komend	10
Lista komend	11
Format daty i czasu	12
Test pamięci SPI Flash	13
Format odpowiedzi	14
Sprawdzanie i synchronizacja RTC z komputerem	14
Linki	15

Opis

KAmoD STM32H503 - Płytkę ewaluacyjną z mikrokontrolerem STM32H503CBU6 Cortex-M33

KAmoD STM32H503 jest niewielką płytką ewaluacyjną z mikrokontrolerem STM32H503CBU6. Moduł zawiera elementy niezbędne do uruchomienia mikrokontrolera, rezonatory kwarcowe 8 MHz i 32,768 kHz, złącze USB-C, stabilizator 3,3 V oraz przyciski i diody LED. Na płytce przewidziano również nieobsadzone miejsce na opcjonalną pamięć SPI NOR Flash do samodzielnego wlutowania.

Złącze USB-C może służyć do zasilania, komunikacji USB 2.0 Full Speed oraz programowania mikrokontrolera przez fabryczny bootloader USB DFU. Interfejs SWD wyprowadzono na osobne złącze. Pozostałe sygnały mikrokontrolera są dostępne na dwóch listwach po 20 styków, o rastrze 2,54 mm.



Podstawowe parametry

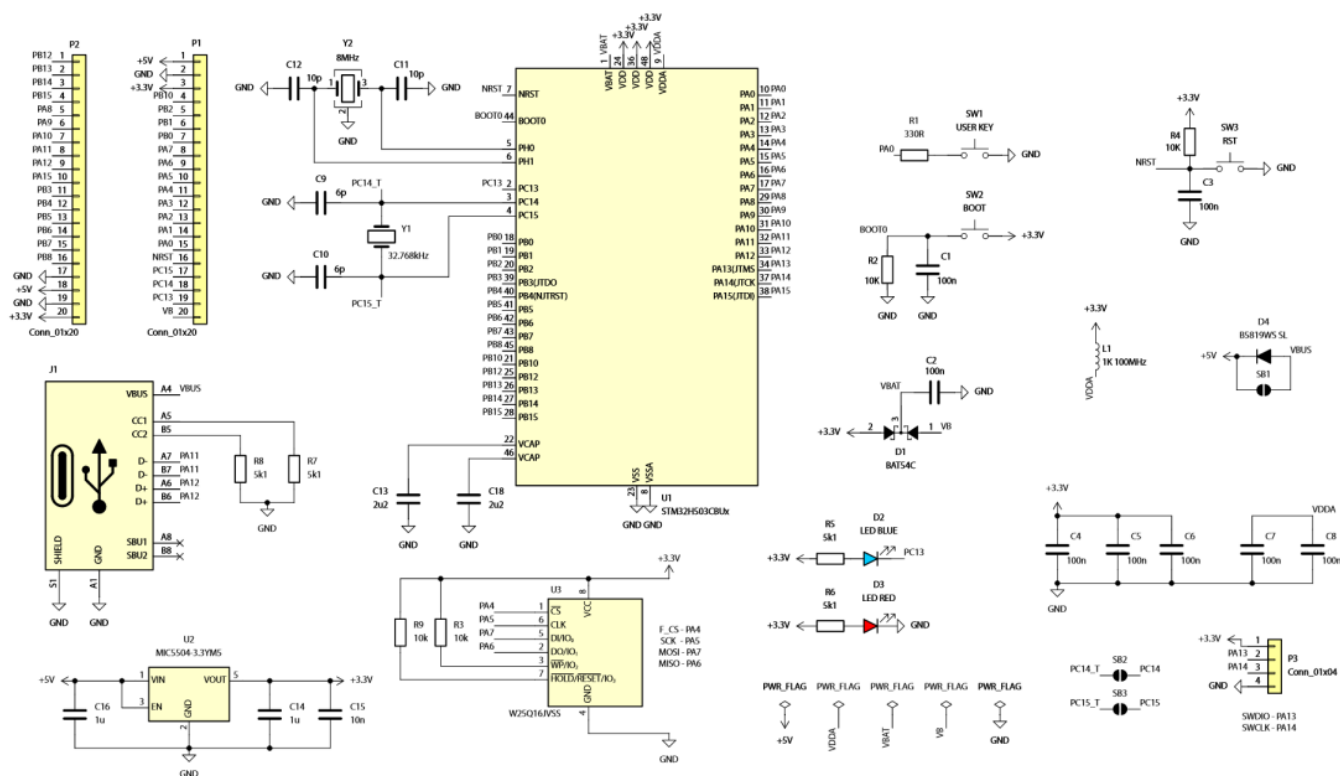
- mikrokontroler STM32H503CBU6 w obudowie UFQFPN48:
 - rdzeń Arm Cortex-M33 z FPU i instrukcjami DSP, taktowany do 250 MHz,
 - 128 kB pamięci Flash z ECC,
 - 32 kB pamięci SRAM z ECC oraz 2 kB podrzmywanej pamięci SRAM,
 - 8 kB pamięci podręcznej instrukcji,
 - 12-bitowy ADC, dwukanałowy 12-bitowy DAC, komparator i wzmacniacz operacyjny,
 - timery 16- i 32-bitowe, RTC, dwa układy watchdog,
 - interfejsy I2C/I3C, USART/LPUART, SPI/I2S, FDCAN oraz USB 2.0 Full Speed,
- precyzyjny rezonator kwarcowy 8 MHz dla zegara głównego,
- rezonator 32,768 kHz dla RTC,
- możliwość dołączenia baterii podtrzymującej RTC,
- nieobsadzone miejsce U3 na opcjonalną pamięć SPI NOR Flash; W25Q16JV 16 Mbit (2 MB) jest przykładem zgodnego układu,
- złącze USB-C pracujące w trybie urządzenia,
- stabilizator 3,3 V MIC5504 o maksymalnym prądzie wyjściowym 300 mA,
- 33 sygnały GPIO dostępne na złączach P1, P2 i P3; część z nich jest współdzielona z układami płytki,

- złącze programowania i debugowania SWD,
- przyciski USER, BOOT i RST,
- dioda zasilania oraz dioda LED sterowana przez PC13,
- złącza P1 i P2: po 20 styków, raster 2,54 mm,
- złącze SWD P3: 4 styki, raster 2,54 mm,
- wymiary płytki: 53 × 20,75 mm.

Wypożenie standardowe

Komponent	Opis
KAmoD STM32H503	• Zmontowany i uruchomiony moduł • 2 x prosta listwa goldpin 20-pin raster 2,54 mm • 1 x kątowa listwa goldpin 4-pin raster 2,54 mm

Schemat elektryczny



Zasilanie

Złącze	Styki	Funkcja
USB-C (J1)	VBUS, GND	Doprowadzenie napięcia 5 V ze źródła USB
P1	1:+5V, 2:GND, 3:+3.3V	Zasilanie płytki lub dostęp do linii zasilających
P2	17:GND, 18:+5V, 19:GND, 20:+3.3V	Zasilanie płytki lub dostęp do linii zasilających
P3	20:VB	Wejście baterii podtrzymującej domenę VBAT i RTC

Optymalnym sposobem zasilania płytki KAmoD STM32H503 jest podłączenie do interfejsu USB poprzez złącze USB-C. Napięcie ze złącza USB-C (VBUS) przechodzi przez diodę Schottky'ego D4 na linię +5V. Innym sposobem zasilania płytki jest doprowadzenie stabilizowanego napięcia 5 V bezpośrednio do styku +5V jednego ze złączy P1 lub P2, oraz masę do styku GND. Napięcie zasilające musi mieścić się w przedziale 4,5...5,5 V. Dioda D4 ogranicza możliwość przepływu prądu z płytki do złącza USB.

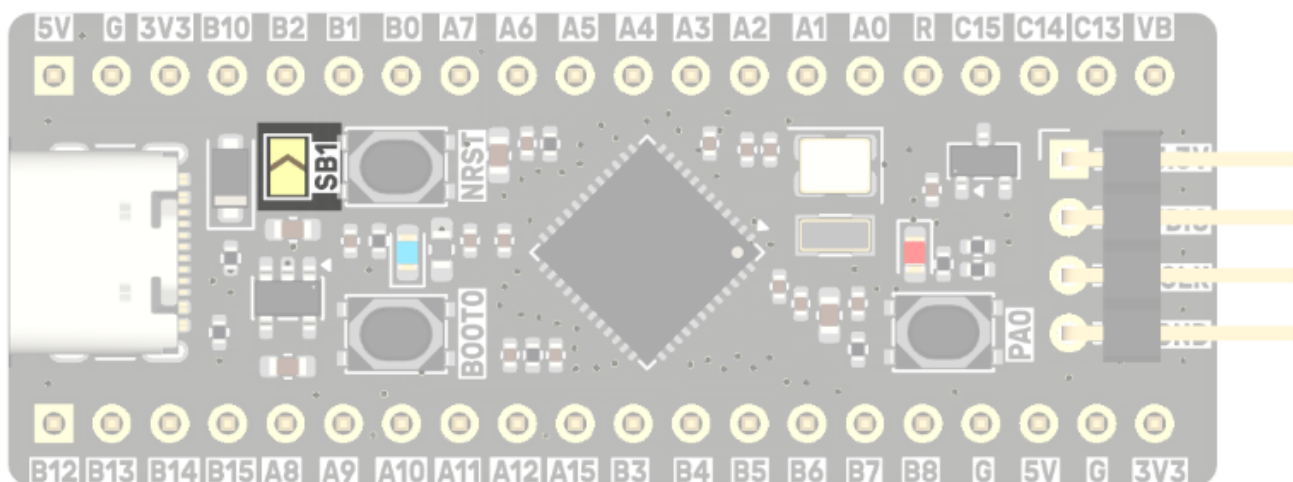
Stabilizowane napięcie 3,3 V jest otrzymywane ze stabilizatora MIC5504 i jest dostępne na stykach +3.3V przede wszystkim jako wyjście zasilające dla dołączanych układów. Czerwona dioda D3 sygnalizuje obecność napięcia 3,3 V. Maksymalna wydajność tego źródła to 300 mA, ale część tej puli pobiera mikrokontroler i pozostałe elementy modułu. Po zamontowaniu opcjonalnej pamięci Flash należy uwzględnić również pobierany przez nią prąd.

Zasilanie RTC

Wejście VB na styku 20 złącza P1 jest przeznaczone do dołączenia dodatniego bieguna baterii podtrzymującej RTC; biegun ujemny należy połączyć z GND. Układ D1 automatycznie zasilą domenę VBAT z linii 3,3 V podczas normalnej pracy, a po zaniku głównego zasilania przełącza ją na baterię. Napięcie na pinie VBAT mikrokontrolera musi mieścić się w zakresie określonym w jego karcie katalogowej (1,2...3,6 V). Należy uwzględnić spadek napięcia na diodzie Schottky'ego D1; typowym wyborem jest bateria litowa 3 V.

Zworka SB1

Otwarta fabrycznie zworka lutowana SB1 jest połączona równolegle do diody D4. Jej zwarcie omija spadek napięcia na diodzie, ale usuwa zabezpieczenie przed przepływem prądu z linii +5V do VBUS złącza USB. W typowym zastosowaniu SB1 powinna pozostać otwarta. Nie należy zwierać SB1, gdy płytki może być jednocześnie zasilana z USB i ze złącza P1 lub P2.



Interfejs USB

Złącze	Funkcja
--------	---------

USB-C (J1) Zasilanie 5 V, USB 2.0 Full Speed Device, fabryczny bootloader USB DFU

Mikrokontroler ma zintegrowany kontroler USB 2.0 Full Speed. Linie złącza są połączone bezpośrednio z wyprowadzeniami:

PA11 – USB DM

PA12 – USB DP

Te same sygnały są dostępne odpowiednio na stykach 8 i 9 złącza P2. Gdy używany jest interfejs USB, nie należy obciążać tych styków innymi układami. Rezystory 5,1 kΩ na liniach CC konfiguruje złącze USB-C jako urządzenie pobierające zasilanie. Płytkę nie zawiera obwodu zasilania VBUS wymaganego do pracy jako host USB, dlatego jest przeznaczona do pracy w trybie Device.

Funkcja USB zależy od oprogramowania mikrokontrolera. Dołączony projekt przykładowy konfiguruje urządzenie CDC ACM, widoczne w komputerze jako wirtualny port szeregowy.

Interfejs programowania i debugowania SWD

Złącze P3 udostępnia interfejs SWD (Serial Wire Debug), który umożliwia programowanie pamięci Flash oraz debugowanie programu przy użyciu zewnętrznego programatora, np. ST-LINK.

Styk P3	Sygnał	Funkcja
1	+3.3V	Napięcie odniesienia interfejsu programatora
2	PA13/SWDIO	Dane SWD
3	PA14/SWCLK	Zegar SWD
4	GND	Masa

Sygnały należy połączyć z odpowiadającymi im sygnałami programatora. Wyprowadzenie 3,3 V służy programatorowi jako odniesienie poziomów logicznych; płytkę powinna mieć własne zasilanie, chyba że używany programator jawnie obsługuje zasilanie układu docelowego.

Programowanie mikrokontrolera

Mikrokontroler można programować przez:

- złącze P3 i interfejs SWD,
- złącze USB-C i fabryczny bootloader USB DFU,
- inne interfejsy obsługiwane przez bootloader STM32H503, jeżeli zostaną odpowiednio dołączone do złączy GPIO.

Programowanie przez SWD

1. Podłącz programator do złącza P3 zgodnie z tabelą pinów.
2. Zasil płytkę przez USB-C albo linię +5V.
3. W STM32CubeProgrammer wybierz interfejs ST-LINK i połącz się z mikrokontrolerem.
4. Wybierz plik programu, zaprogramuj pamięć Flash i wykonaj weryfikację.

Programowanie przez USB DFU

Fabryczny bootloader znajduje się w pamięci systemowej mikrokontrolera i nie jest kasowany podczas zwykłego programowania pamięci Flash. Aby go uruchomić:

1. Podłącz płytkę przewodem USB-C do komputera.
2. Naciśnij i przytrzymaj przycisk B00T (SW2).

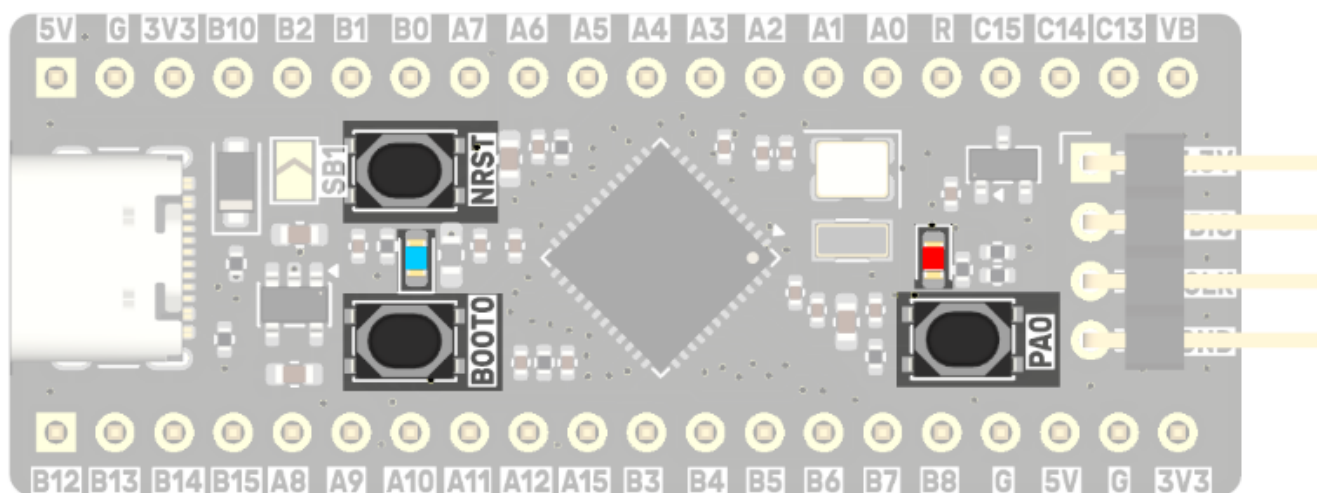
3. Naciśnij i zwolnij przycisk RST (SW3), nadal trzymając B00T.
4. Zwolnij B00T.
5. W STM32CubeProgrammer wybierz interfejs USB i połącz się z urządzeniem DFU.

Po zaprogramowaniu upewnij się, że przycisk BOOT jest zwolniony, i zresetuj płytke. Zamiast opisanej sekwencji można również włączyć zasilanie przy wciśniętym przycisku BOOT.

PA13 i PA14 są zajęte przez SWD podczas programowania i debugowania. PA11 i PA12 są zajęte przez USB podczas programowania DFU lub komunikacji USB.

Dodatkowe elementy - przyciski i diody LED

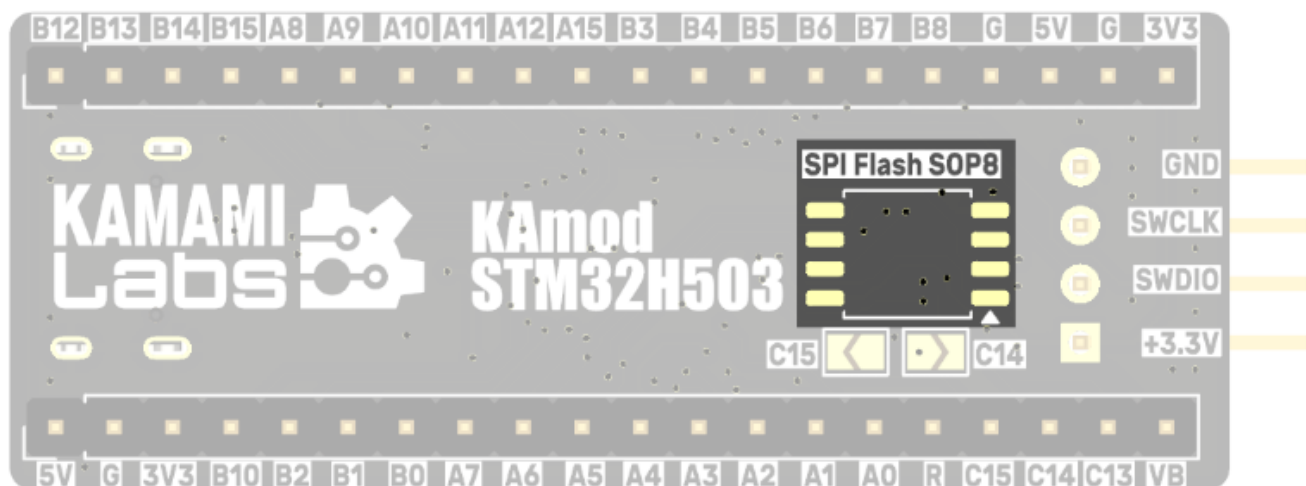
Element	Połączenie	Funkcja
SW1 / PA0 USER KEY	PA0 - aktywny stan niski	Przycisk użytkownika; rezystor szeregowy 330 Ω
SW2 B00T	B00T0 - aktywny stan wysoki	Wybór uruchomienia pamięci systemowej/bootloadera
SW3 RST	NRST - aktywny stan niski	Reset mikrokontrolera
D2 niebieska	PC13 - aktywny stan niski	Dioda sterowana przez program użytkownika
D3 czerwona	+3.3V	Sygnalizacja obecności zasilania 3,3 V



Przycisk USER zwiera wejście PA0 do masy. Oprogramowanie powinno włączyć rezystor podciągający wejście do 3,3 V; tak skonfigurowano dołączony projekt przykładowy. Niebieska dioda świeci po ustawieniu na PC13 stanu niskiego.

Pamięć SPI Flash

Pamięć SPI NOR Flash nie jest montowana fabrycznie. Płytkę ma nieobsadzone pola U3, na których użytkownik może samodzielnie włutować zgodny układ o wybranej pojemności. W25Q16JVSS o pojemności 16 Mbit (2 MB), umieszczony na schemacie, jest przykładem obsługiwanego układu, a nie elementem wyposażenia standardowego.



Montowana pamięć musi spełniać takie warunki:

- zasilanie napięciem 3,3 V,
- obudowa SOIC-8 o szerokości 208 mil, wymiarach około 5,3 × 5,3 mm i rastrze 1,27 mm,
- pinout zgodny z rodziną W25Qxx w trybie SPI,
- obsługa standardowych poleceń SPI NOR używanych przez oprogramowanie.

Sprzęt nie narzuca pojemności pamięci. Dołączony program testowy używa poleceń z adresem 24-bitowym, dlatego bez modyfikacji oprogramowania może bezpośrednio adresować pierwsze 16 MB pamięci. Układy o większej pojemności mogą wymagać obsługi adresowania 4-bajtowego, jeżeli aplikacja ma korzystać z obszaru powyżej tej granicy.

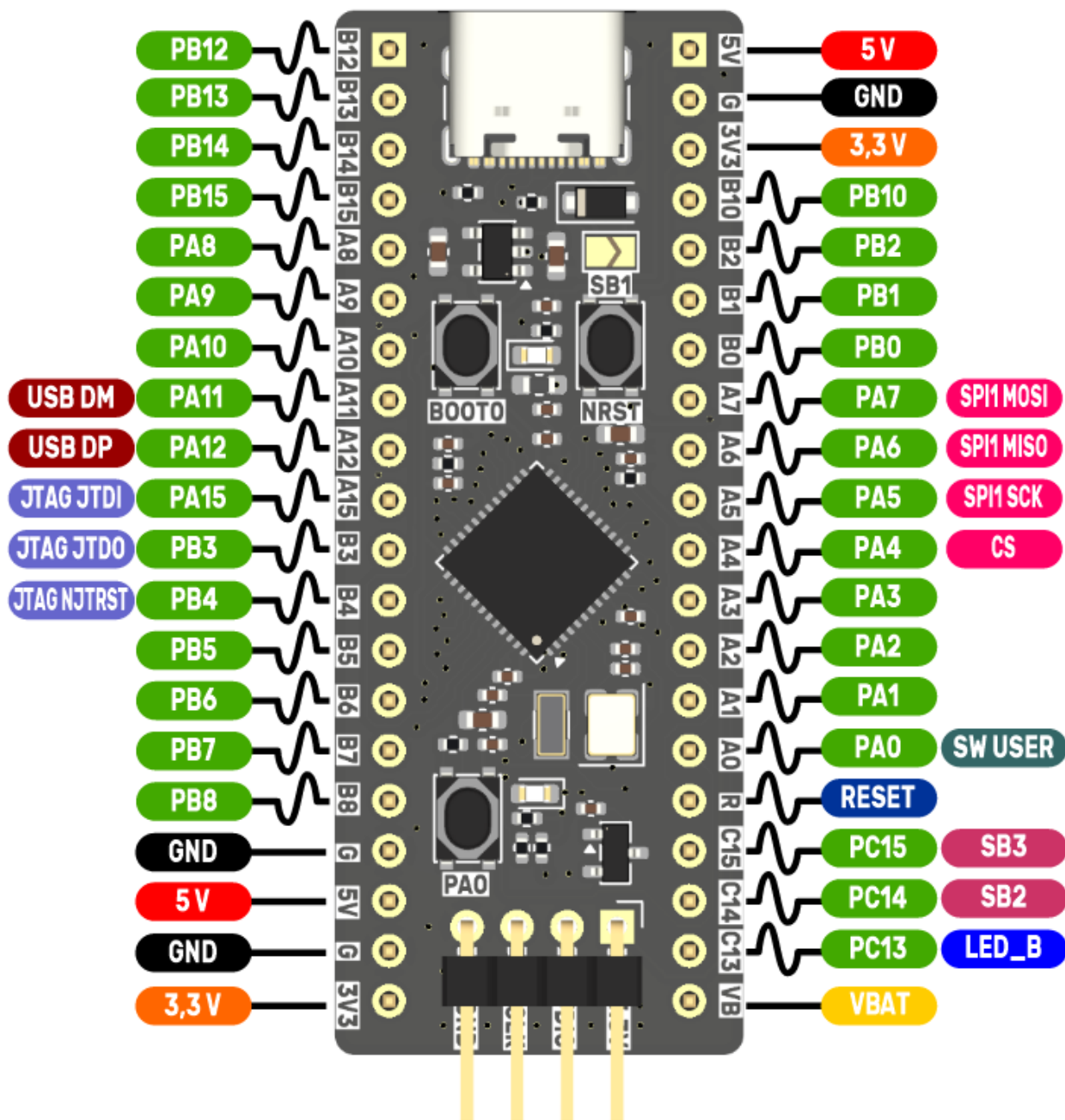
Pola U3 są połączone z SPI1 mikrokontrolera:

Sygnał pamięci	GPIO mikrokontrolera	Złącze P1
CS	PA4	11
SCK	PA5	10
MISO / D0	PA6	9
MOSI / DI	PA7	8

Wyprowadzenia PA4...PA7 są również dostępne na złączu P1 i trwale połączone z polami U3. Po zamontowaniu pamięci użycie tych linii przez dodatkowy układ wymaga zachowania poprawnych stanów sygnału CS i uwzględnienia obciążenia magistrali.

Złącza GPIO

Złącza P1 i P2 mają po 20 styków w rastrze 2,54 mm. Poziomy logiczne mikrokontrolera wynoszą 3,3 V. Tylko wyprowadzenia oznaczone w karcie katalogowej mikrokontrolera jako FT tolerują napięcie 5 V; nie dotyczy to wszystkich pinów ani trybu analogowego.



Złącze P1

Styk	Sygnal	Uwagi
1	+5V	Linia zasilania 5 V
2	GND	Masa
3	+3.3V	Linia zasilania 3,3 V
4	PB10	GPIO
5	PB2	GPIO
6	PB1	GPIO
7	PB0	GPIO
8	PA7	GPIO, SPI1 MOSI pamięci Flash
9	PA6	GPIO, SPI1 MISO pamięci Flash
10	PA5	GPIO, SPI1 SCK pamięci Flash

11	PA4	GPIO, CS pamięci Flash
12	PA3	GPIO
13	PA2	GPIO
14	PA1	GPIO
15	PA0	GPIO, przycisk USER
16	NRST	Reset, aktywny stan niski
17	PC15	GPIO/LSE; połączenie przez SB3
18	PC14	GPIO/LSE; połączenie przez SB2
19	PC13	GPIO, niebieska dioda LED
20	VB	Wejście baterii RTC przez diodę Schottky'ego

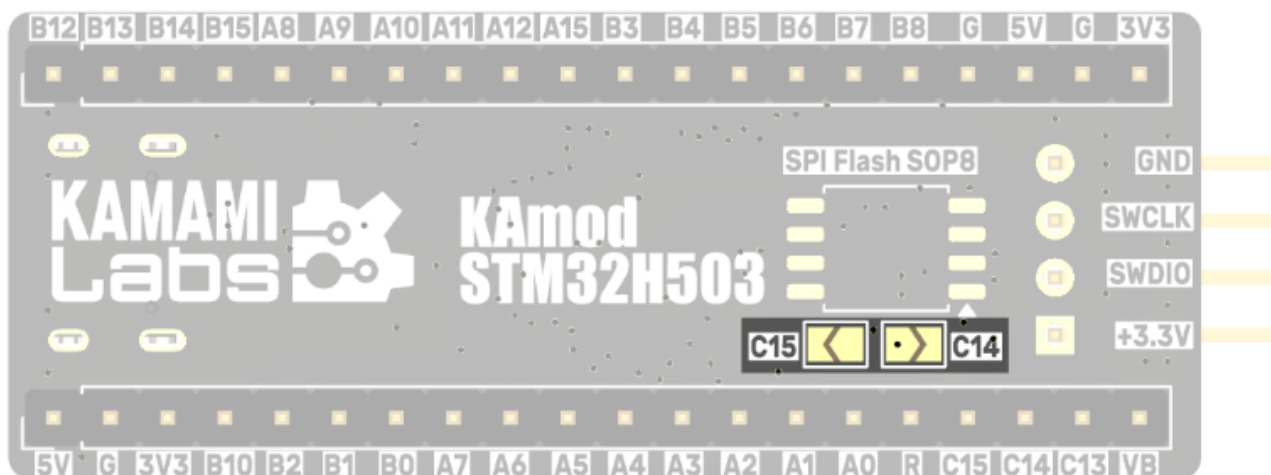
Złącze P2

Styk	Sygnal	Uwagi
1	PB12	GPIO
2	PB13	GPIO
3	PB14	GPIO
4	PB15	GPIO
5	PA8	GPIO
6	PA9	GPIO
7	PA10	GPIO
8	PA11	GPIO, USB DM
9	PA12	GPIO, USB DP
10	PA15	GPIO/JTAG JTDI
11	PB3	GPIO/JTAG JTDO
12	PB4	GPIO/JTAG NJTRST
13	PB5	GPIO
14	PB6	GPIO
15	PB7	GPIO
16	PB8	GPIO
17	GND	Masa
18	+5V	Linia zasilania 5 V
19	GND	Masa
20	+3.3V	Linia zasilania 3,3 V

Pełna lista funkcji alternatywnych oraz informacje o tolerancji napięć znajdują się w karcie katalogowej STM32H503CB.

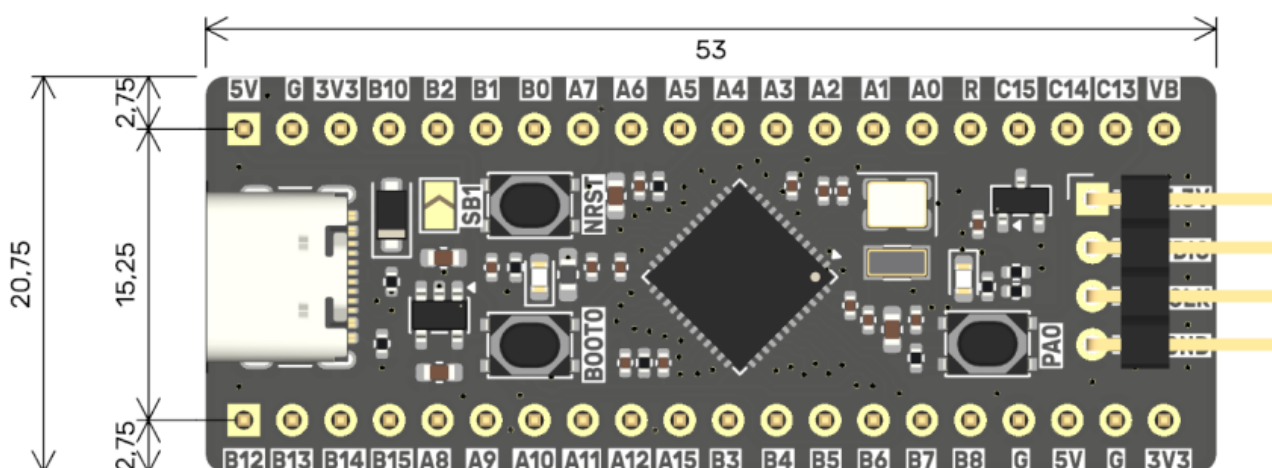
Zworki SB2 (PC14) i SB3 (PC15)

Wyprowadzenia PC14 i PC15 mikrokontrolera są używane przez rezonator 32,768 kHz. Otwarta zworka **SB2** odłącza **PC14** od styku **18** złącza **P1**, a otwarta zworka **SB3** odłącza **PC15** od styku **17**. Zwarcie zworek udostępnia na złączu sygnały oscylatora, ale dodaje do nich pojemność przewodów i dołączonych układów, co może zakłócić pracę rezonatora. W typowym zastosowaniu RTC obie zworki powinny pozostać otwarte.



Wymiary

Wymiary płytki KAmo STM32H503 wynoszą 53 × 20,75 mm.



Program testowy

Katalog KAmo STM32H503 zawiera przykładowy projekt dla STM32CubeIDE. Projekt korzysta z STM32CubeH5 HAL, systemu ThreadX i stosu USBX. Po uruchomieniu płytka zgłasza się jako urządzenie USB CDC ACM, czyli wirtualny port szeregowy. W systemie Linux jest to zwykle /dev/ttyACM0; w systemie Windows urządzenie otrzymuje numer portu COM. Ustawienie prędkości transmisji jest dla USB CDC umowne — można wybrać 115200 bit/s, 8N1.

Komendy należy wpisywać wielkimi literami i kończyć klawiszem Enter. Polecenie HELP wyświetla ich listę.

Komenda	Działanie
HELP	Wyświetla listę komend

RTC GET	Odczytuje datę i czas RTC
RTC SET YYYY-MM-DD HH:MM:SS	Ustawia datę i czas RTC
RTC INIT	Jednorazowo ustawia wartość początkową RTC
LED ON, LED OFF	Steruje niebieską diodą na PC13
BTN GET	Odczytuje stan przycisku USER na PA0
FLASH ID	Odczytuje identyfikator JEDEC opcjonalnej pamięci Flash
FLASH STATUS	Odczytuje rejestr statusu opcjonalnej pamięci Flash
FLASH TEST	Kasuje sektor, zapisuje i weryfikuje 256 bajtów; wymaga zamontowanej pamięci
REBOOT	Resetuje mikrokontroler

Uwaga: Polecenia FLASH . . . wymagają samodzielnie zamontowanej, zgodnej pamięci SPI NOR. FLASH TEST niszczy wcześniejszą zawartość sektora 4 kB rozpoczynającego się pod adresem 0x010000.

Szczegółowy opis protokołu i odpowiedzi

USB CDC CLI – dokumentacja komend

Interfejs CLI jest dostępny jako USB CDC ACM (wirtualny port szeregowy). W systemie Linux urządzenie zwykle pojawia się jako /dev/ttyACM0. Wartość prędkości ustawiona w terminalu jest dla USB CDC umowna; można pozostawić 115200 8N1.

Komendy należy wpisywać **wielkimi literami** i kończyć klawiszem Enter (CR, LF albo CR+LF). Po aktywacji połączenia oraz po każdej obsłużonej komendzie urządzenie wysyła:

READY

Lista komend

Komenda	Działanie
HELP	Wyświetla pionową listę dostępnych komend jako linie OK CMD . . . , zakończoną OK END.
RTC GET	Odczytuje bieżącą datę i czas RTC.
RTC SET YYYY-MM-DD HH:MM:SS	Ustawia datę i czas RTC. Dozwolone lata: 2000 . . 2099.
RTC INIT	Inicjalizuje RTC wartością 2026-01-01 00:00:00, jeśli nie zapisano wcześniej znacznika inicjalizacji.
LED ON	Włącza LED na PC13 (LED jest aktywna stanem niskim).
LED OFF	Wyłącza LED na PC13.
BTN GET	Odczytuje stan wejścia PA0.
FLASH ID	Odczytuje JEDEC ID opcjonalnej pamięci SPI Flash i rozpoznaje producenta.
FLASH STATUS	Odczytuje rejestr statusu SR1 (BUSY i WEL) opcjonalnej pamięci.
FLASH TEST	Kasuje sektor 4 KiB, zapisuje 256 B i weryfikuje dane; wymaga zamontowanej pamięci.
REBOOT	Wysyła potwierdzenie i resetuje mikrokontroler.

Format daty i czasu

Polecenie RTC SET wymaga dokładnie formatu:

RTC SET YYYY-MM-DD HH:MM:SS

Przykład:

RTC SET 2026-07-12 14:30:00

RTC GET

Wymagania:

- rok: 2000..2099,
- miesiąc: 01..12,
- dzień zgodny z miesiącem i rokiem przestępnym,
- godzina: 00..23,
- minuta i sekunda: 00..59,
- wszystkie zera wiodące oraz separatory są obowiązkowe.

Przykładowa odpowiedź:

OK RTC 2026-07-12 14:30:00

READY

Test pamięci SPI Flash

Pamięć SPI NOR nie jest montowana fabrycznie. Polecenia FLASH ID, FLASH STATUS i FLASH TEST działają po samodzielnym wlutowaniu zgodnego układu na pola U3. Pola pamięci są podłączone do SPI1, a sygnał CS znajduje się na PA4. Komenda FLASH TEST używa adresu 0x010000.

Test wykonuje kolejno:

1. odczyt JEDEC ID,
2. skasowanie sektora 4 KiB,
3. sprawdzenie, czy skasowane bajty mają wartość 0xFF,
4. zapis 256-bajtowego wzorca,
5. ponowny odczyt i porównanie danych.

Uwaga: FLASH TEST wymaga zamontowanej pamięci i niszczy wcześniejszą zawartość sektora rozpoczynającego się pod adresem 0x010000.

Format odpowiedzi

- powodzenie zaczyna się od OK,
- błąd zaczyna się od ERR,
- nieznana komenda: ERR CLI unknown_command: <komenda>,
- każda odpowiedź kończy się osobną linią READY,
- linia dłuższa niż 127 znaków: ERR LINE_T00_LONG.

Przykłady:

```
OK
READY
ERR RTC bad_format use: RTC SET YYYY-MM-DD HH:MM:SS
READY
OK BTN PA0=1
READY
OK FLASH SR1=0x00 BUSY=0 WEL=0
READY
```

Sprawdzanie i synchronizacja RTC z komputerem

Skrypt [rtc_sync.py](#) komunikuje się z programem testowym przez USB CDC. Odczytuje czas mikrokontrolera poleceniem RTC GET, porównuje go z lokalnym czasem komputera i - jeśli różnica przekracza ustawiony próg — wysyła polecenie RTC SET. Po synchronizacji ponownie odczytuje RTC i sprawdza wynik.

Skrypt wymaga Pythona 3 i pakietu pyserial:

```
python3 -m pip install pyserial
```

Przykładowe uruchomienie w systemie Linux:

```
python3 tools/rtc_sync.py --port /dev/ttyACM0
```

W systemie Windows jako port należy podać nazwę przydzieloną urządzeniu, na przykład:

```
python tools/rtc_sync.py --port COM5
```

Domyślnie akceptowana różnica wynosi 2 sekundy. Najważniejsze opcje programu:

Opcja	Znaczenie
--port PORT	Port urządzenia; domyślnie /dev/ttyACM0
--baud N	Umowna prędkość portu CDC; domyślnie 115200
--threshold S	Maksymalna akceptowana różnica czasu w sekundach; domyślnie 2.0
--timeout S	Limit czasu odpowiedzi na jedną komendę; domyślnie 3.0
--dry-run	Tylko porównuje czas, bez zmiany RTC
--no-initial-drain	Nie usuwa oczekujących komunikatów po otwarciu portu

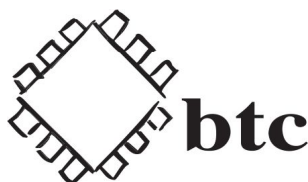
Przykład samego sprawdzenia czasu z progiem 0,5 sekundy:

```
python3 tools/rtc_sync.py --port /dev/ttyACM0 --threshold 0.5 --dry-run
```

Program zwraca kod 0, gdy różnica mieści się w progu, kod 1, gdy po sprawdzeniu nadal jest zbyt duża, oraz kod 2 w przypadku błędu komunikacji, braku pakietu pyserial albo nieprawidłowej odpowiedzi urządzenia.

Linki

- [Podstawowe informacje o układzie STM32H503CB](#)
- [Karta katalogowa STM32H503xx \(DS14053\)](#)
- [Reference manual STM32H503 \(RM0492\)](#)
- [Opis fabrycznych bootloaderów STM32 \(AN2606\)](#)
- [STM32CubeProgrammer](#)
- [Karta katalogowa przykładowej pamięci W25Q16JV](#)
- [Karta katalogowa stabilizatora MIC5504](#)



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.