

UM2237

Instrukcja obsługi

Opis oprogramowania STM32CubeProgrammer

Wprowadzenie

STM32CubeProgrammer (STM32CubeProg) zapewnia wszechstronne narzędzie programowe do programowania urządzeń STM32 w dowolnym środowisku: multi-OS, graficzny interfejs użytkownika lub interfejs wiersza poleceń, obsługa szerokiego wyboru połączeń (JTAG, SWD, USB, UART, SPI, CAN, I2C), z obsługą ręczną lub automatyzacją za pomocą skryptów.

W tym dokumencie szczegółowo opisano wymagania wstępne dotyczące sprzętu i oprogramowania, a także dostępne funkcje oprogramowania STM32CubeProgrammer.

Zawartość

1 Pierwsze kroki.	7
1.1 Wymagania systemowe	7
1.2 Instalowanie STM32CubeProgrammer.	7
1.2.1 Instalacja systemu Linux.	7
1.2.2 Instalacja systemu Windows.	8
1.2.3 instalacja systemu macOS.	8
1.2.4 Sterownik DFU.	8
1.2.5 Sterownik ST-LINK.	9
2 Interfejs użytkownika STM32CubeProgrammer dla MCU.	10
2.1 Główne okno	10
2.1.1 Menu główne	10
2.1.2 Panel dziennika.	11
2.1.3 Pasek postępu	11
2.1.4 Docelowy panel konfiguracji.	12
2.2 Pamięć i edycja plików.	20
2.2.1 Czytanie i wyświetlanie pamięci celu.	20
2.2.2 Czytanie i wyświetlanie pliku.	21
2.3 Programowanie i kasowanie pamięci.	22
2.3.1 Programowanie wewnętrznej pamięci Flash.	22
2.3.2 Programowanie zewnętrznej pamięci Flash.	23
2.3.3 Opracowywanie niestandardowych programów ładujących dla pamięci zewnętrznej.	25
2.4 Bajty opcji.	27
2.5 Tryb automatyczny	28
2.6 Programowanie OTA STM32WB.	32
2.6.1 Konfiguracja klucza USB.	32
2.6.2 Procedura aktualizacji OTA.	33
2.7 W programowaniu aplikacji (IAP).	37
2.8 Sflashuj plik binarny koprocatora używając interfejsu graficznego.	38
2.8.1 Korzystanie z JTAG.	38
2.8.2 Korzystanie z programu ładującego.	41
2.9 Przeglądarka przewodów szeregowych (SWV).	41

2.11	Dane uwierzytelniające SigFox™	44
3	Interfejs wiersza poleceń STM32CubeProgrammer (CLI) dla MCU...	46
3.1	Użycie wiersza poleceń	46
3.2	Ogólne polecenia	48
3.2.1	Połącz polecenie	48
3.2.2	Usuń polecenie	55
3.2.3	Pobierz polecenie	56
3.2.4	Pobierz 32-bitowe polecenie danych	56
3.2.5	Pobierz 64-bitowe polecenie danych	57
3.2.6	Czytaj polecenie	57
3.2.7	Uruchom polecenie	58
3.2.8	Polecenia debugowania	58
3.2.9	Lista poleceń	59
3.2.10	Polecenie QuietMode	60
3.2.11	Polecenie gadatliwość	60
3.2.12	Zaloguj polecenie	61
3.2.13	Polecenie zewnętrznego modułu ładującego	62
3.2.14	Polecenie zewnętrznego programu ładującego z interfejsem programu ładującego	63
3.2.15	Przeczytaj polecenie unprotect	64
3.2.16	Polecenie regresji TZ	64
3.2.17	Opcja bajtów, polecenie	64
3.2.18	Polecenie biblioteki bezpieczeństwa	64
3.2.19	Bezpieczne programowanie poleceń specyficznych dla SFL	68
3.2.20	Bezpieczne programowanie poleceń specyficznych dla SFLx	68
3.2.21	Polecenia związane z HSM	70
3.2.22	Polecenia specyficzne dla STM32WB	71
3.2.23	Polecenie przeglądarki przewodów szeregowych (SWV)	73
3.2.24	Specyficzne polecenia dla STM32WL	74
3.2.25	Polecenia poświadczeń SigFox	76
4	Interfejs użytkownika STM32CubeProgrammer dla MPU	78
4.1	Główne okno	78
4.2	Programowanie okien	79
5	STM32CubeProgrammer CLI dla MPU	80
5.1	Dostępne polecenia dla STM32MP1	80

Zawartość

UM2237

4/ [93](#)

UM2237 wer. 14

5.1.1Połącz polecenie.....	80
5.1.2Polecenie GetPhase.....	81
5.1.3Pobierz polecenie.....	81
5.1.4Usługa flashowania.....	82
5.1.5Uruchom polecenie.....	82
5.1.6Przeczytaj polecenie partycji.....	83
5.1.7Lista, polecenie.....	83
5.1.8Polecenie QuietMode.....	84
5.1.9Polecenie gadatliwość.....	84
5.1.10Zaloguj polecenie.....	84
5.1.11Programowanie OTP.....	86
5.1.12Programowanie komendy SAFMEM.....	86
5.1.13Odłącz polecenie.....	87
5.1.14Polecenie GetCertif.....	87
5.1.15Napisz polecenie blob.....	87
5.1.16Polecenie wyświetlania.....	87
5.2Bezpieczne programowanie poleceń specyficznych dla SSP.....	87
6STM32CubeProgrammer C ++ API.....	90
7Historia zmian.....	91

Spis rysunków

Rysunek 1.Usuwanie starego oprogramowania sterownika.	8
Rysunek 2.Urządzenie STM32 DFU ze sterownikiem DfuSe.	9
Rycina 3.Urządzenie STM32 DFU ze sterownikiem STM32CubeProgrammer.	9
Rysunek 4.Główne okno programu STM32CubeProgrammer.	10
Rycina 5.Rozwinięte menu główne.	11
Rysunek 6.Panel konfiguracyjny STLINK.	12
Rysunek 7.Panel konfiguracyjny UART.	14
Cyfra 8.Panel konfiguracyjnyUSB.	15
Ryc.9.Docelowy panel informacyjny.	16
Rysunek 10. Panel konfiguracji SPI.	17
Rysunek 11. Panel konfiguracjiCAN.	18
Rysunek 12. Panel konfiguracji I2C.	19
Rysunek 13. Pamięć i edycja plików: Zakładka Pamięć urządzenia.	20
Rysunek 14. Pamięć i edycja plików: menu kontekstowe.	21
Rysunek 15. Pamięć i edycja plików: Wyświetlanie plików.	21
Rysunek 16. Programowanie i kasowanie pamięci Flash (pamięć wewnętrzna).	22
Rysunek 17. Programowanie pamięci Flash (pamięć zewnętrzna).	24
Rysunek 18. Kasowanie pamięci Flash (pamięć zewnętrzna).	24
Rysunek 19. Panel bajtów opcji.	27
Rysunek 20. Tryb automatyczny w oknie Kasowanie i programowanie.	28
Rysunek 21. Tryb automatyczny Śledzenie dziennika.	29
Rysunek 22. Algorytm.	31
Rysunek 23. Aktualizacja OTA firmware'u STM32WB poprzez połączenie BLE.	32
Rysunek 24. Programowanie klucza USB (tryb USB DFU) za pomocą STM32CubeProgrammer.	33
Rysunek 25. Okno aktualizacji OTA.	34
Rysunek 26. Wyszukiwanie urządzeń BLE.	34
Rysunek 27. Wybór urządzenia OTA.	35
Rysunek 28. Wybór pliku obrazu.	36
Rysunek 29. Błyskanie OTA.	36
Rysunek 30. Zakończono flashowanie OTA.	37
Rysunek 31. Programator STM32Cube w trybie IAP.	38
Rysunek 32. Połączenie STM32CubeProgrammer API SWD.	39
Rysunek 33. Kroki aktualizacji oprogramowania sprzętowego.	39
Rysunek 34. Wyskakujące okienko potwierdzające pomyślne usunięcie oprogramowania sprzętowego..	40
Rysunek 35. Wyskakujące okienko potwierdzające pomyślną aktualizację oprogramowania.	40
Rysunek 36. Kroki aktualizacji oprogramowania za pomocą interfejsu programu ładującego.	41
Rysunek 37. Okno SWV.	42
Rysunek 38. Podłącz przez panel USBDFU.	43
Rysunek 39. Okno główne po połączeniu.	44
Rysunek 40. PoświadczeniaSigFox.	45
Rysunek 41. STM32CubeProgrammer: dostępne polecenia.	47
Rysunek 42. Połącz operację za pomocą RS232.	50
Rysunek 43. Podłączanie za pomocąUSB.	51
Rysunek 44. Podłączanie za pomocą opcji USB DFU.	52
Rysunek 45. Połącz operację przy użyciu portu debugowania SWD.	52
Rysunek 46. Łączenie za pomocą portu SPI.	54
Rysunek 47. Podłączanie za pomocą portu CAN.	54
Rysunek 48. Połącz operację za pomocą portu I2C.	55

Spis rycin

6/ 93

<u>Rysunek 49. Operacja pobierania.</u>	<u>56</u>
<u>Rysunek 50. 32-bitowa operacja odczytu.</u>	<u>58</u>
<u>Rysunek 51. Lista dostępnych portów szeregowych.</u>	<u>60</u>
<u>Rysunek 52. Polecenie gadatliwości.</u>	<u>61</u>
<u>Rysunek 53. Polecenie dziennika.</u>	<u>61</u>
<u>Rysunek 54. Zawartość pliku dziennika.</u>	<u>62</u>
<u>Rysunek 55. Polecenie Safety lib.</u>	<u>65</u>
<u>Rysunek 56. Mapowanie pamięci flash.</u>	<u>66</u>
<u>Rysunek 57. Przykład odwzorowania pamięci flash.</u>	<u>67</u>
<u>Rysunek 58. Polecenie SWV.</u>	<u>73</u>
<u>Rysunek 59. Wyłącz zabezpieczenia.</u>	<u>75</u>
<u>Rysunek 60. Skonfiguruj bajty opcji do ich wartości domyślnych.</u>	<u>75</u>
<u>Rysunek 61. Przykład polecenia -ssigfoxc.</u>	<u>76</u>
<u>Rysunek 62. Przykład (1) polecenia -wsigfoxc.</u>	<u>77</u>
<u>Rysunek 63. Przykład (2) polecenia -wsigfoxc.</u>	<u>77</u>
<u>Rysunek 64. Główne okno STM32CubeProgrammer.</u>	<u>78</u>
<u>Rysunek 65. Okno programowania TSV.</u>	<u>79</u>
<u>Rysunek 66. Łączenie za pomocą RS232.</u>	<u>81</u>
<u>Rysunek 67. Operacja pobierania.</u>	<u>82</u>
<u>Rysunek 68. Format pliku TSV.</u>	<u>82</u>
<u>Rysunek 69. Zawartość pliku dziennika.</u>	<u>85</u>
<u>Rysunek 70. Pomyślnie zainstalowano SSP.</u>	<u>89</u>

1 Pierwsze kroki

W tej sekcji opisano wymagania i procedury dotyczące instalowania narzędzia programowego STM32CubeProgrammer.

STM32CubeProgrammer obsługuje 32-bitowe MCU STM32 oparte na Arm® (a) Cortex® -M procesory i 32-bitowe MPU STM32 oparte na procesorach Arm® Cortex® -A.

1.1 Wymagania systemowe

Obsługiwane systemy operacyjne i architektury:

- Linux® 64-bitowy
- Windows® 7/8/10 32-bitowy i 64-bitowy
- macOS® (minimalna wersja OS X® Yosemite)

Od wersji 2.6.0 nie ma potrzeby instalowania żadnego środowiska wykonawczego Java™ SE. STM32CubeProgrammer działa z dołączonym JRE dostępnym w pobranym pakiecie, a nie z zainstalowanym na twoim komputerze.

Uwaga:

Dołączone środowisko JRE to Liberica 8.0.265

Minimalna obsługiwana rozdzielczość ekranu to 1024x768.

1.2 Instalowanie STM32CubeProgrammer

W tej sekcji opisano wymagania i procedurę korzystania z oprogramowania STM32CubeProgrammer.

Instalator oferuje również opcjonalną instalację narzędzia „Twórca zaufanych pakietów STM32”, używanego do tworzenia bezpiecznych plików oprogramowania układowego w celu bezpiecznej instalacji i aktualizacji oprogramowania układowego. Aby uzyskać więcej informacji, zapoznaj się z opisem oprogramowania narzędzia STM32 Trusted Package Creator (UM2238), dostępnym na stronie www.st.com.

1.2.1 Instalacja systemu Linux

Jeśli używasz portu USB do połączenia z urządzeniem STM32, zainstaluj pakiet libusb1.0 przez wpisując następujące polecenie:

```
sudo apt-get install libusb-1.0.0-dev
```

Aby użyć sondy ST-LINK lub USB DFU do połączenia z celem, skopiuj pliki reguł znajdujące się w Folder sterownika / reguł w `/etc/udev/rules.d/` w systemie Ubuntu („*sudo cp *. */Etc/udev/rules.d*”).

Uwaga:

Do uruchomienia STM32CubeProgrammer wymagana jest wersja libusb1.0.12 lub wyższa.

Aby zainstalować narzędzie STM32CubeProgrammer, pobierz i rozpakuj pakiet zip w pliku Maszyna Linux z numeru części STM32CubeProg-Linux na stronie internetowej i wykonaj *SetupSTM32CubeProgrammer-vx.yzlinux*, który poprowadzi Cię przez proces instalacji proces.

1.2.2 Instalacja systemu Windows

Aby zainstalować narzędzie STM32CubeProgrammer, pobierz i wyodrębnij pakiet zip z STM32CubeProg-Win-32bits lub STM32CubeProg-Win-64bits odpowiednio dla Windows 32 bity i Windows 64 bity, a następnie uruchom *SetupSTM32CubeProgrammer-vx.yzexe*, który przeprowadzi Cię przez proces instalacji.

1.2.3 instalacja systemu macOS

Aby zainstalować narzędzie STM32CubeProgrammer, pobierz i wyodrębnij pakiet zip z Numer części STM32CubeProg-Mac na stronie internetowej i wykonaj *SetupSTM32CubeProgrammer-vx.yzapp*, który poprowadzi Cię przez proces instalacji proces.

Uwaga:

Jeśli instalacja się nie powiedzie, uruchom ją w trybie CLI za pomocą polecenia

./SetupSTM32CubeProgrammer-

xyzapp / Contents / MacOS / SetupSTM32CubeProgrammer-x_y_z_macos.

1.2.4 Sterownik DFU

Jeśli używasz urządzenia STM32 w trybie USB DFU, zainstaluj STM32CubeProgrammer's Sterownik DFU, uruchamiając plik „*STM32 Bootloader.bat*”. Ten sterownik jest dostarczany z pakiet wersji, można go znaleźć w folderze sterownika DFU.

Jeśli masz zainstalowany sterownik DFUSE na swoim komputerze, najpierw odinstaluj go, a następnie uruchom ponownie maszynę i uruchom wspomniany wcześniej plik „.bat”. Musisz zaznaczyć opcję „Usuń sterownik oprogramowanie dla tego urządzenia, aby uniknąć ponownej instalacji starego sterownika, gdy później zostanie zainstalowana płyta podłączony.

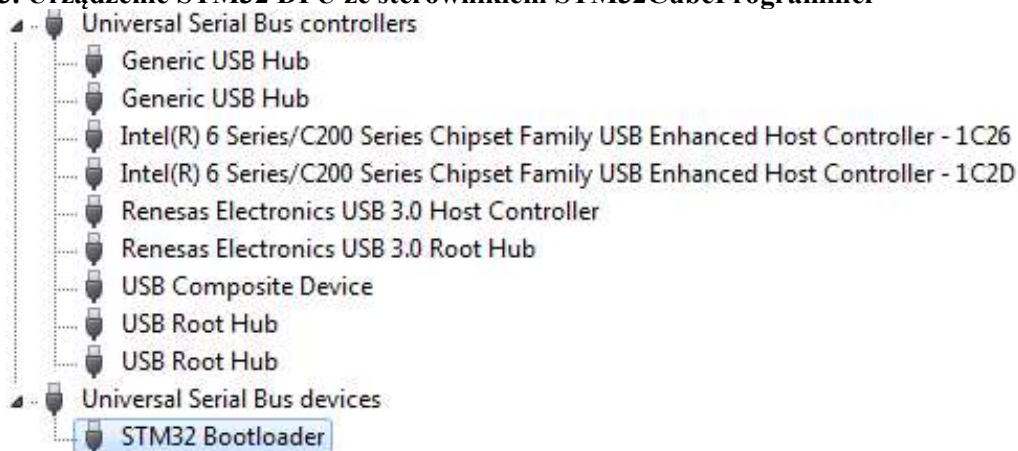
Rysunek 1. Usuwanie starego oprogramowania sterownika



Rysunek 2. Urządzenie STM32 DFU ze sterownikiem DfuSe



Rysunek 3. Urządzenie STM32 DFU ze sterownikiem STM32CubeProgrammer



Uwaga:

Podczas korzystania z interfejsu USB DFU lub interfejsu STLink na komputerze z systemem Windows 7 upewnij się, że wszystkie sterowniki kontrolera USB 3.0 są aktualne. Starsze wersje sterowników mogą zawierać błędy uniemożliwić dostęp lub spowodować problemy z połączeniem z urządzeniami USB.

1.2.5

Sterownik ST-LINK

Aby połączyć się z urządzeniem STM32 przez interfejs debugowania za pomocą ST-LINK / V2, ST-LINKV2-1

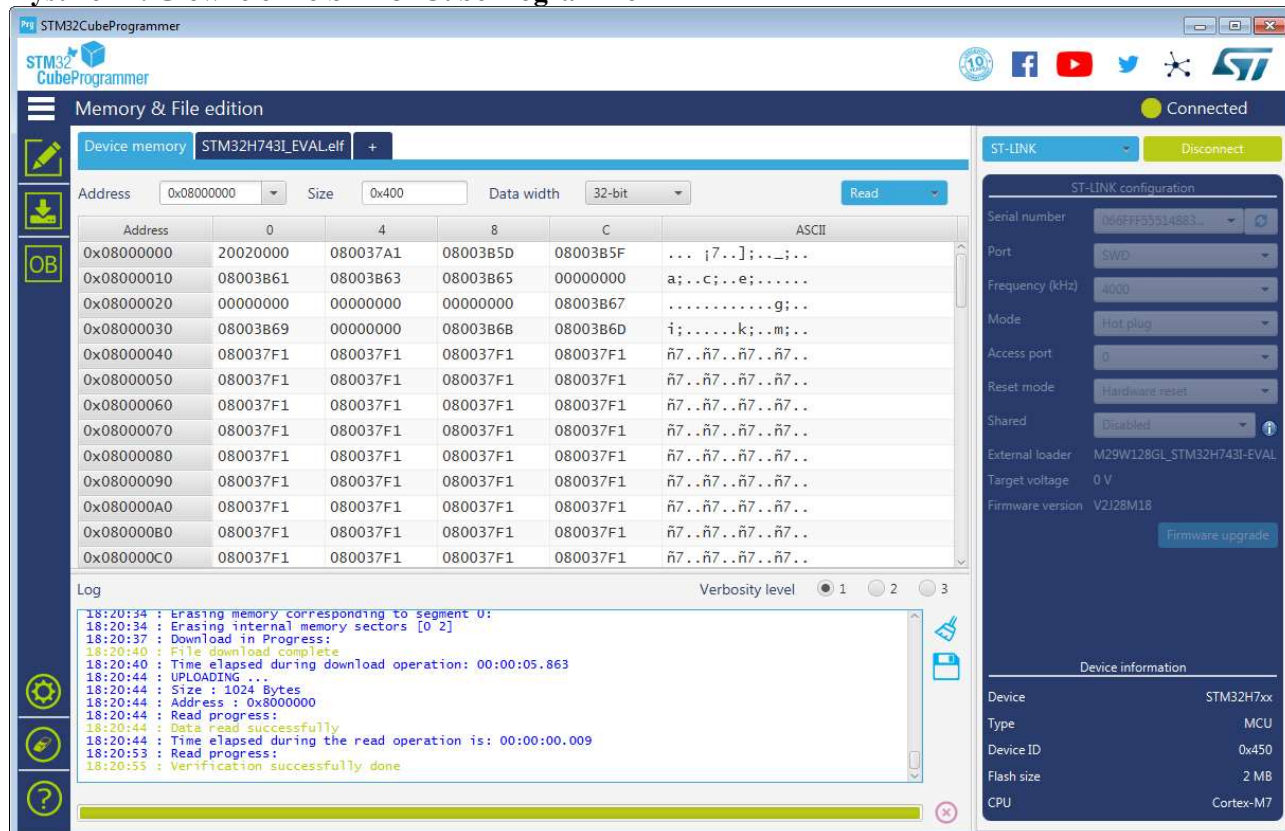
lub ST-LINK-V3, zainstaluj sterownik ST-LINK, uruchamiając plik „stlink_winusb_install.bat”. To sterownik jest dostarczany z pakietem wydania, można go znaleźć w pliku

Folder „Driver / stsw-link009_v3”.

2 Interfejs użytkownika STM32CubeProgrammer dla MCU

2.1 Główne okno

Rysunek 4. Główne okno STM32CubeProgrammer



Okno główne składa się z części opisanych w kolejnych sekcjach.

2.1.1 Menu główne

Menu główne umożliwia użytkownikowi przełączanie się między trzema głównymi panelami narzędzi edycji pamięci i plików, programowania i kasowania pamięci oraz opcji byes.

Klikając menu Hamburger (przycisk z trzema liniami) w lewym górnym rogu, menu główne rozwija się i wyświetla opis tekstowy pokazany na rysunku 5.

Rysunek 5. Rozwinięte menu główne

2.1.2 Panel dziennika

Wyświetla błędy, ostrzeżenia i zdarzenia informacyjne związane z operacjami wykonywanymi przez narzędzie. Szczegółowość wyświetlanych komunikatów można doprecyzować za pomocą przycisków opcji szczegółowości nad strefą tekstu dziennika. Minimalny poziom szczegółowości wynosi 1, a maksymalny 3, w którym rejestrowane są wszystkie transakcje za pośrednictwem wybranego interfejsu. Wszystkie wyświetlane komunikaty są oznaczone znacznikiem czasu w formacie „gg:mm:ss:ms”, gdzie „gg” oznacza godziny, „mm” minuty, „ss” sekundy i „ms” milisekundy (trzy cyfry).

Po prawej stronie panelu dziennika znajdują się dwa przyciski, pierwszy do czyszczenia dziennika, a drugi do zapisywania go w pliku dziennika.

2.1.3 Pasek postępu

Pasek postępu wizualizuje postęp dowolnej operacji lub transakcji wykonanej przez narzędzie (Czytaj, pisz, kasuj...). Możesz przerwać każdą trwającą operację, klikając przycisk „Zatrzymaj” przed paskiem postępu.

2.1.4 Docelowy panel konfiguracji

To pierwszy panel, na który należy spojrzeć przed połączeniem się z celem. Pozwala użytkownikowi wybrać interfejs docelowy; albo interfejs debugowania za pomocą sondy debugującej ST-LINK lub interfejs bootloadera przez UART, USB, SPI, CAN lub I2C.

Przycisk odświeżania umożliwia sprawdzenie dostępnych interfejsów podłączonych do komputera. Po naciśnięciu tego przycisku, gdy wybrany jest interfejs ST-LINK, narzędzie sprawdza podłączone sondy ST-LINK i wyświetla je w polu kombi Numery seryjne. Jeśli wybrany jest interfejs UART, sprawdza dostępne porty COM komputera i wyświetla je w polu kombo Port. Jeśli wybrany jest interfejs USB, sprawdza urządzenia USB w trybie DFU podłączone do komputera i wyświetla je również w polu kombi Port. Każdy interfejs ma własne ustawienia, które należy ustawić przed połączeniem.

Ustawienia ST-LINK

Rysunek 6. Panel konfiguracyjny ST-LINK

ST-LINK configuration

Serial number: 066FFF555148837167025130

Port: SWD

Frequency (kHz): 4000

Mode: Hot plug

Access port: 0

Reset mode: Hardware reset

Shared: Disabled

External loader: MT25TL01G_STM32H743I-EVAL

Target voltage: 3.20 V

Firmware version: V2J28M18

Firmware upgrade

Device information

Device: -

Type: -

Device ID: -

Flash size: -

CPU: -

- **Numer seryjny** : To pole zawiera numery seryjne wszystkich podłączonych ST-LINK sondy. Użytkownik może wybrać jeden z nich na podstawie jego numeru seryjnego.
- **Port** : Sonda ST-LINK obsługuje dwa protokoły debugowania, JTAG i SWD.

Uwaga:

JTAG nie jest dostępny we wszystkich wbudowanych ST-LINK w płytach STM32 Nucleo lub Discovery.

- **Częstotliwość** : częstotliwość zegara JTAG lub SWD
- **Port dostępu** : wybór portu dostępu do połączenia. Większość urządzeń STM32 ma tylko jeden port dostępu, którym jest Access port 0.

• **Tryb** :

- **Normalny** : W trybie połączenia „Normalny” cel jest resetowany, a następnie zatrzymywany. Typ resetowania wybiera się za pomocą opcji „Reset Mode”.

- **Połącz w ramach resetowania** : Ten tryb umożliwia połączenie z celem za pomocą resetowania vector catch przed wykonaniem jakichkolwiek instrukcji. Jest to przydatne w wielu przypadkach przykład, gdy cel zawiera kod, który wyłącza piny JTAG / SWD.

- **Hot Plug** : umożliwia połączenie z celem bez zatrzymywania lub resetowania. Jest to przydatne do aktualizacji adresów pamięci RAM lub rejestrów IP podczas działania aplikacji

🕒 **Tryb resetowania:**

- **Reset systemu oprogramowania**: resetuje wszystkie komponenty STM32 z wyjątkiem debugowania poprzez przerwanie aplikacji Cortex-M i resetowanie rejestru sterującego (AIRCR).

- **Reset sprzętowy**: resetuje urządzenie STM32 przez pin nRST. Pin RESET złącza JTAG (pin 15) musi być połączony z pinem resetowania urządzenia.

- **Reset rdzenia**: resetuje tylko rdzeń Cortex-M za pośrednictwem AIRCR.

• **Shared** :Włącza tryb współdzielony umożliwiający podłączenie dwóch lub więcej instancji STM32CubeProgrammer lub innego debuggera do tej samej sondy ST-LINK.

• **Moduł ładujący zewnętrzny** : wyświetla nazwę programu ładującego pamięć zewnętrzną wybranego w Panel „Zewnętrzne ładowarki” dostępny z menu głównego (menu Hamburger).

• **Napięcie docelowe** : Tutaj mierzone i wyświetlane jest napięcie docelowe.

• **Wersja oprogramowania** : Wyświetla **wersję oprogramowania sprzętowego** ST-LINK. Aktualizacja oprogramowania układowego -przycisk umożliwia aktualizację oprogramowania ST-LINK.

Rysunek 7. Panel konfiguracji UART

Not connected

UART Connect

UART configuration

Port COM190

Baudrate 115200

Parity Even

Data bits 8

Stop bits 1.0

Flow control Off

Device information

Device -

Type -

Device ID -

Flash size -

CPU -

- **Port** : wybiera port com, do którego jest podłączony docelowy STM32. Użyj przycisku odświeżania, aby ponownie sprawdzić dostępny port COM na komputerze.

Uwaga:

STM32 musi uruchomić się w trybie bootloadera przy użyciu pinów rozruchowych i / lub bitów opcji.

Sprawdź “STM32 microcontroller system memory boot mode” (AN2606), dostępny na www.st.com, aby uzyskać więcej informacji na temat bootloadera STM32.

- **Baudrate** : Wybiera prędkość transmisji UART.
- **Parzystość** : Wybiera parzystość (parzysta, nieparzysta, brak). Musi być równe dla wszystkich urządzeń STM32.
- **Bitów danych** : zawsze muszą być 8. Tylko dane 8-bitowe są obsługiwane przez STM32.
- **Bitów stopu** : Zawsze muszą być 1. Tylko 1-bitowy bit stopu jest obsługiwany przez STM32.
- **Kontrola przepływu** : zawsze musi być wyłączona.

Ustawienia USB

Rysunek 8. Panel konfiguracyjny USB



- **Port** : Wybiera urządzenia USB w trybie DFU podłączone do komputera. Możesz użyć przycisk odświeżania, aby ponownie sprawdzić dostępne urządzenia.

Uwaga:

STM32 musi uruchomić się w trybie bootloadera przy użyciu pinów rozruchowych i / lub bitów opcji.

Sprawdź

AN2606, dostępny na www.st.com , aby uzyskać więcej informacji na temat bootloadera STM32.

Po skonfigurowaniu prawidłowych ustawień interfejsu kliknij przycisk „Połącz”, aby połączyć się z interfejsem docelowym. Jeśli połączenie się powiedzie, jest to pokazane na wskaźniku nad przyciskiem, który zmienia kolor na zielony.

Po podłączeniu informacje o celu zostaną wyświetlone w sekcji informacji o urządzeniu poniżej sekcja ustawień, która jest następnie wyłączona, jak w [Rys. 9](#) .

Rysunek 9. Panel informacji o celu

ST-LINK

Connect

ST-LINK configuration

Serial number

52FF7006526654...

Port

SWD

Frequency (kHz)

4000

Mode

Normal

Access port

0

Reset mode

Hardware reset

Shared

Disabled

External loader

M29W128GL_STM32H743I-EVAL

Target voltage

2.44 V

Firmware version

V2J32S7

Firmware upgrade

Device information

Device

-

Type

-

Device ID

-

Flash size

-

CPU

-

Ustawienia SPI

Rysunek 10. Panel konfiguracji SPI

The screenshot shows the ST-Link configuration interface. At the top, it says 'Not connected' with a red dot. Below that is a dropdown menu set to 'ST-LINK' and a green 'Connect' button. The main section is titled 'ST-LINK configuration' and contains several settings: 'Serial number' (0034002130375... with a refresh icon), 'Port' (SPI), 'Baudrate (kHz)' (375), 'nss' (Soft), 'nsspulse' (No Pulse), 'delay' (No delay), 'Direction' (Full duplex), 'External loader' (-), 'Target voltage' (3.26 V), and 'Firmware version' (V3J1M1B1S1) with a 'Firmware upgrade' button. At the bottom, there is a 'Device information' section with fields for Device, Type, Device ID, Flash size, and CPU, all showing dashes.

ST-LINK configuration	
Serial number	0034002130375...
Port	SPI
Baudrate (kHz)	375
nss	Soft
nsspulse	No Pulse
delay	No delay
Direction	Full duplex
External loader	-
Target voltage	3.26 V
Firmware version	V3J1M1B1S1
Firmware upgrade	

Device information	
Device	-
Type	-
Device ID	-
Flash size	-
CPU	-

- **Numer seryjny** : To pole zawiera numery seryjne wszystkich podłączonych ST-LINK-V3 sondy w przypadku użycia bootloadera SPI.
- **Port** : Wybiera urządzenia SPI podłączone do komputera. Możesz użyć przycisku odświeżania, aby ponownie sprawdzić dostępne urządzenia.
- **Baudrate** : Wybiera prędkość transmisji SPI.
- **nss** : Slave Wybierz oprogramowanie lub sprzęt.
- **nsspulse** : sygnał Slave Selection może działać w trybie impulsowym, w którym master generuje impulsy na sygnale wyjściowym nss między ramkami danych przez czas jednego SPI okres zegarowy, gdy istnieje ciągły okres przesyłania.
- **Opóźnienie** : służy do wstawiania opóźnienia kilku mikrosekund między danymi.
- **Kierunek** : zawsze musi być w trybie pełnego duplexu, używane są obie linie danych i dane synchroniczne płynię w obu kierunkach.

Ustawienia CAN

Rysunek 11. Panel konfiguracyjny CAN

The image shows a software interface for configuring an ST-LINK. At the top, a status bar indicates 'Not connected' with a red dot. Below this, there are two buttons: 'ST-LINK' and 'Connect'. The main section is titled 'ST-LINK configuration' and contains several settings:

- Serial number:** A dropdown menu showing '0034002130375...' with a refresh icon.
- Port:** A dropdown menu set to 'CAN'.
- Baudrate (kHz):** A dropdown menu set to '125'.
- Assigned fifo:** A dropdown menu set to 'FIFO 0'.
- Filter mode:** A dropdown menu set to 'Mask'.
- Filter scale:** A dropdown menu set to '32 bits'.
- Filter bank:** A numeric input field set to '0' with up and down arrows.
- External loader:** A text field with a hyphen '-'.
- Target voltage:** A text field set to '0.00 V'.
- Firmware version:** A text field set to 'V3J1M1B1S1'.

Below these settings is a blue button labeled 'Firmware upgrade'. At the bottom, there is a section titled 'Device information' with a table of device details:

Device information	
Device	-
Type	-
Device ID	-
Flash size	-
CPU	-

- **Numer seryjny** : To pole zawiera numery seryjne wszystkich podłączonych ST-LINK-V3 sondy w przypadku użycia bootloadera CAN.
- **Port** : Wybiera urządzenia CAN podłączone do komputera. Możesz użyć przycisku odświeżania, aby ponownie sprawdzić dostępne urządzenia.
- **Baudrate** : Wybiera prędkość transmisji CAN.
- **Assigned FIFO** : Wybiera pamięć odbioru FIFO do przechowywania wiadomości przychodzących.
- **Tryb filtra** : wybiera typ filtra, MASKE lub LISTĘ.
- **Filter scale** : wybiera szerokość banku filtrów, 16 lub 32 bity.
- **Bank filtrów** : wartości od 0 do 13, aby wybrać numer banku filtrów.

Ustawienia I2C

Rysunek 12. Panel konfiguracji I2C

The screenshot shows the ST-LINK configuration interface. At the top, it says 'Not connected' with a red dot. Below that is a 'Connect' button. The main section is titled 'ST-LINK configuration' and contains several fields: 'Serial number' (0034002130375...), 'Port' (I2C), 'Baudrate (kHz)' (400), 'Address' (0x4A), 'Speed mode' (Fast), 'Rise time (ns)' (0), 'Fall time (ns)' (0), 'External loader' (-), 'Target voltage' (3.26 V), and 'Firmware version' (V3J1M1B1S1). There is a 'Firmware upgrade' button. At the bottom, there is a 'Device information' section with fields for 'Device', 'Type', 'Device ID', 'Flash size', and 'CPU', all showing dashes.

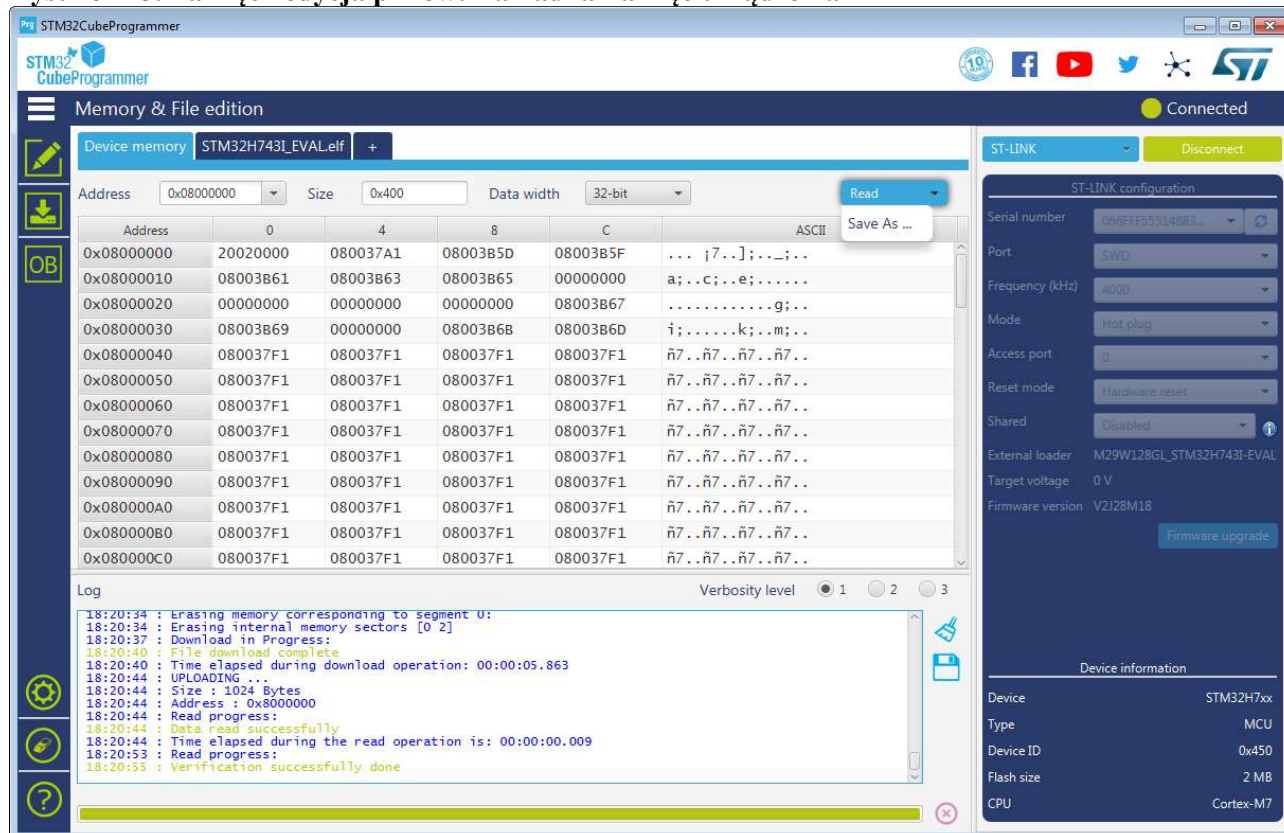
- **Numer seryjny** : To pole zawiera numery seryjne wszystkich podłączonych ST-LINK-V3 sondy w przypadku użycia bootloadera I2C.
- **Port** : Wybiera urządzenia I2C podłączone do komputera. Możesz użyć przycisku odświeżania, aby ponownie sprawdzić dostępne urządzenia.
- **Baudrate** : Wybiera prędkość transmisji I2C.
- **Adres** : dodaje adres podrzędnego programu ładującego w formacie szesnastkowym.
- **Tryb prędkości** : Wybiera tryb szybkości transmisji Standardowa lub Szybka.
- **Czas narastania** : wybiera wartości zgodnie z trybem prędkości, 0-1000 (STANDARD), 0-300 (SZYBK).
- **Czas opadania** : wybiera wartości zgodnie z trybem prędkości, 0-300 (STANDARD), 0-300 (SZYBK).

2.2 Pamięć i edycja plików

Panel Pamięć i edycja plików umożliwia użytkownikowi odczyt i wyświetlenie pamięci docelowej oraz zawartość pliku.

2.2.1 Czytanie i wyświetlanie pamięci celu

Rysunek 13. Pamięć i edycja plików: Zakładka Pamięć urządzenia

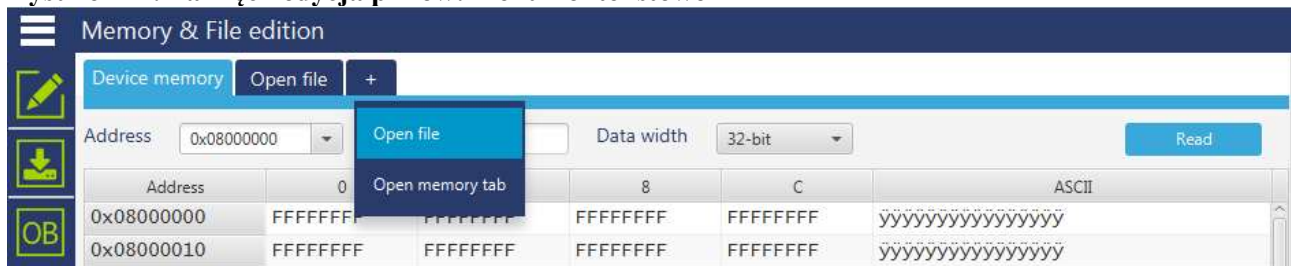


Po połączeniu docelowym możesz odczytać pamięć docelową STM32 za pomocą tego panelu. Aby to zrobić, określ adres i rozmiar danych do odczytania, a następnie kliknij przycisk Czytaj w lewym górnym rogu. Dane mogą być wyświetlane w różnych formatach (8-, 16- i 32-bitowych) za pomocą pola combo „Szerokość danych”.

Zawartość pamięci urządzenia można również zapisać w pliku .bin, .hex lub .srec za pomocą menu „Zapisz jako...” z menu kontekstowego zakładki lub przycisku akcji.

Możesz otworzyć wiele kart pamięci urządzenia, aby wyświetlić różne lokalizacje pamięci docelowej. W tym celu wystarczy kliknąć zakładkę „+”, aby wyświetlić menu kontekstowe, które pozwala dodać nową zakładkę „Pamięć urządzenia” lub otworzyć plik i wyświetlić go w zakładce „Plik”:

Rysunek 14. Pamięć i edycja plików: menu kontekstowe



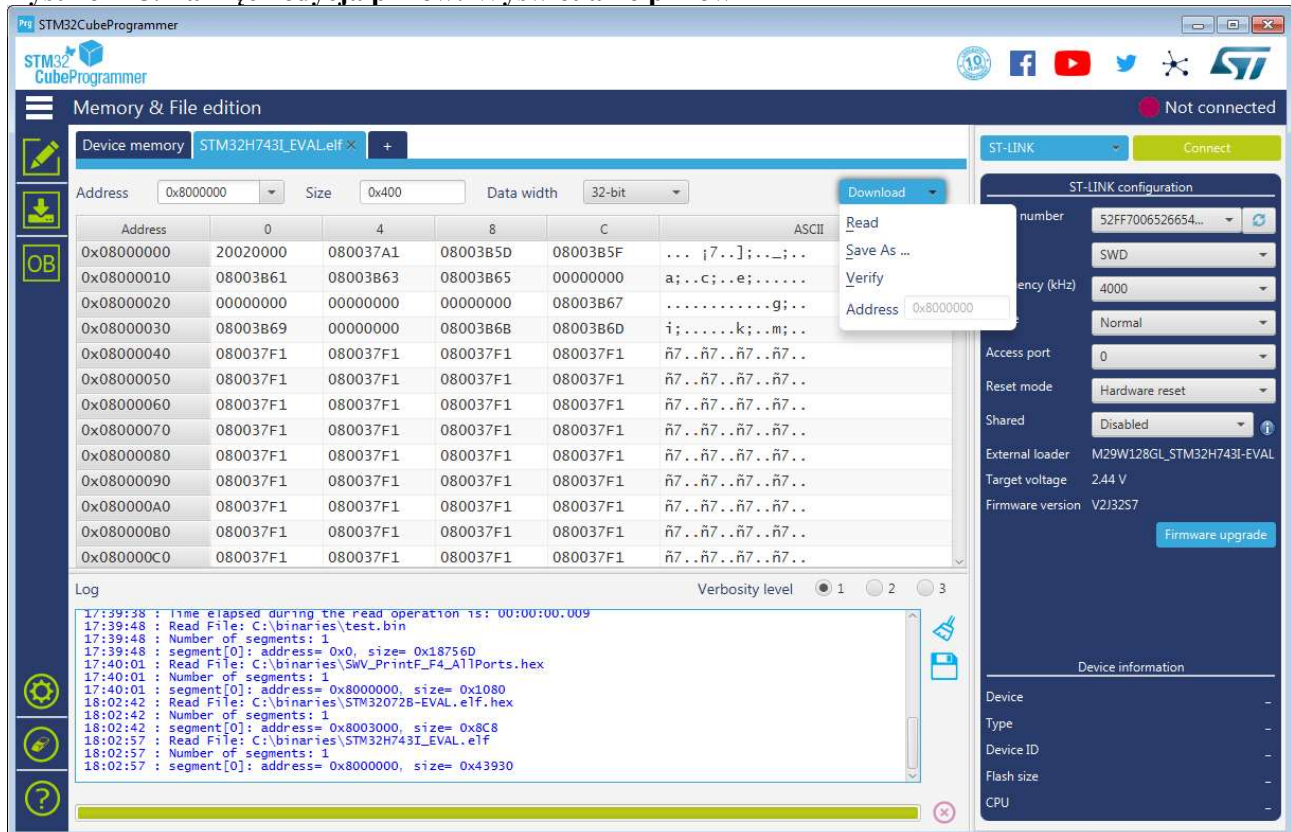
2.2.2 Czytanie i wyświetlanie pliku

Aby otworzyć i wyświetlić plik, po prostu kliknij „+” i wybierz menu „Otwórz plik”, jak pokazano na ilustracji [Rysunek 14](#).

Obsługiwane formaty plików to pliki binarne (.bin), pliki ELF (.elf, .axf, .out), pliki szesnastkowe firmy Intel (.hex) i pliki Motorola S-record (.Srec).

Po otwarciu i przeanalizowaniu pliku jest on wyświetlany na specjalnej karcie ze swoją nazwą, np. zilustrowane w [Rysunek 15](#). Rozmiar pliku jest wyświetlany w polu „Rozmiar”, a adres początkowy pliku hex, srec lub ELF są wyświetlane w polu „Adres”, dla pliku binarnego jest to 0.

Rysunek 15. Pamięć i edycja plików: Wyświetlanie plików



Pole adresu można zmodyfikować, aby wyświetlić zawartość pliku, zaczynając od przesunięcia. Za pomocą menu kontekstowe karty lub przycisk akcji, możesz pobrać plik za pomocą „Pobierz” przycisk / menu. W przypadku pliku binarnego musisz określić adres pobierania w polu „Adres” menu. Możesz sprawdzić, czy plik został już pobrany za pomocą menu „Weryfikuj”, a także zapisać w innym formacie (.bin, .hex lub .srec).

Jeśli chodzi o zakładkę „Pamięć urządzenia”, możesz wyświetlić zawartość pamięci plików w różnych formatach

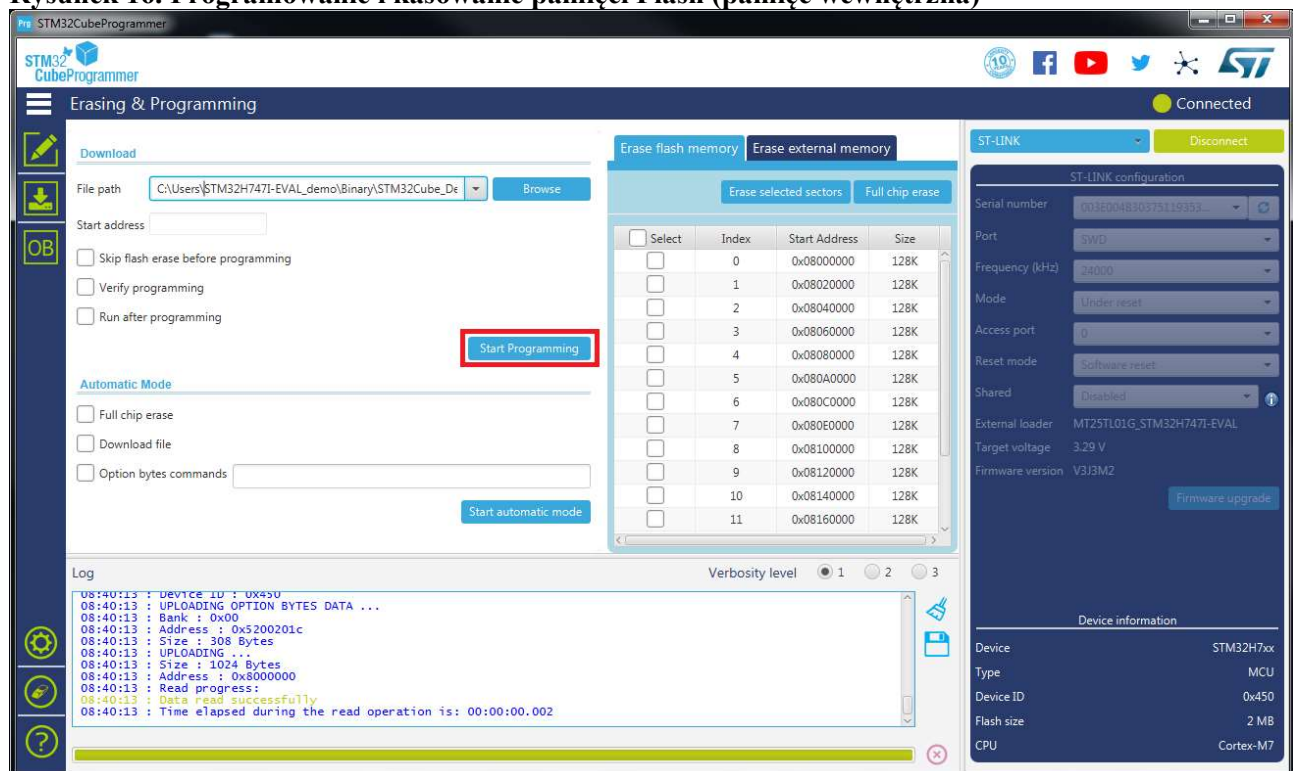
(8-, 16- i 32-bitowe) za pomocą pola kombi „Szerokość danych”.

2.3 Programowanie i kasowanie pamięci

Panel ten jest przeznaczony do programowania i kasowania pamięci Flash.

2.3.1 Programowanie wewnętrznej pamięci Flash

Rysunek 16. Programowanie i kasowanie pamięci Flash (pamięć wewnętrzna)



Kasowanie pamięci

Po podłączeniu do celu sektory pamięci są wyświetlane w panelu po prawej stronie pokazujący adres początkowy i rozmiar każdego sektora. Aby usunąć jeden lub więcej sektorów, wybierz je w pierwszej kolumnie, a następnie kliknij przycisk „Usuń wybrane sektory”.

Przycisk „Full chip erase” kasuje całą pamięć Flash.

Programowanie pamięci

Aby zaprogramować pamięć, wykonaj następujące czynności:

1. Kliknij przycisk przeglądania i wybierz plik do zaprogramowania. Format pliku obsługiwane są pliki binarne (.bin), pliki ELF (.elf, .axf, .out), pliki szesnastkowe firmy Intel (.hex) i Pliki Motorola S-record (.Srec).
 2. W przypadku programowania pliku binarnego należy ustawić adres.
 3. Wybierz opcje programowania:
 - Sprawdź po zaprogramowaniu: Odczytaj zaprogramowaną pamięć i porównaj bajt na bajt z plikiem.
 - Pomiń kasowanie Flasha przed programowaniem: jeśli zaznaczone, pamięć nie jest kasowana przed programowaniem. Tę opcję należy zaznaczyć tylko wtedy, gdy masz pewność, że plik pamięć docelowa jest już wymazana.
 - Uruchom po zaprogramowaniu: Uruchom aplikację zaraz po zaprogramowaniu.
 4. Kliknij przycisk „Rozpocznij programowanie”, aby rozpocząć programowanie.
- Pasek postępu w dolnej części okna pokazuje postęp usuwania i operacje programowania.

2.3.2 Programowanie zewnętrznej pamięci Flash

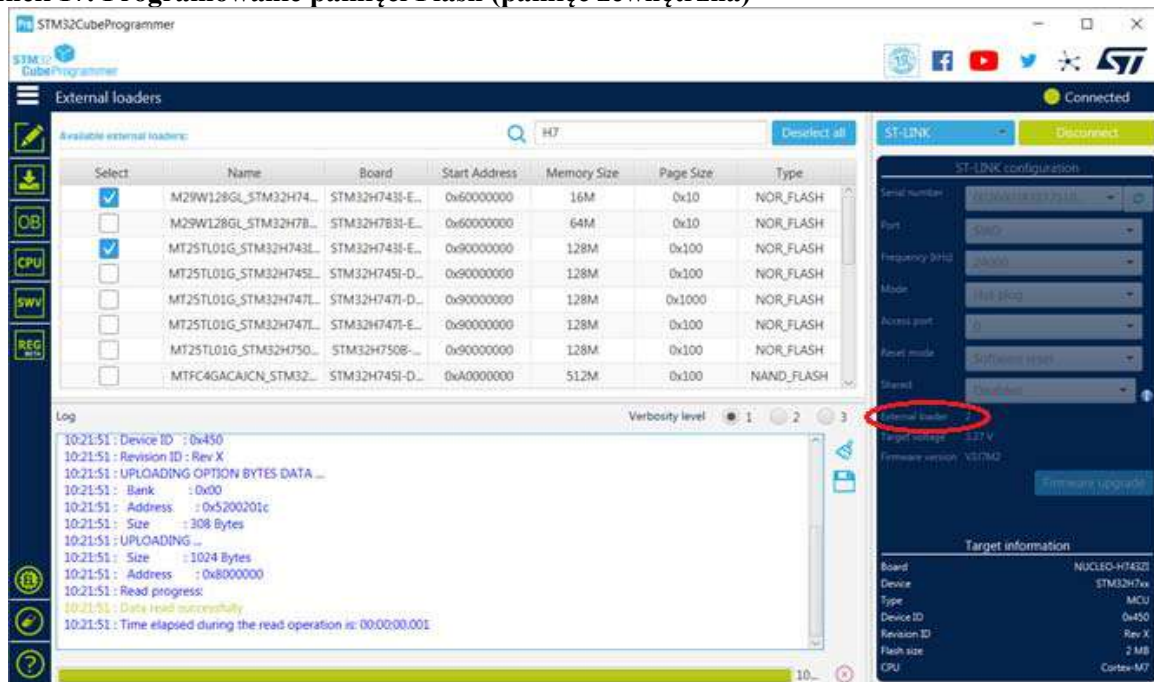
Aby zaprogramować pamięć zewnętrzną podłączoną do mikrokontrolera za pomocą dowolnego z dostępnych interfejsów (np. SPI, FMC, FSMC, QSPI, OCTOSPI) potrzebny jest zewnętrzny program ładujący.

STM32CubeProgrammer jest dostarczany z zewnętrznymi programami ładującymi dla większości dostępnych płyt STM32 Evaluation i Discovery dostępnych w katalogu „bin/ExternalLoader”. Jeśli potrzebujesz utworzyć nowy zewnętrzny program ładujący, zobacz Sekcja 2.3.3 po więcej szczegółów.

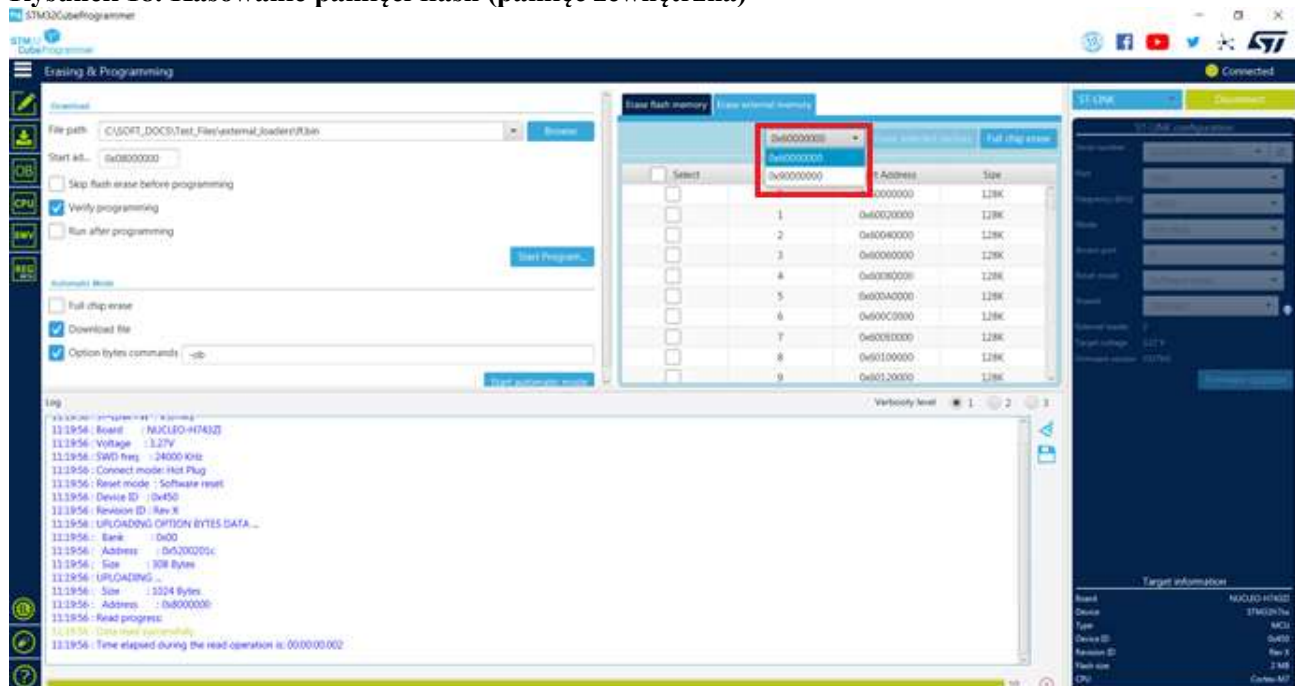
Aby zaprogramować pamięć zewnętrzną, wybierz jeden lub więcej zewnętrznych programów ładujących z panelu „ExternalLoader”, które mają być używane przez narzędzie do odczytu, programowania lub kasowania pamięci zewnętrznych, jak pokazano na Rysunku 17. Po wybraniu zewnętrznego programu ładującego jest (są) używane do dowolnej operacji pamięci w swoim (ich) zakresie pamięci.

Zakładka „Wymazywanie zewnętrznej pamięci flash” po prawej stronie panelu „Erasing and Programming” wyświetla sektory pamięci dla każdego wybranego programu ładującego i umożliwia wymazywanie sektorów lub całego chipa, jak pokazano na rysunku 18.

Rysunek 17. Programowanie pamięci Flash (pamięć zewnętrzna)



Rysunek 18. Kasowanie pamięci flash (pamięć zewnętrzna)



2.3.3 Opracowywanie niestandardowych programów ładujących dla pamięci zewnętrznej

Na podstawie przykładów dostępnych w katalogu „*bin / ExternalLoader*” użytkownicy mogą opracowywać własne programy ładujące dla danej pamięci zewnętrznej. Te przykłady są dostępne dla trzy zestawów narzędzi: MDK-ARM™, EWARM i TrueSTUDIO®. Rozwój niestandardowych programów ładujących można wykonać za pomocą jednego z tych łańcuchów narzędzi, zachowując te same konfiguracje kompilatora/linkera, jak w przykładach.

Zewnętrzny mechanizm programowania Flash jest taki sam, jak używany w STM32 ST-LINK narzędzie użytkowe. Każdy program ładujący Flash opracowany do użytku z narzędziem ST-LINK jest kompatybilny z narzędziem STM32CubeProgrammer i może być używany bez żadnych modyfikacji.

Aby utworzyć nowy program ładujący pamięć zewnętrzną, wykonaj poniższe czynności:

1. Aktualizacja informacji o urządzeniu w *StorageInfo* struktury w *Dev_Inf.c* pliku z poprawne informacje dotyczące pamięci zewnętrznej.
2. Przepisz odpowiedni kod funkcji w pliku *Loader_Src.c*.
3. Zmień nazwę pliku wyjściowego.

Uwaga:

*Niektóre funkcje są obowiązkowe i nie można ich pominąć (zobacz opis funkcji w Plik *Loader_Src.c*).*

Nie wolno modyfikować plików konsolidatora ani plików rozproszonych.

Po zbudowaniu projektu zewnętrznego modułu ładującego generowany jest plik ELF. Rozszerzenie ELF plik zależy od używanego łańcucha narzędzi (.axf dla Keil, .out dla EWARM i .elf dla TrueSTUDIO lub dowolny łańcuch narzędzi oparty na gcc).

Rozszerzenie pliku ELF należy zmienić na „.stldr”, a plik należy skopiować pod katalog „*bin / ExternalLoader*”.

Plik *Loader_Src.c*

Opracowanie zewnętrznego modułu ładującego dla pamięci opartego na określonym adresie IP wymaga następujących elementów

Funkcje:

- Funkcja **Init**

Funkcja **Init** definiuje użyte piny GPIO łączące pamięć zewnętrzną z urządzeniem i inicjalizuje zegar używanych adresów IP.

Zwraca 1 w przypadku sukcesu i 0 w przypadku niepowodzenia.

int Init (void)

- Funkcja **zapisu**

Funkcja **Write** programuje bufor zdefiniowany przez adres w zakresie RAM.

Zwraca 1 w przypadku sukcesu i 0 w przypadku niepowodzenia.

int Write (adres uint32_t, rozmiar uint32_t, bufor uint8_t *)

- Funkcja **SectorErase**

Funkcja **SectorErase** usuwa określone sektory pamięci.

Zwraca 1 w przypadku sukcesu i 0 w przypadku niepowodzenia.

int SectorErase (uint32_t StartAddress, uint32_t EndAddress)

Gdzie „**StartAddress**” oznacza adres pierwszego sektora do usunięcia i

„**EndAddress**” to adres sektora końcowego, który ma zostać usunięty.

Uwaga:

Ta funkcja nie jest używana w przypadku zewnętrznego modułu ładującego SRAM.

Konieczne jest zdefiniowanie wyżej wymienionych funkcji w zewnętrznym module ładującym. Są one wykorzystywane przez narzędzie, aby usunąć i zaprogramować pamięć zewnętrzną. Na przykład, jeśli użytkownik kliknie plik przycisk programu z menu zewnętrznego modułu ładującego, narzędzie wykonuje następujące czynności:

- Automatycznie wywołuje funkcję **Init** w celu zainicjowania interfejsu (QSPI, FMC...) i

Pamięć flash

- Wywołuje **SectorErase ()**, aby usunąć potrzebne sektory pamięci Flash
- Wywołuje funkcję **Write ()** w celu zaprogramowania pamięci

Oprócz tych funkcji możesz także zdefiniować poniższe funkcje:

- Funkcja **odczytu**

Funkcja **Read** służy do odczytu określonego zakresu pamięci i zwraca odczyt w buforze w pamięci RAM.

Zwraca 1 w przypadku sukcesu i 0 w przypadku niepowodzenia.

int Read (adres uint32_t, rozmiar uint32_t, bufor uint16_t *)

Gdzie „**Adres**” = adres początkowy operacji odczytu, „**Rozmiar**” jest rozmiarem odczytu operacja, a „**bufor**” jest wskaźnikiem do odczytu danych.

Uwaga:

W przypadku pamięci QSPI / OSPI (Quad-SPI / Octo-SPI), tryb mapowania pamięci można zdefiniować w funkcji Init; w takim przypadku funkcja Read jest bezużyteczna, ponieważ dane mogą być czytać bezpośrednio z interfejsu JTAG / SWD.

- **Sprawdź** działanie

Funkcja **weryfikacji** jest wywoływana po wybraniu trybu „weryfikacja podczas programowania”.

Ta funkcja sprawdza, czy zaprogramowana pamięć odpowiada buforowi zdefiniowanemu w pamięć RAM. Zwraca uint64 zdefiniowany w następujący sposób:

Wartość zwracana = ((suma kontrolna << 32) + AddressFirstError)

gdzie „**AddressFirstError**” to adres pierwszej niezgodności, a „**checksum**” to wartość sumy kontrolnej zaprogramowanego bufora

uint64_t Verify (uint32_t FlashAddr, uint32_t RAMBufferAddr, uint32_t rozmiar)

- Funkcja **MassErase**

Funkcja **MassErase** kasuje całą pamięć.

Zwraca 1 w przypadku sukcesu i 0 w przypadku niepowodzenia.

int MassErase (void)

- Funkcja sumy kontrolnej

Wszystkie opisane funkcje zwracają 1 w przypadku pomyślnej operacji i 0 w przypadku niepowodzenia.

Plik Dev_Inf.c

Struktura StorageInfo zdefiniowana w tym pliku dostarcza informacji o pamięci zewnętrznej.

Przykład rodzaju informacji, które definiuje ta struktura, podano poniżej:

#if zdefiniowane (__ICCARM__)

__root struct StorageInfo const StorageInfo = {

#jeszcze

struct StorageInfo const StorageInfo = {

#endif

"External_Loader_Name", // nazwa urządzenia + numer wersji

MCU_FLASH, // Typ urządzenia

0x08000000, // Adres początkowy urządzenia

```

0x00100000, // Rozmiar urządzenia w bajtach (1 MB / 8 Mb)
0x00004000, // Programowanie rozmiaru strony 16 KB
0xFF, // Początkowa zawartość skasowanej pamięci
// Określ rozmiar i adres sektorów (zobacz przykład poniżej)
0x00000004, 0x00004000, // Numer sektora: 4, Rozmiar sektora: 16 KB
0x00000001, 0x00010000, // Numer sektora: 1, Rozmiar sektora: 64 KB
0x00000007, 0x00020000, // Numer sektora: 7, Rozmiar sektora: 128 KB
0x00000000, 0x00000000,
};

```

2.4 Bajty opcji

Panel bajtów opcji pozwala użytkownikowi odczytać i wyświetlić docelowe bajty opcji pogrupowane według kategorii. Bity opcji są wyświetlane w tabelach z trzema kolumnami zawierającymi bity nazwa, wartość i opis wpływu na urządzenie.

Następnie użytkownik może modyfikować wartości tych bajtów opcji, aktualizując pola wartości klikając przycisk zastosuj, które programy, a następnie sprawdzają, czy zmodyfikowana opcja bajtów są poprawnie zaprogramowane. Użytkownik może w dowolnym momencie kliknąć przycisk czytaj, aby przeczytać i odświeżyć wyświetlane bajty opcji.

Rysunek 19. Panel bajtów opcji

The screenshot shows the 'Option bytes' configuration window. It features a sidebar with icons for editing, saving, and viewing options. The main area is titled 'Option bytes' and contains a table for 'Read Out Protection'. The table has three columns: 'Name', 'Value', and 'Description'. The 'RDP' option is listed with a value of 'AA'. Below this, there are several expandable sections: 'RSS', 'BOR Level', 'User Configuration', 'Boot address Option Bytes', and 'PCROP Protection'. The 'PCROP Protection' section is expanded, showing a table with three rows: 'PROT_AREA_START1', 'PROT_AREA_END1', and 'DMEP1'. The 'DMEP1' option is checked. At the bottom right, there are 'Apply' and 'Read' buttons.

Name	Value	Description
RDP	AA	Read protection option byte. The read protection is used to protect the software code stored in Flash memory. AA : Level 0, no protection BB : Level 1, read protection of memories CC : Level 2, chip protection

Name	Value	Description
PROT_AREA_START1	0xff 0x8001fe0	Flash Bank 1 PCROP start address
PROT_AREA_END1	0x0 0x8000000	Flash Bank 1 PCROP End address. Deactivation of PCROP can be done by enabling DMEP1 bit and changing RDP from level 1 to level 0 while putting
DMEP1	☒	Unchecked : Flash Bank 1 PCROP zone is kept when RDP level regression (change from level 1 to 0) occurs Checked : Flash Bank 1 PCROP zone is erased when RDP level regression (change from level 1 to 0) occurs

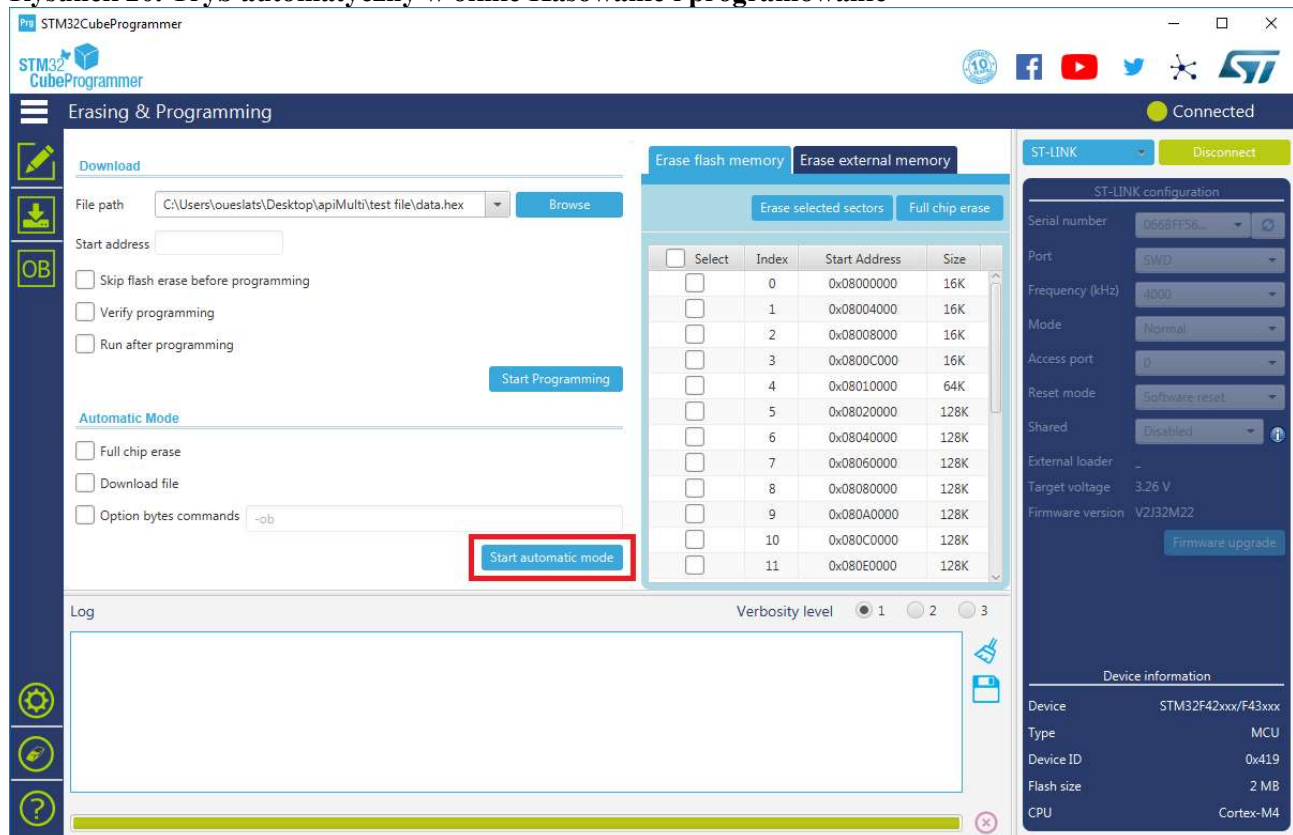
Więcej informacji można znaleźć w sekcji dotyczącej bajtów opcji w instrukcji programowania pamięci Flash oraz instrukcja obsługi dostępna na www.st.com.

2.5 Tryb automatyczny

Funkcja trybu automatycznego pokazana w oknie Kasowanie i programowanie (patrz [Rysunek 20](#)) umożliwia użytkownikowi programowanie i konfigurowanie urządzeń STM32 w pętli. Dozwolone działania:

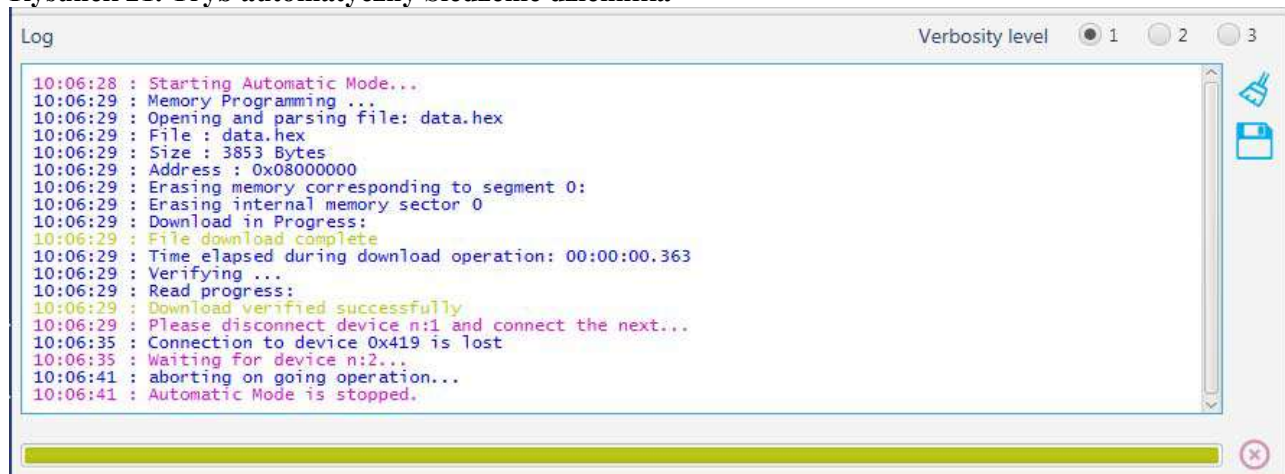
- Pełne wymazywanie chipa: kasowanie całej pamięci Flash
- Pobierz plik: aktywuj i ustaw opcje programowania w sekcji Download:
 - Ścieżka pliku
 - Adres początkowy
 - Pomiń kasowanie pamięci flash przed programowaniem
 - Sprawdź programowanie
 - Uruchom po zaprogramowaniu
- Polecenia bajtów opcji: skonfiguruj urządzenie, ustawiając wiersz polecenia bajtów opcji

Rysunek 20. Tryb automatyczny w oknie Kasowanie i programowanie



Wszystkie ślady trybu automatycznego są wskazywane w panelu dziennika (patrz [Rysunek 21](#)), aby pokazać plik ewolucja procesu i komunikaty dotyczące interwencji użytkownika.

Rysunek 21. Tryb automatyczny Śledzenie dziennika



Graficzny przewodnik

- Połączenie z pierwszym celem musi zostać ustanowione przed wykonaniem trybu automatycznego w celu zebrania wartości parametrów połączenia związanych ze wszystkimi kolejnymi urządzeniami.
- Jeśli zaznaczona jest opcja Pobierz plik, system bierze pod uwagę wszystkie opcje pobierania pliku, w przeciwnym razie zostanie wykonana dowolna opcja pobierania.
- Jeśli zaznaczone jest polecenie opcji bajtów, pole tekstowe jest aktywowane, a następnie użytkownik może wstawiać polecenia bajtów opcji (podobnie jak polecenia CLI) i upewnić się, że nie ma spacji na początku:
-ob [opcjonalny bajt = wartość] [opcjonalny bajt = wartość] [opcjonalny bajt = wartość]...
- Przykład polecenia bajtów opcji: „**-ob BOR_LEV = 0 nBOOT0 = 1**”
- Jeśli zostanie naciśnięty przycisk Uruchom tryb automatyczny, system wchodzi w pętlę, aż do wywołania zatrzymania systemu.
- Gdy tryb automatyczny jest w stanie wykonywania, wszystkie obiekty graficzne są wyłączone.
- Użytkownik może zatrzymać proces w dowolnym preferowanym czasie, naciskając przycisk anulowania lub zatrzymaj przycisk trybu automatycznego.