



# Instrukcja programowania płytek edycji 2014

## Spis treści

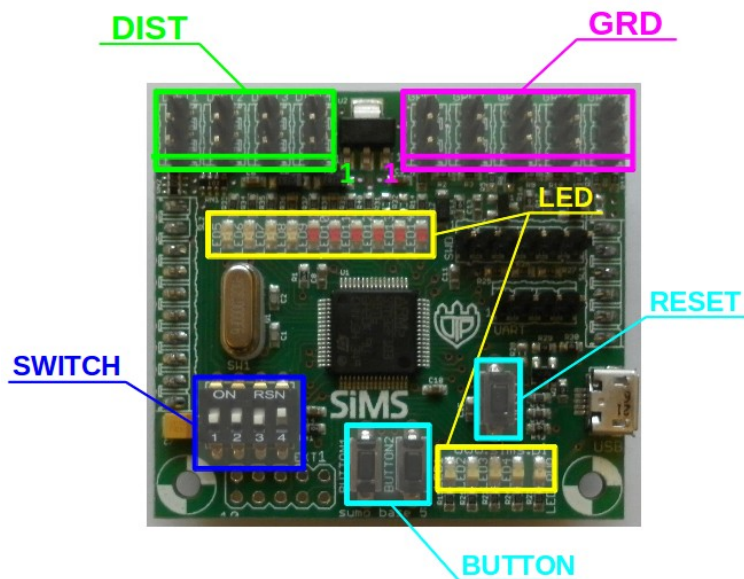
1. Opis płytki procesorowej (sumo_base_5).....	1
1.1. Podstawowe elementy płytki.....	2
1.2. Pozostałe elementy płytki.....	3
2. Opis płytki sterującej napędami (sumo_motor_5).....	3
2.1. Elementy płytki.....	4
2.2 Parametry układu.....	4
3. Zalecane rozmieszczenie czujników i silników.....	5
4. Programowanie układu przez USB (DfuSe) krok po kroku.....	5
5. Konfiguracja środowiska CooCox i kompilacja programu.....	8
5.1. Instalacja i przygotowanie środowiska.....	8
5.2. Otwieranie i kompilacja projektu.....	9
5.3. Przygotowanie pliku *dfu.....	10
6. Debugowanie za pomocą programatora ST-link.....	11
6.1. Przygotowanie środowiska.....	11
6.2. Debugowanie.....	12
6.3. Powrót do pracy z bootloaderem.....	13

## 1. Opis płytki procesorowej (sumo\_base\_5)

Układ stanowi „mózg” robota, posiada złącza czujników, interfejsy komunikacyjne, diody LED, przyciski i przełączniki ogólnego przeznaczenia. Jest on oparty na mikroprocesorze STM32F103RBT6 z rodziny ARM Cortex-M3. Układ można programować na trzy sposoby:

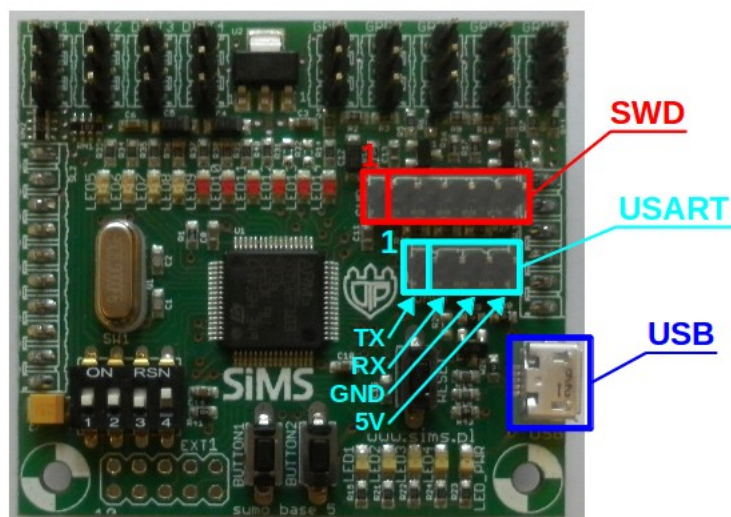
- Przez interfejsu USB, za pomocą zaprogramowanego bootloadera DFU (Device Firmware Upgrade). **Metoda zalecana**, opisana w punkcie 4.
- Przez interfejs SWD, za pomocą programatora ST-link. Metoda pozwala na debugowanie programu.
- Przez interfejs USART (port szeregowy).

## 1.1. Podstawowe elementy płytki



- LED1 – 11 – diody LED do dyspozycji użytkownika (programisty)
- LED12 – dioda LED informująca o włączeniu portu USB
- LED\_PWR – zielona dioda, świeci gdy zasilanie jest podłączone
- SWITCH1 – 3 – przełączniki do dyspozycji programisty
- SWITCH4 – Wybór trybu bootowania. Poza programowaniem przez UART, **ZAWSZE W POZYCJI ON**
- BUTTON1, 2 – przyciski do dyspozycji użytkownika (programisty)
- RESET – Przycisk resetu
- GRD1 – 5 – wejścia czujników podłóża (do wykrywania krawędzi ringu)
- DIST1 – 4 – wejścia czujników odległości (do wykrywania przeciwnika)

## 1.2. Pozostałe elementy płytki

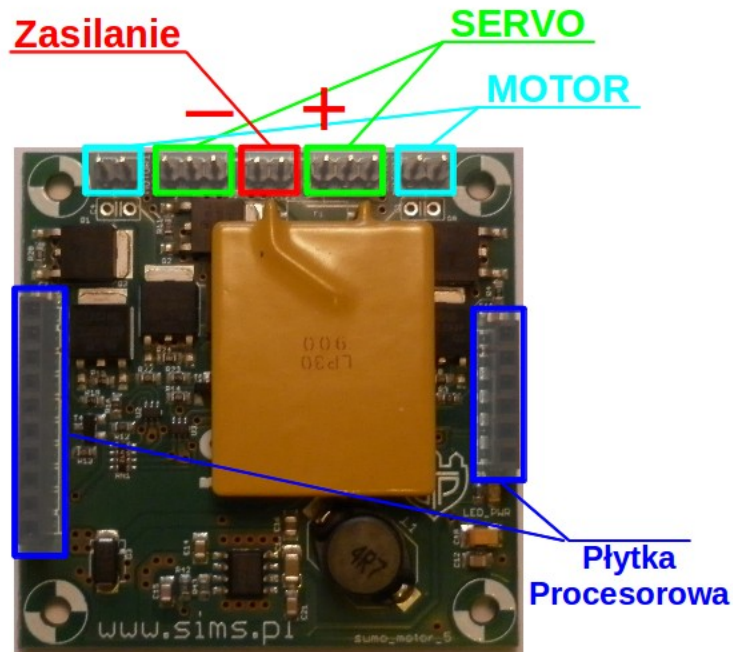


- USB – służy głównie do programowania procesora. Może być też wykorzystany w innych celach
- SWD – Złącze debuggera.
- USART – port szeregowy, pozwala na komunikację z procesorem w czasie działania programu. Może też służyć do programowania.
- EXT – złącze z wyprowadzonymi niewykorzystanymi sygnałami procesora. Zawiera 4 kanały przetwornika ADC, interfejsy SPI i I2C. Umożliwia np. podłączenie dodatkowych czujników odległości, karty SD, akcelerometru i żyroskopu.

## 2. Opis płytki sterującej napędami (sumo\_motor\_5)

Układ odpowiedzialny jest za zasilanie płytki bazowej i serwomechanizmów, oraz sterowanie silnikami DC. Za wytwarzanie napięcia 5V odpowiedzialna jest przetwornica impulsowa ST1S10.

## 2.1. Elementy płytki

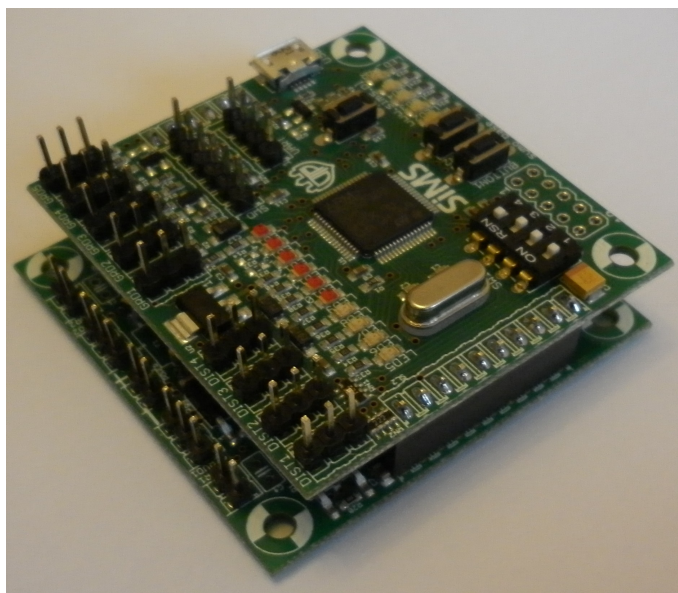


- Zasilanie – złącze baterii, zakres napięć zasilania: **6V – 14V**. Zalecane jest stosowanie baterii litowo-polimerowych 7,2V lub 11,1V. **UWAGA NA PRAWIDŁOWE PODŁĄCZENIE BATERII** ('+' baterii do '+' płytki, '-' baterii do '-' płytki)
- MOTOR1, 2 – złącza silników DC (prądu stałego, szczotkowych)
- SERVO1, 2 – złącza serwomechanizmów

## 2.2 Parametry układu

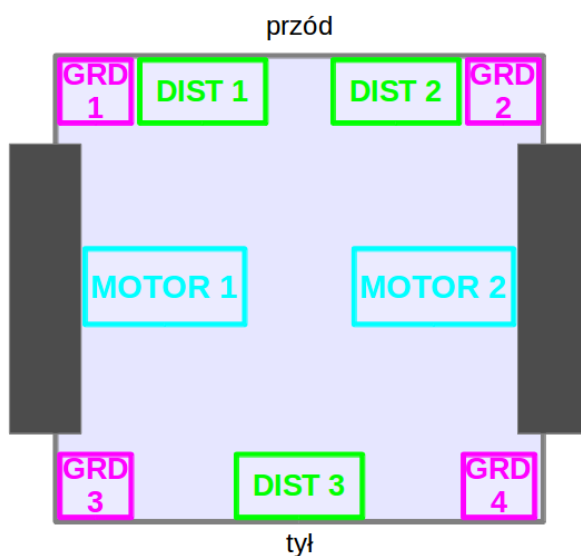
- Zakres napięć zasilania: **6-14V**.
- Maksymalny prąd pobierany przez serwa i płytki procesorowa: **1,5A**
- Maksymalny prąd pobieramy przez każdy z silników: **4A**
- Wymiary: **55x55mm**

Po złożeniu zestaw powinien wyglądać następująco:



### 3. Zalecane rozmieszczenie czujników i silników

Na schemacie pokazano zalecane rozmieszczenie czujników odległości (DIST), podłoża (GRD) i silników (MOTOR). Takie rozmieszczenie obsługuje program podstawowy.

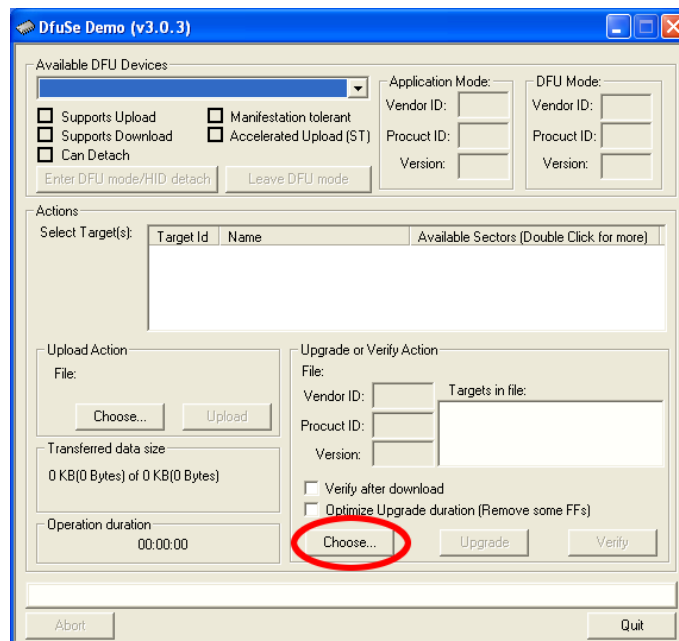


### 4. Programowanie układu przez USB (DfuSe) krok po kroku

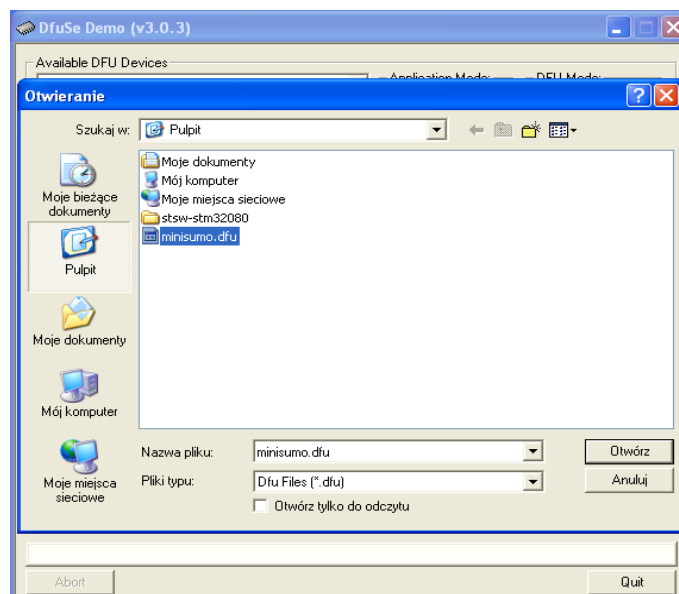
Metoda pozwala na wgrywanie plików \*.dfu (sposób ich przygotowania opisany w dalszej części instrukcji). Niezbędne do tego jest pobranie DfuSe.zip, rozpakowanie i zainstalowanie programu. Program można też pobrać ze strony:

<http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1533/PF257916>

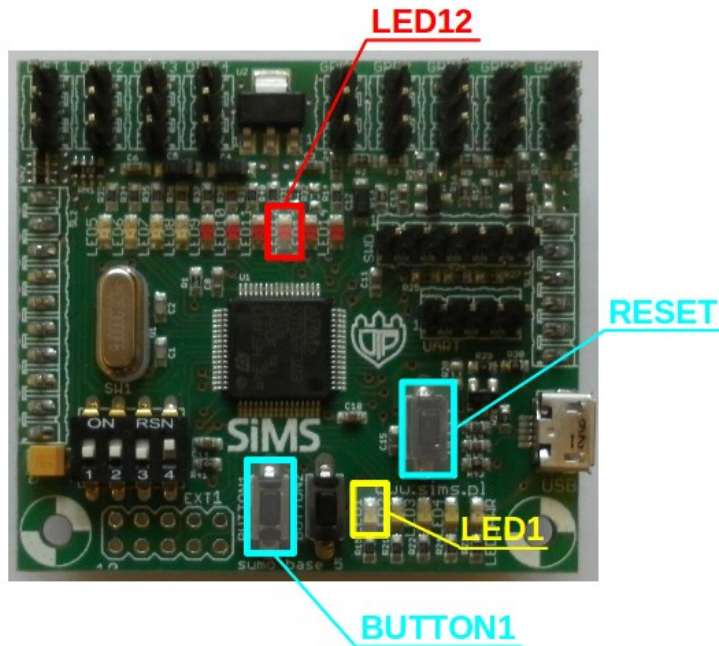
1. Uruchomić aplikację DfuSe Demo.



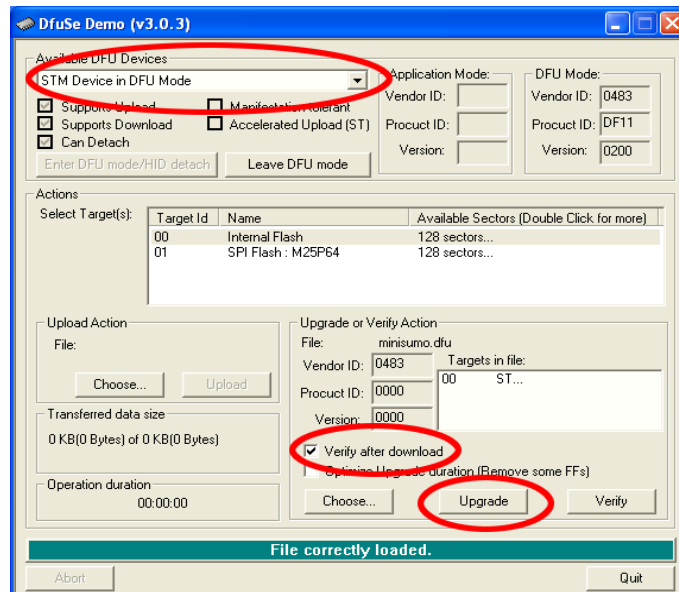
2. Kliknąć na przycisk *Choose* i wybrać plik \*.dfu.



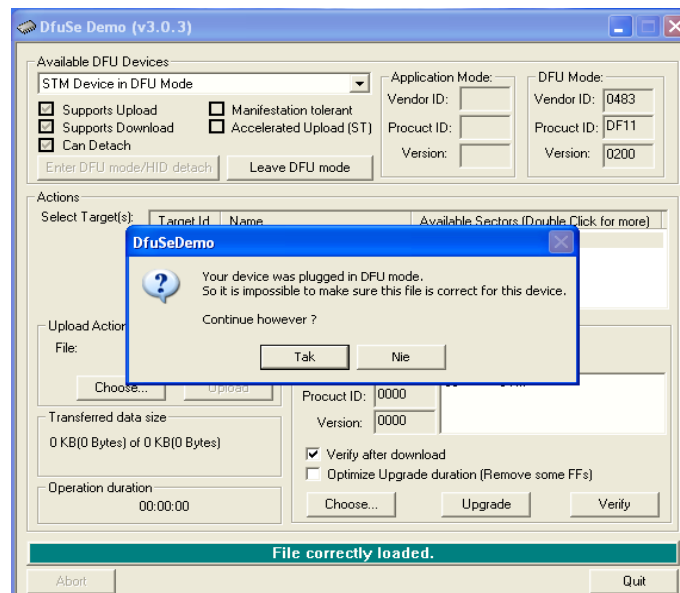
3. Podłączyć płytkę do portu USB, płytka musi być zasilana np. z baterii.
4. Trzymając wciśnięty przycisk BUTTON1 na płytce wcisnąć RESET. Spowoduje to uruchomienie bootloadera.



5. Powinna się zapalić dioda LED12 i mrgać dioda LED1
6. Po kilku sekundach w polu *Available DFU Devices* powinno się pojawić *STM Device in DFU Mode*.



7. Zaznaczyć opcję *Verify after download* i kliknąć *Upgrade*.
8. Przy pytaniu *Continue however* (wyskakujące okno) kliknąć *Tak*.



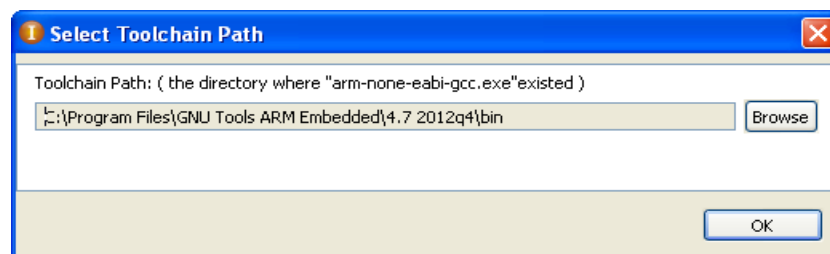
9. Jeśli wszystko pójdzie dobrze, na dolnym pasku postępu powinien pojawić się napis *Verify successful !*
10. Aby wystartować wgrany program wcisnąć przycisk RESET na płytce,.

## 5. Konfiguracja środowiska CoCoX i kompilacja programu

Środowisko CoCoX CoIDE i zestaw narzędzi GCC umożliwiają napisanie w języku C i kompilację programu dla robota.

### 5.1. Instalacja i przygotowanie środowiska

1. Pobrać środowisko CoIDE ze strony: [http://www.cocox.org/CooCox\\_CoIDE.htm](http://www.cocox.org/CooCox_CoIDE.htm)
2. Pobrać zestaw narzędzi GCC ze strony: <https://launchpad.net/gcc-arm-embedded/+download>
3. Zainstalować CoIDE i GCC, zapamiętać, gdzie został zainstalowany GCC
4. Uruchomić CoIDE, wybrać *Project* → *Select Toolchain Path*

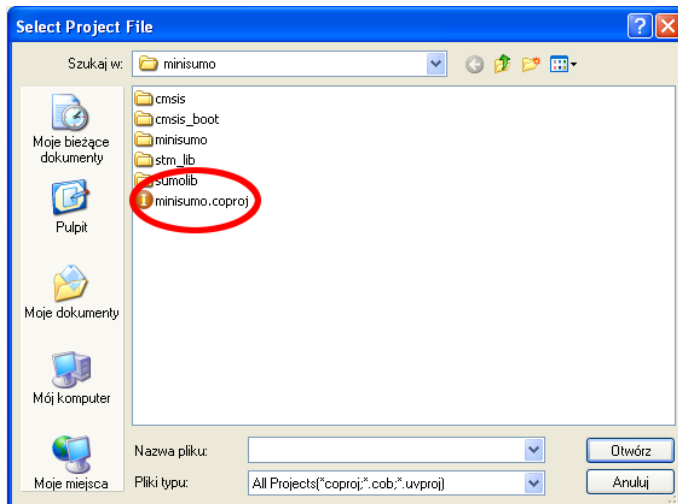


5. W oknie dialogowym wygrać ścieżkę do GCC (wcześniej zapamiętaną)

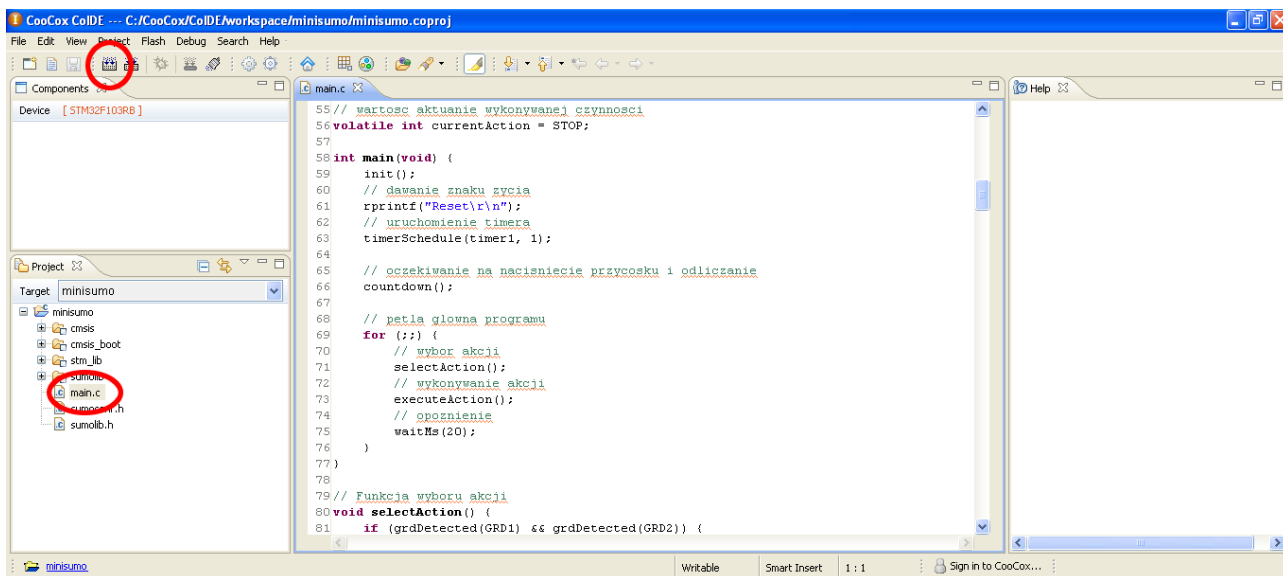


## 5.2. Otwieranie i kompilacja projektu

1. Pobrać plik minisumo\_coocox.zip – bibliotekę i przykładowy program dla robota
2. Rozpakować go w katalogu *workspace* programu CoIDE.
3. Uruchomić CoIDE.
4. Kliknąć Project → Open project i wybrać plik minisumo.coproj z rozpakowanego katalogu.



5. Aby edytować program, otworzyć w środowisku plik main.c

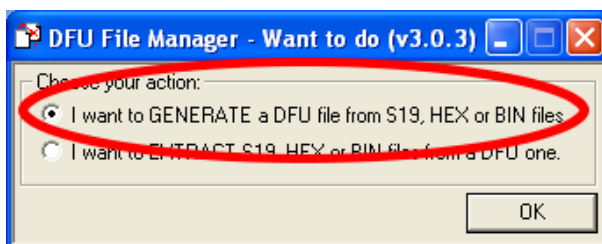


6. Kliknąć na przycisk build na górnym pasku
7. W oknie w dolnej części ekranu pojawi się log z kompilacji i ewentualne komunikaty o błędach
8. Aby program wgrać na płytke, trzeba przygotować plik \*.dfu (opisane poniżej), lub posłużyć się programatorem ST-link (opisane w punkcie 6)

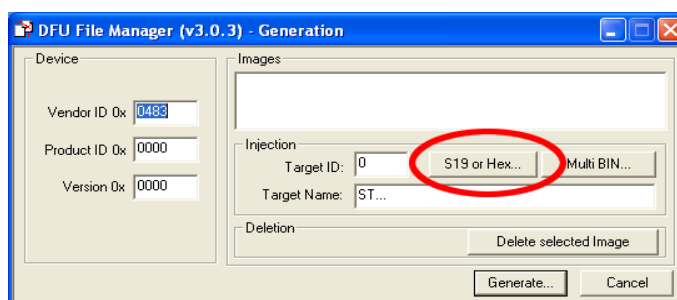
### 5.3. Przygotowanie pliku \*.dfu

Konwersja do plików \*.dfu jest potrzebna, jeśli ma zostać użyty bootloader opisany w punkcie 4.

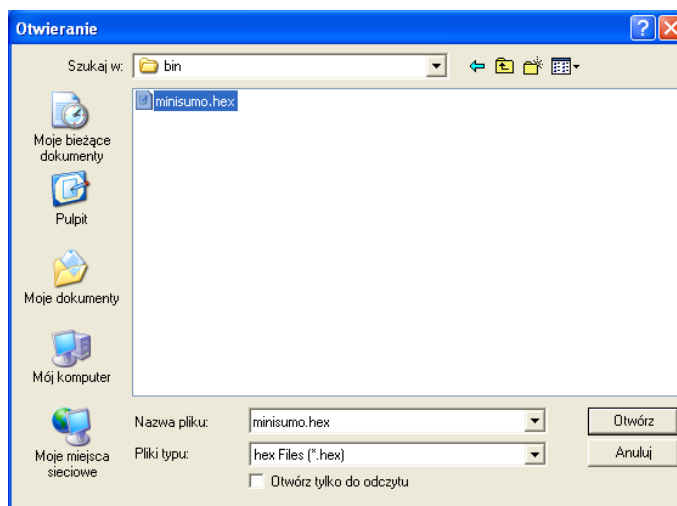
1. Uruchomić program *DFU File Manager* (zainstalowany wraz z DfuSe Demo).
2. Wybrać opcję pierwszą



3. Kliknąć na *S19 or HEX*.



4. Wybrać plik *minisumo.hex*. Znajduje się on w podkatalogu */minisumo/Debug/bin* katalogu z projektem *minisumo* (z punktu 5.2.). Być może trzeba będzie zmienić typ pliku na \*.hex w oknie wyboru pliku.



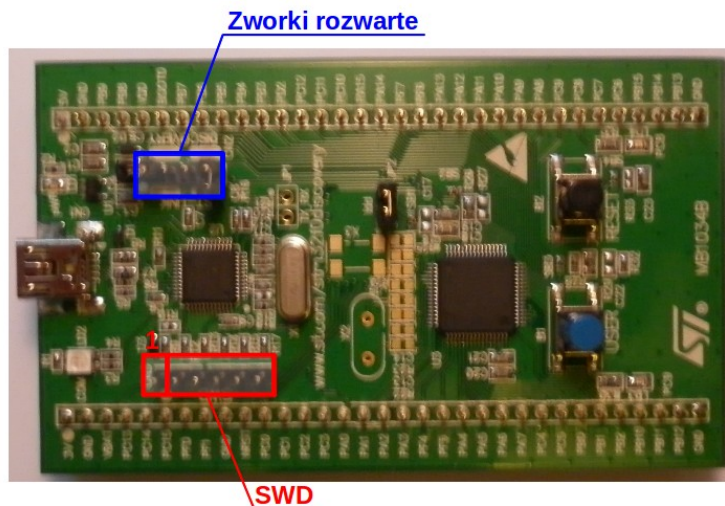
5. Kliknąć na *Generate* i podać nazwę i wybrać miejsce zapisu pliku \*.dfu.
6. Dalsze postępowanie opisane w punkcie 4.

## 6. Debugowanie za pomocą programatora ST-link

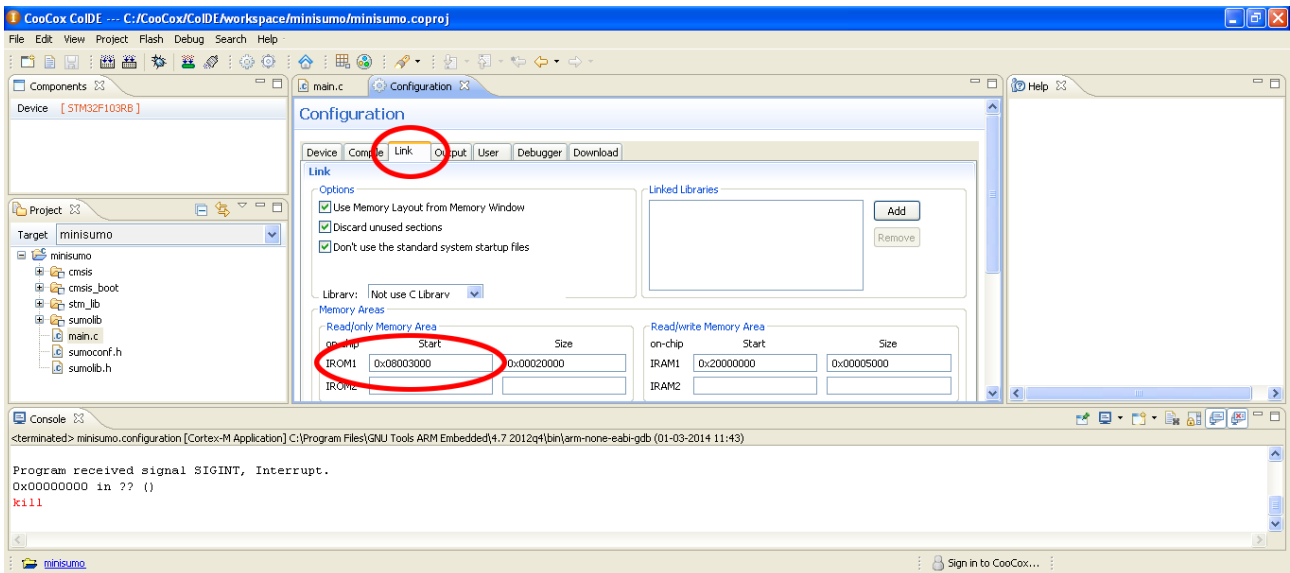
Programator ST-link umożliwia debugowanie kodu programu. Aby taki posiłk, najtaniej jest zaopatrzyć się z którąś z płytek ewaluacyjnych STM32xDISCOVERY, np. STM32F0DISCOVERY. Użycie programatora powoduje nadpisanie bootloadera, jednak nic nie stoi na przeszkodzie, żeby do niego później wrócić.

### 6.1. Przygotowanie środowiska

1. Pobrać i zainstalować sterownik do ST-linka:  
<http://www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF251168>
2. Wykonać czynności z punktów 5.1 i 5.2.
3. Ustawić zworki na płytce DISCOVERY w pozycji rozwartej
4. Połączyć złącza SWD płytki DISCOVERY i procesorowej, zwracając uwagę, gdzie jest pierwszy pin. Kolejność pinów płytki procesorowej jest taka sama jak dla układu STM32F0DISCOVERY.



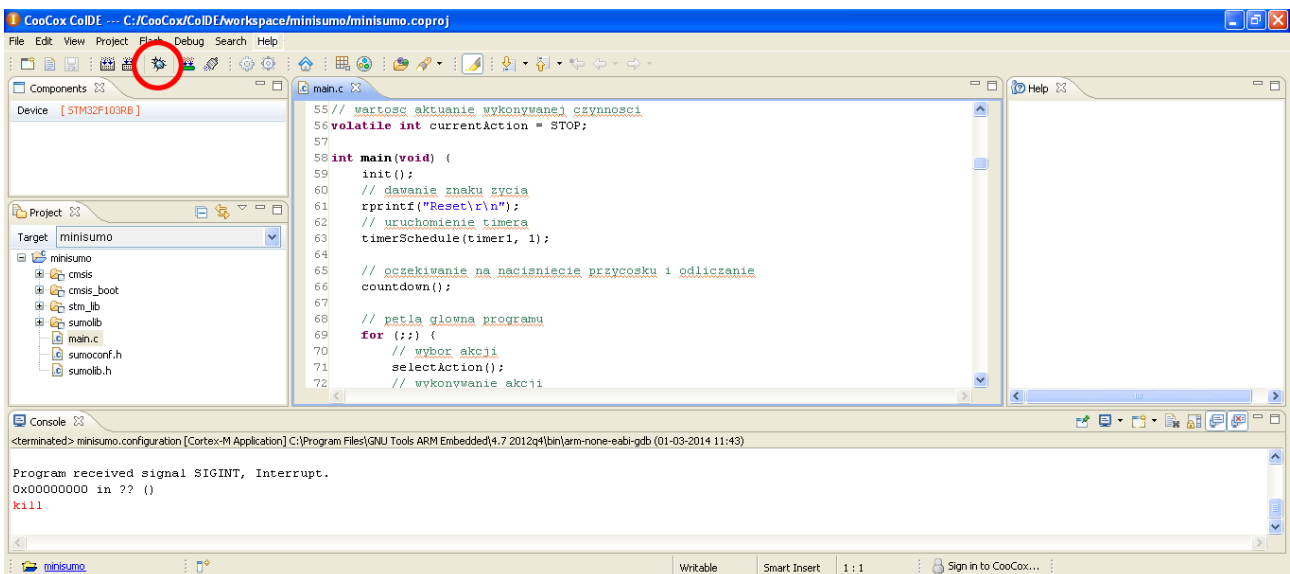
5. Włączyć programator do portu USB komputera.
6. W środowisku CoIDE kliknąć na *View* → *Configuration*.
7. Przejść do karty Link i wartość *IROM1* ustawić z 0x08003000 na 0x08000000.



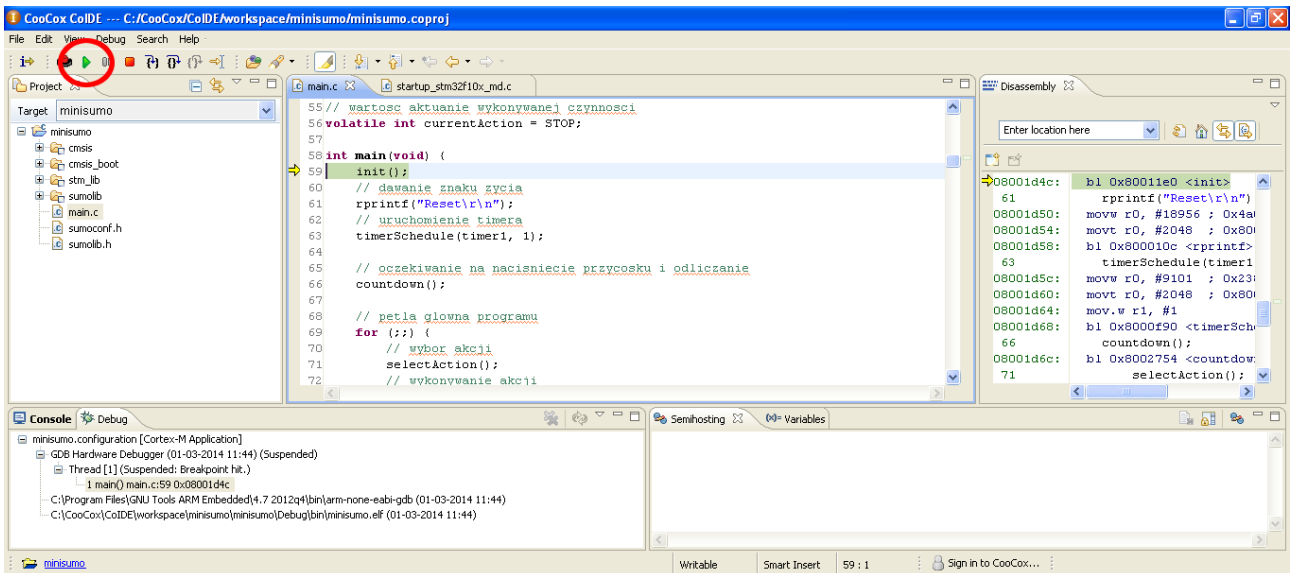
8. W pliku sumoconf.h zmienić `_BOOTLOADER_PRESENT` z 1 na 0
9. Skompilować projekt ponownie.

## 6.2. Debugowanie

1. Kliknąć na ikonę *Start Debug*

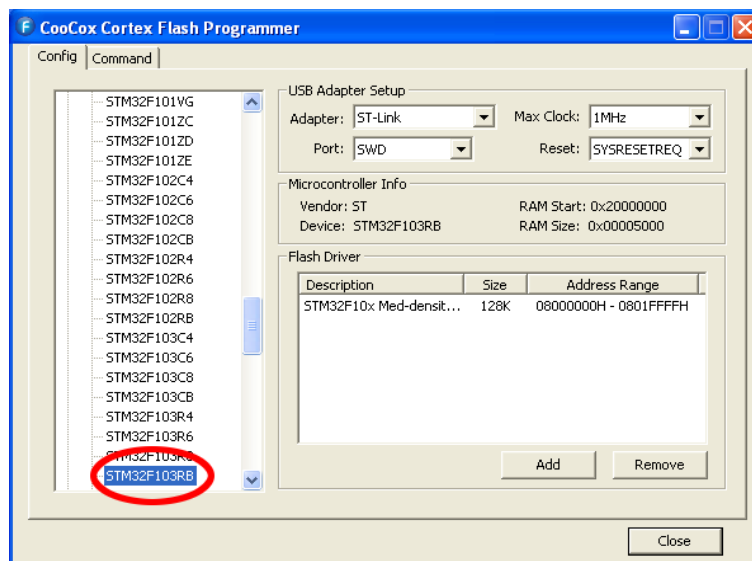


2. Środowisko zmieni perspektywę
3. Aby uruchomić aplikację kliknąć Run, możliwe też jest ustawianie pułapek, podglądanie zmiennych bądź praca krok po kroku

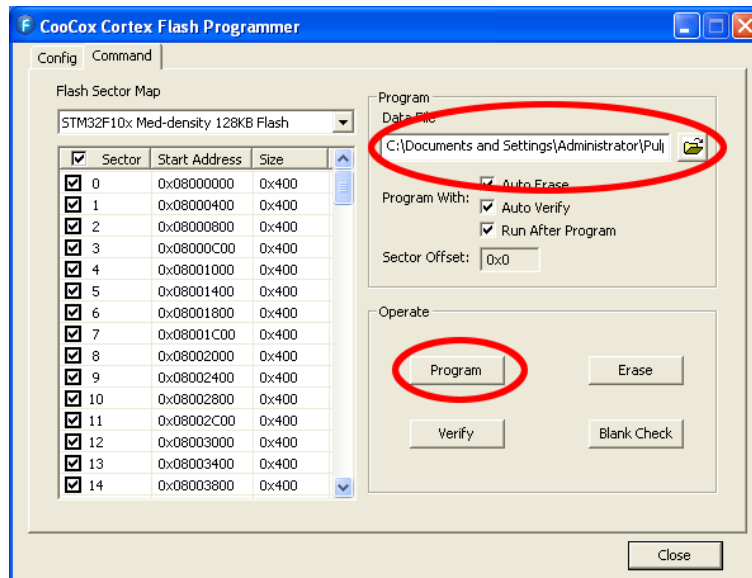


### 6.3. Powrót do pracy z bootloaderem

1. Pobrać i zainstalować program CoFlash: [http://www.cocox.org/CoFlash\\_Programmer.htm](http://www.cocox.org/CoFlash_Programmer.htm)
2. Pobrać plik DFU.elf
3. Uruchomić program CoFlash
4. W zakładce *Config* wybrać układ STM32F103RB



5. Przejść do zakładki *Command*, w polu *Data File* wskazać plik DFU.elf



6. Kliknąć *Program*. Po zaprogramowaniu procesor przejdzie w tryb DFU.