

Programowanie mikrokontrolera w STM32 Butterfly przez USB

Mikrokontrolery STM32 z grupy Connectivity Line, w skład której wchodzi układy STM32F105 oraz STM32F107, można programować na dwa główne sposoby: za pomocą interfejsu JTAG oraz bootloadera znajdującego się w pamięci systemowej.

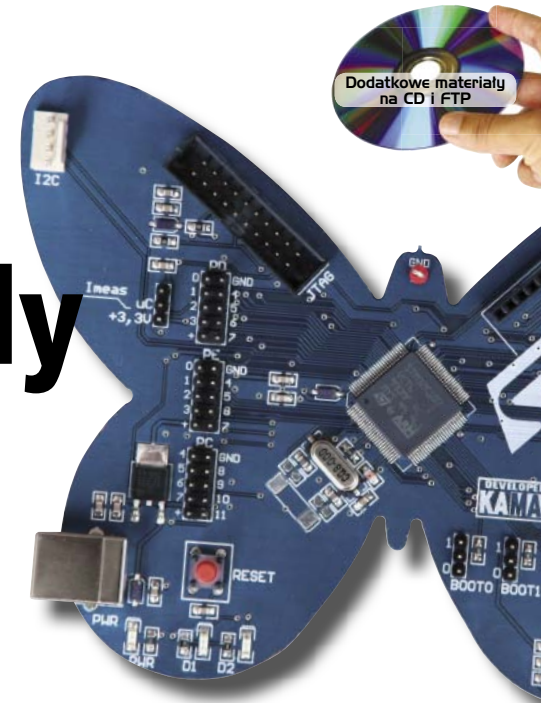
Zastosowanie interfejsu JTAG pociąga za sobą konieczność posiadania odpowiedniego sprzętowego interfejsu JTAG, co generuje dodatkowe koszty. Użycie bootloadera pozwala na zrezygnowanie z zakupu dodatkowego sprzętu, choć okupione jest nieco mniejszą wygodą podczas programowania pamięci.

W artykule przedstawiamy sposób zaprogramowania pamięci z wykorzystaniem klasy DFU (Device Firmware Upgrade) interfejsu USB na przykładzie zestawu STM32 Butterfly wyposażonego w mikrokontroler STM32F107VBT6.

Bootloader umieszczony w pamięci systemowej mikrokontrolerów STM32F105 oraz STM32F107 udostępnia trzy główne kanały komunikacji, za pośrednictwem których można wgrać program do pamięci Flash: UART (opisany w EP 12/2009), interfejs CAN oraz interfejs USB z wykorzystaniem klasy DFU. Programowanie pamięci mikrokontrolerów STM32 z grupy Medium Density (STM32F103) z wykorzystaniem mechanizmu DFU zostało opisane w numerze 6/2008 „Elektroniki Praktycznej”. W przypadku mikrokontrolerów STM32F103 konieczne było wgranie kodu odpowiedzialnego za obsługę klasy DFU przez użytkownika za pomocą innego kanału komunikacji, natomiast mikrokontrolery STM32F105 i STM32F107 zawierają ten kod w pamięci systemowej. Dzięki temu możliwe jest ich zaprogramowanie za pomocą interfejsu USB bezpośrednio po „wyjęciu z pudełka”. Jedyną rzeczą, jaka jest wymagana do zaprogramowania mikrokontrolera znajdującego się na płytce STM32 Butterfly poprzez USB, jest przewód USB typu A-A, tzn. zakończony z obu stron wtykiem typu A męskim. W sklepach z akcesoriami komputerowymi kabel oznaczony jako A-A najczęściej jest przedłużaczem, a więc zakończony jest z jednej strony wtykiem typu A żeńskim,

który do naszego zastosowania się nie nadaje. W przypadku problemów z zaopatrzeniem w odpowiedni kabelek można zmontować go samodzielnie np. z dwóch kabli typu A-B.

Część mikrokontrolerów z grupy Connectivity Line jest obciążona wadą, która w skrajnych przypadkach uniemożliwia wykorzystanie bootloadera. Wada ta występuje w przypadku mikrokontrolerów oznaczonych kodem daty produkcji mniejszym niż 937 i ma wpływ na działanie wszystkich kanałów komunikacji z bootloaderem. Szczegółowe informacje na temat problemu i jego rozwiązania zamieszczone są w dokumencie „STM32F105xx and STM32F107xx Errata sheet” dostępnym do pobrania na stronie www.st.com. Pisząc w skrócie: aby możliwe było uruchomienie bootloadera konieczne, jest wymuszenie stanu wysokiego na kilku dodatkowych, poza BOOT0, wyprowadzeniach mikrokontrolera. W przypadku mikrokontrolerów w obudowie LQFP64 wymuszenie tego stanu na wszystkich wymaganych liniach jest niemożliwe, w związku z czym bootloader w tych układach jest po prostu niedostępny. Na szczęście w przypadku układu STM32 Butterfly zastosowano mikrokontroler w obudowie 100-wyprowadzeniowej, dzięki czemu uruchomienie boot-

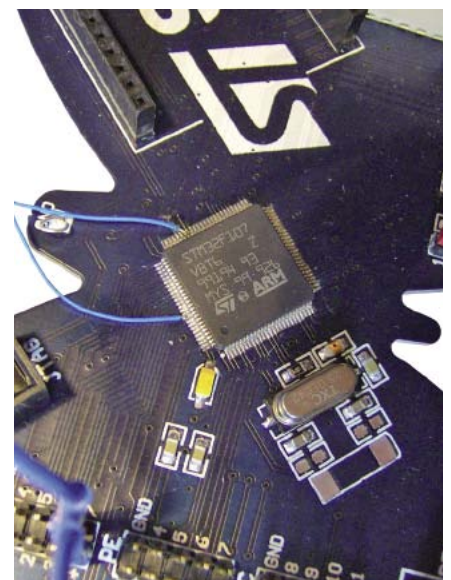


Dodatkowe materiały na CD i FTP:

ftp://ep.com.pl, user: 18366, pass: 3scpp470

- Program
- Przykłady
- Poprzednia część artykułu

loadera jest możliwe. Niestety, dwie z trzech interesujących nas linii I/O mikrokontrolera nie zostały wyprowadzone na żadne złącze, w związku z czym konieczne będzie dolutowanie przewodów bezpośrednio do wyprowadzeń mikrokontrolera. Wyprowadzenia te to linie PA10 i PB5, którym odpowiadają wyprowadzenia nr 69 oraz 91 obudowy TQFP-100. Do tych wyprowadzeń należy dolutować przewody i połączyć je z napięciem zasilania (fot. 1). Wyprowadzenie PD6 dostępne jest na złączu szpilkowym PD i należy je połączyć z napięciem zasilania. Nie należy zapominać o założeniu zworki BOOT0 w pozycję 1 oraz zworki BOOT 1 w pozycję 0. Po wykonaniu tych czynności i podłączeniu napięcia zasil-



Fot. 1.

lania mikrokontroler powinien wystartować z pamięci systemowej.

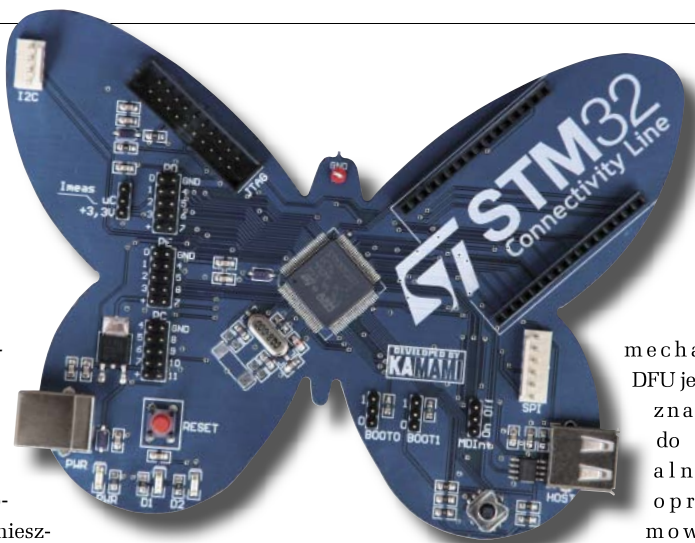
Jak już wcześniej wspomniano, programowanie z wykorzystaniem mechanizmu DFU okupione jest dość sporym nakładem pracy. Pierwszą czynnością jest ściągnięcie i zainstalowanie aplikacji DfuSe (<http://www.st.com/stonline/products/support/micro/files/um0412.zip>). W katalogu z aplikacją znajduje się sterownik klasy DFU, który musi zostać zainstalowany, aby możliwa była komunikacja aplikacji z mikrokontrolerem. W tym celu należy połączyć płytke STM32 Butterfly przewodem typu USB A-A z komputerem PC poprzez złącze HOST, umieszczone na prawym skrzydle motylka. System Windows powinien wykryć nowe urządzenie klasy DFU i rozpocząć proces instalacji sterowników (fot. 2). Po wskazaniu kreatorowi, aby automatycznie wyszukał odpowiedni sterownik (fot. 3), kreator powinien odnaleźć sterownik i go zainstalować. W trakcie instalacji może pojawić się monit o wskazanie pliku STTub30.sys (należy wskazać plik w podkatalogu odpowiednim dla posiadanej wersji systemu operacyjnego), a także informacja o tym, że sterownik nie został cyfrowo podpisany (fot. 4). Proces instalacji sterownika powinien się zakończyć oknem informującym o pomyślnym zakończeniu procesu instalacji (fot. 5). Po zainstalowaniu sterownika należy uruchomić program *Dfu-*

Se Demonstration (fot. 6). Program powinien wykryć mikrokontroler umieszczony na płytce STM32 Butterfly jako „STM Device in DFU mode”. W polu „DFU Mode” wyświetlane są dane identyfikujące urządzenie USB: numery Vendor ID, Product ID oraz Version. Mikrokontroler STM32F107VBT6 posiada dwa możliwe do zaprogramowania obszary: *Internal Flash* reprezentujący pamięć Flash oraz *Option Bytes* reprezentujący bajty konfiguracyjne. W celu zaprogramowania pamięci konieczne jest przygotowanie pliku *.dfu.

Plik *.dfu oprócz danych przeznaczonych do zaprogramowania zawiera również dane identyfikujące wersję aplikacji urządzenia, dla którego plik jest przeznaczony. Ponieważ

mechanizm DFU jest przeznaczony do uaktualniania oprogramowania, zwykle stanowi on do- datek do oprogramowania umieszczonego w urządzeniu, które wykorzystuje interfejs USB do innych celów. Wykorzystując mechanizm DFU do programowania pamięci mikrokontrolerów STM32, w rzeczywistości nie uaktualniamy oprogramowania, tylko wgrujemy jego „pierwszą” wersję, tak więc dane identyfikujące urządzenie oraz wersję zainstalowanego oprogramowania nie mają znaczenia.

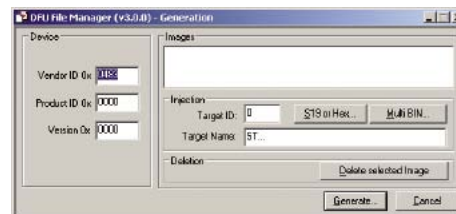
Do przygotowywania plików *.dfu służy program *DFU File Manager* (fot. 7). Pierwszą czynnością, jaką należy wykonać po uruchomieniu programu, jest wybór operacji, jaką zamierzamy wykonać. Do wyboru mamy utworzenie pliku *.dfu z plików binarnych oraz odtworzenie plików binarnych z pli-



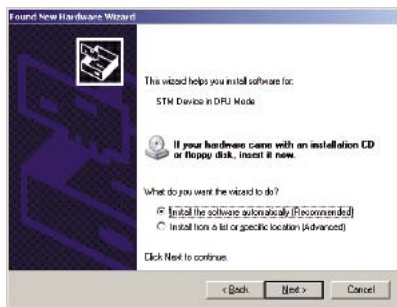
Fot. 2.



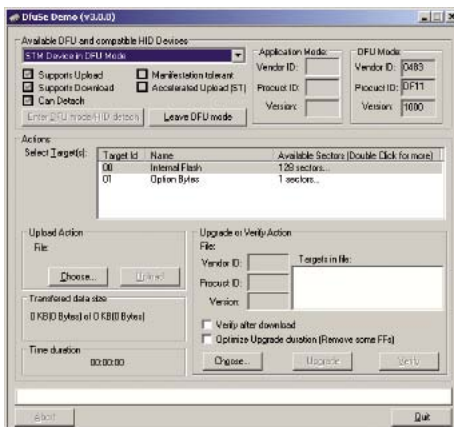
Fot. 5.



Fot. 8.



Fot. 3.



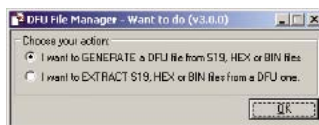
Fot. 6.



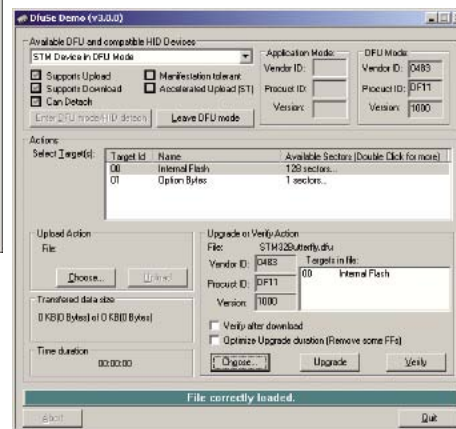
Fot. 9.



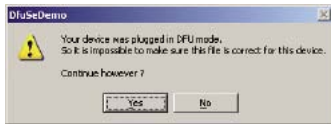
Fot. 4.



Fot. 7.

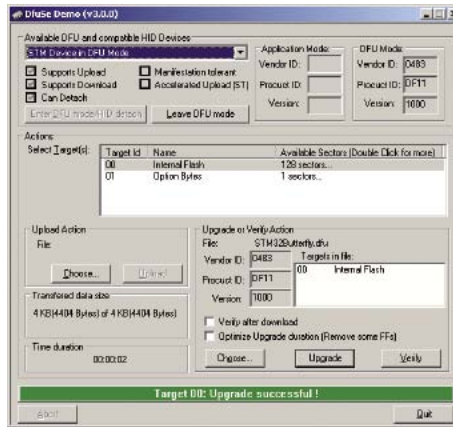


Fot. 10.



Fot. 11.

ku *.dfu. Nas interesuje oczywiście opcja „I want to GENERATE a DFU file...”. Po jej wybraniu wyświetlone zostanie okno służące do wprowadzenia parametrów tworzonego pliku DFU (fot. 8). Numery Vendor ID, Product ID oraz Version są nieistotne, pozostawmy je więc niezmienione. W zasadzie jedynym istotnym parametrem jest parametr Target ID, który w przypadku wewnętrznej pamięci Flash mikrokontrolera ma wartość 0. Zaletą mechanizmu DFU jest możliwość umieszczenia w jednym pliku DFU danych dla kilku różnych pamięci, w tym również pamięci zewnętrznych podłączanych np. poprzez magistrale szeregowo. Klikając przycisk „S19 or hex”, należy wskazać plik zawierający kod przeznaczony do zaprogramowania pamięci mikrokontrolera. Po wskazaniu pliku z kodem należy kliknąć przycisk „Generate” oraz wskazać lokalizację i nazwę wyjściowego pliku *.dfu. Pomyślnie zakończony proces generowania pliku zostanie zasygnalizowany odpowiednim komunikatem (fot. 9).



Fot. 12.

Po wygenerowaniu pliku *.dfu należy ponownie uruchomić program DfuSe Demonstration, a następnie załadować wygenerowany uprzednio plik *.dfu (fot. 10). Po poprawnym załadowaniu pliku *.dfu wyświetlone zostaną informacje na temat jego zawartości. W celu zaprogramowania pamięci Flash należy kliknąć przycisk „Upgrade”. Wyświetlony zostanie komunikat (fot. 11), informujący o braku możliwości sprawdzenia, czy plik *.dfu wgrany jest do właściwego urządzenia. Należy potwierdzić chęć kontynuacji procesu wgrzywania poprzez kliknięcie przycisku

„Yes”. Pomyślnie wgranie programu do pamięci Flash zostanie zasygnalizowane odpowiednim komunikatem (fot. 12).

Każdorazowe przygotowywanie pliku *.dfu z pliku *.hex jest dość kłopotliwe i czyni metodę programowania pamięci z wykorzystaniem mechanizmu DFU mało przydatną na etapie tworzenia programu. Jak już wcześniej wspomniano, mechanizm DFU został opracowany z myślą o prostym uaktualnieniu oprogramowania w działających urządzeniach i na etapie produkcyjnym jego przydatność jest mocno dyskusyjna. Programowanie z wykorzystaniem DFU traktować należy jako rozwiązanie „awaryjne” w przypadku braku możliwości zaprogramowania pamięci inną, bardziej wygodną metodą programowania pamięci.

Można się również pokusić o zautomatyzowanie procesu tworzenia pliku *.dfu poprzez stworzenie aplikacji automatycznie tworzącej plik *.dfu lub też stworzyć aplikację wgrywającą program wprost z pliku *.hex. Pomocne w tym będą źródła programów *Dfu-Se Demonstrator* oraz *Dfu File Manager*, które dostępne są w podkatalogu Sources katalogu instalacyjnego oprogramowania DfuSe.

Radosław Kwiecień, EP
radoslaw.kwiecien@ep.com.pl

R E K L A M M A

Altium Designer

**Zostań Pionierem!
Wyrzędź Pozostałych**

Altium oferuje narzędzia, które ułatwiają realizację złożonych projektów urządzeń elektronicznych. Otrzymujesz najnowsze technologie i cały potencjał, abyś mógł swobodnie realizować swoje pomysły.

Teraz oferujemy większe możliwości za niższą cenę. Sprawdź nasze promocje.

