

*prototypowanie układów, zestaw uruchomieniowy,  
mikrokontroler 32-bitowy, mikrokontroler ARM*

Krzysztof P. DYRCZ\*, Marcin SKÓRA\*\*

## **SZYBKIE PROTOTYPOWANIE UKŁADÓW STEROWANIA Z WYKORZYSTANIEM NOWOCZESNYCH MIKROPROCESOROWYCH ZESTAWÓW URUCHOMIENIOWYCH**

W artykule przedstawiono możliwości wykorzystania nowoczesnych zestawów uruchomieniowych z 32-bitowymi, wydajnymi mikrokontrolerami z rdzeniem ARM w procesach szybkiego prototypowania układów sterowania. W sposób zwięzły opisano nowoczesne narzędzia programistyczne, dedykowane do zastosowań z wymienionymi zestawami, umożliwiające szybkie napisanie kodu programu oraz uruchomienie i przetestowanie aplikacji w układzie docelowym. Zaprezentowano przykładową aplikację generatora PWM, napisaną z wykorzystaniem jednej z omawianych technik programowania.

### 1. WSTĘP

Projektowanie układów sterowania związane jest niejednokrotnie z budowaniem układów prototypowych, które umożliwiają uruchomienie i przetestowanie algorytmów sterowania przed ostateczną implementacją w urządzeniu docelowym. W przypadku systemów, które nie mają złożonej struktury i realizują stosunkowo proste zadania, zwykle wykonuje się prototypowy, mikroprocesorowy układ elektroniczny, który programuje się wykorzystując ogólnie dostępne narzędzia programistyczne. Bardziej złożone układy, realizujące skomplikowane algorytmy sterowania i wymagające mikrokontrolerów o znacznych mocach obliczeniowych, najczęściej budowane są na etapie prototypu z wykorzystaniem dostępnych zestawów uruchomieniowych, spełniających wymogi technologii nazywanej szybkim prototypowaniem (ang. *Rapid*

---

\* Politechnika Wroclawska, Instytut Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-372 Wrocław, e-mail: krzysztof.dyrcz@pwr.wroc.pl

\*\* Politechnika Wroclawska, Instytut Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-372 Wrocław, e-mail: marcin.skora@pwr.wroc.pl

*Prototyping*). Aby możliwe było maksymalne skrócenie czasu projektowania i testowania budowanego układu, wraz z odpowiednim systemem mikroprocesorowym należy wykorzystać dedykowane środowisko programistyczne, które zapewni przede wszystkim wydajne tworzenie algorytmu sterującego, ale również odpowiednio komfortowe uruchamianie, testowanie i poprawianie ewentualnych błędów w kodzie programu.

Obecnie istnieje kilka wydajnych systemów szybkiego prototypowania układów sterowania, m.in. NI PXI Platform [4], xPC Target [10], DSP2 [8]. Jednak jednym z najczęściej używanych są produkty firmy dSPACE, w tym popularne karty DS1103 i DS1104 [1], [2], [13]. Ich zaletą jest niewątpliwie łatwa integracja sprzętowa z układami otoczenia, uzyskana przez zastosowanie specjalnych paneli połączeniowych oraz od strony programistycznej, rozbudowane środowisko ControlDesk, wyposażone w wydajne kompilatory języka C/C++, interpreter języka Python oraz narzędzia do wizualizacji wybranych zmiennych. Dodatkową cechą, niejednokrotnie decydującą o wyborze właśnie tego systemu, jest łatwa i niezawodna integracja z jednym z najbardziej popularnych pakietów obliczeniowych Matlab/Simulink, pozwalająca na generowanie kodu sterującego bezpośrednio z modelu obiektu zbudowanego w Simulinku, praktycznie bez znajomości żadnego z wymienionych wcześniej języków programowania. Podejście takie obarczone jest jednak dużymi kosztami, ponieważ obecna cena systemu dSpace wynosi (w zależności od modelu i wyposażenia) od kilkunastu do kilkudziesięciu tysięcy złotych.

Szybki, ciągle postępujący rozwój techniki mikroprocesorowej spowodował, że dostępne są obecnie nieporównywalnie tańsze, niejednokrotnie o zbliżonej wydajności, mikroprocesorowe zestawy uruchomieniowe. Spośród dostępnych wielu rodzin mikrokontrolerów, na szczególną uwagę zasługują dwie: 32-bitowe stałoprzecinkowe mikrokontrolery sygnałowe rodziny C2000 firmy Texas Instruments [3], [12] oraz 32-bitowe mikrokontrolery z rdzeniem ARM [7], [9], [15]. Obydwa rozwiązania charakteryzują się przede wszystkim dużą wydajnością obliczeniową przy umiarkowanej cenie, jednak ze względu na większą różnorodność gotowych układów rozwiązania z mikrokontrolerami ARM zaczynają być używane coraz częściej. Jedne z najbardziej popularnych obecnie systemów uruchomieniowych wykorzystujących te mikrokontrolery, to przede wszystkim moduły serii Discovery firmy ST Microelectronics, których cena waha się od kilkudziesięciu do kilkuset złotych. Dzięki modułowej budowie i łatwemu dostępowi do zasobów sprzętowych mikrokontrolera, zestawy te można łatwo integrować z projektowanym układem. Dużą zaletą tych układów jest także wyposażenie ich (w zależności od wersji) w różnego rodzaju czujniki MEMS (m.in. żyroskopy, trójosiowe akcelerometry, czujniki temperatury) oraz wbudowany programator/JTAG, co do tej pory było rzadko spotykane w tej klasie układów. Na szczególną uwagę zasługuje także sposób programowania – oprócz dostępnych, klasycznych narzędzi programistycznych, umożliwiających pisanie programu w języku C/C++ zestawy Discovery mogą być programowane z użyciem nowoczesnych, graficznych środowisk programistycznych, także z wykorzystaniem dedykowanych bibliotek (popularnie nazywanych toolboxami) w pakiecie Matlab/Simulink.

Należy przy tym zaznaczyć, że obok profesjonalnych, płatnych pakietów programistycznych (kompilatorów i środowisk IDE, m. in. Keil  $\mu$ Vision, Atollic Studio), można używać także narzędzi bezpłatnych, które dorównują im możliwościami (np. środowisko Eclipse z kompilatorem GNU-GCC).

Takie cechy powodują, że omawiane systemy spełniają wymagania technologii szybkiego prototypowania, a atrakcyjna cena zestawu w połączeniu z wysoką wydajnością obliczeniową mikrokontrolerów z rdzeniem ARM i łatwością ich programowania przyczyni się do coraz szerszego ich wykorzystania.

W kolejnych rozdziałach artykułu w sposób zwięzły opisano najbardziej popularne, dostępne obecnie narzędzia programistyczne wykorzystywane do programowania mikrokontrolerów ARM oraz zaprezentowano przykładowy system generatora PWM zbudowany za pomocą niekomercyjnych bibliotek dostępnych w pakiecie Matlab/Simulink.

## 2. PRZYKŁADOWE ROZWIĄZANIA PROGRAMISTYCZNE

### 2.1. WSTĘP

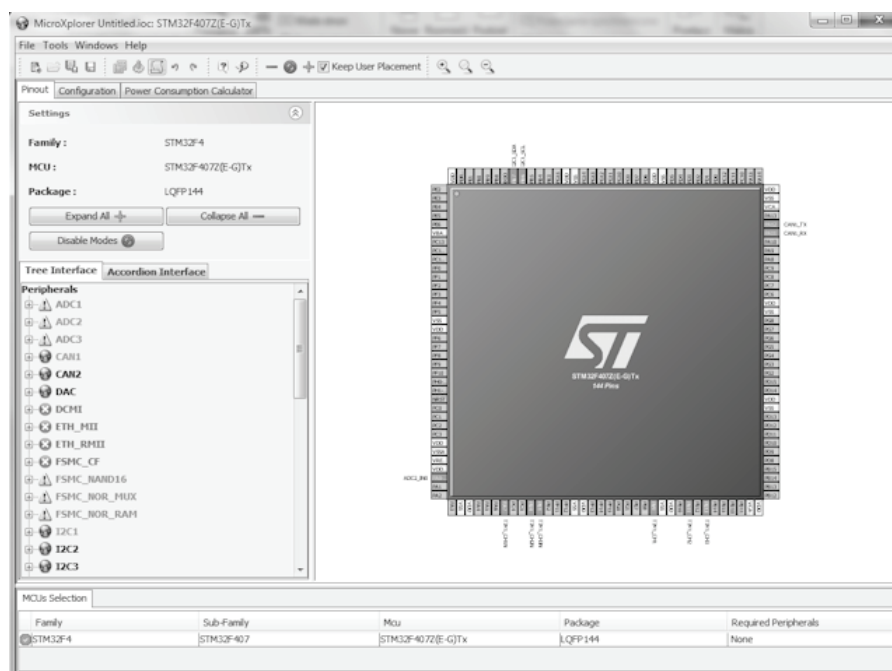
Rosnąca popularność zestawów uruchomieniowych z mikrokontrolerami ARM spowodowała, że obecnie dostępnych jest kilka rozwiązań wydajnych środowisk programistycznych o porównywalnych możliwościach. Wybór określonego środowiska z reguły podyktowany jest subiektywnymi ocenami i przyzwyczajeniami programisty, jednak często warunkuje późniejszy sposób i styl programowania.

W klasycznym podejściu, program sterujący pisany jest w wybranym języku programowania z wykorzystaniem jednego z dostępnych pakietów programistycznych. Takie podejście wymaga od programisty dobrej znajomości wybranego języka programowania (najczęściej C/C++), architektury procesora oraz opanowania technik optymalnego konstruowania struktur programów. Dodatkowo, czas pisania programu jest relatywnie długi, ponieważ najczęściej, obok wykorzystania gotowych bibliotek, należy zbudować własne, zoptymalizowane pod kątem wykorzystania w projektowanej strukturze. Innym, coraz częściej wykorzystywanym sposobem programowania jest wygenerowanie kodu programu z modelu obiektu stworzonego za pomocą graficznych bibliotek, kompatybilnych z jednym z najbardziej popularnych obecnie programów symulacyjnych Matlab/Simulink. Jest to szybka i skuteczna metoda, która zaczyna być coraz częściej stosowana w nowoczesnym prototypowaniu układów sterowania wykorzystujących mikroprocesorowe zestawy uruchomieniowe. Przykładowe rozwiązanie tego typu przedstawiono w rozdziale 4. Istnieje jeszcze trzeci sposób programowania, łączący wymienione techniki. Polega on na wygenerowaniu tylko fragmentu kodu programu za pomocą modelu Matlab/Simulink i wykorzystywanie go, jako dołączalnej biblioteki, w projekcie pisany metodą tradycyjną. Biblioteki pro-

gramu Simulink mogą być w takim podejściu użyte również do zbudowania interfejsu użytkownika, co zaprezentowano m.in. w [11].

## 2.2. MICROXPLOER – NARZĘDZIE DO KONFIGURACJI MIKROKONTROLERÓW ARM

Jednym z podstawowych problemów, pojawiających się podczas pisania programów dla nowoczesnych, rozbudowanych mikrokontrolerów jest umiejętne skonfigurowanie zasobów sprzętowych układu. Ze względu na coraz większą liczbę zadań przez nie realizowanych, a jednocześnie ograniczoną liczbę wyprowadzeń, powszechne jest stosowanie tzw. funkcji alternatywnych, współdzielących wybrane zasoby portów we/wy mikrokontrolera. Aby ułatwić konfigurację sprzętową układu, można posłużyć się narzędziem MicroXplorer, pokazanym na rysunku 1. Dzięki jego zastosowaniu, wygenerowanie plików konfiguracyjnych projektu, zawierających deklaracje użytych zasobów sprzętowych mikrokontrolera i sposobów ich wykorzystania, sprowadza się do wybrania typu układu, zaznaczenia wybranych funkcji i kliknięcia w przycisk „Generate source files”, znajdujący się na pasku narzędzi. W następnym kroku pozostaje uzupełnić tak stworzony program o procedury i funkcje właściwe dla projektowanego układu. Dodatkowo, można wygenerować zbiorczy raport użytych zasobów mikrokontrolera, co ułatwia późniejsze tworzenie dokumentacji programu.

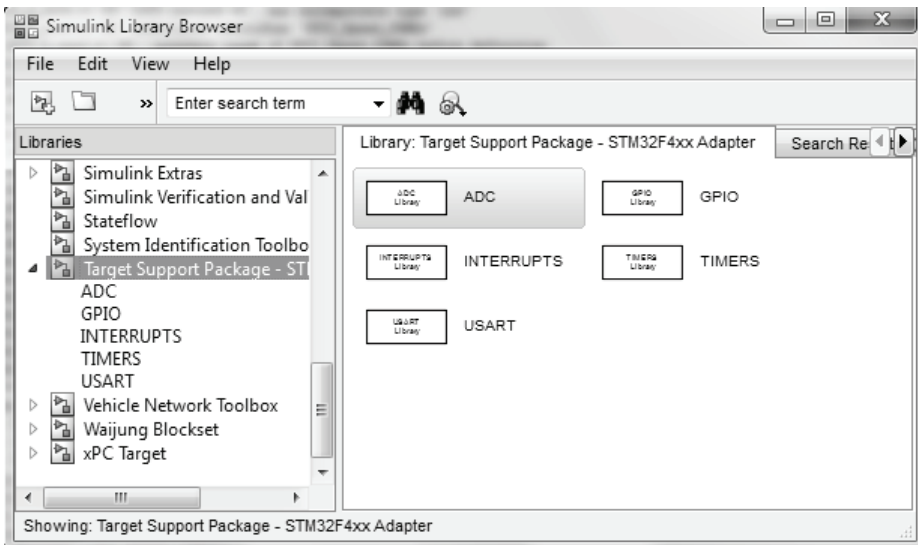


Rys. 1. Widok okna głównego programu MicroXplorer

### 2.3. DEDYKOWANA BIBLIOTEKA STM32-MAT FIRMY ST MICROELECTRONICS

Jedną z używanych obecnie metod programowania zestawów uruchomieniowych Discovery jest użycie pakietu Matlab/Simulink z dedykowaną biblioteką (toolboxem) STM32-MAT. Biblioteka nie wyczerpuje wszystkich możliwości mikrokontrolera, jednak do wygenerowania podstawowego programu sterującego jej zasoby są wystarczające. Należy wziąć pod uwagę również fakt, że biblioteka jest ciągle rozwijana i z pewnością w następnych wydaniach dostępna ilość wbudowanych komponentów ulegnie zwiększeniu. Biblioteka jest udostępniona na stronach internetowych firmy ST Microelectronics pod nazwą STM32-Mat, a po zainstalowaniu w programie Matlab widoczna jest jako toolbox Simulinka pod nazwą STM32-F4xx Adapter. Zawartość biblioteki pokazano na rysunku 2.

Jak wskazuje nazwa biblioteki, jej zawartość ułatwia programowanie 32-bitowych mikrokontrolerów rodziny STM32-F4xx z wydajnym rdzeniem Cortex. Są to najbardziej zaawansowane technologicznie i najbardziej wydajne mikrokontrolery tej rodziny, dodatkowo coraz częściej stosowane, w związku z tym budowanie biblioteki Matlab/Simulink dla tych układów wydaje się uzasadnione.

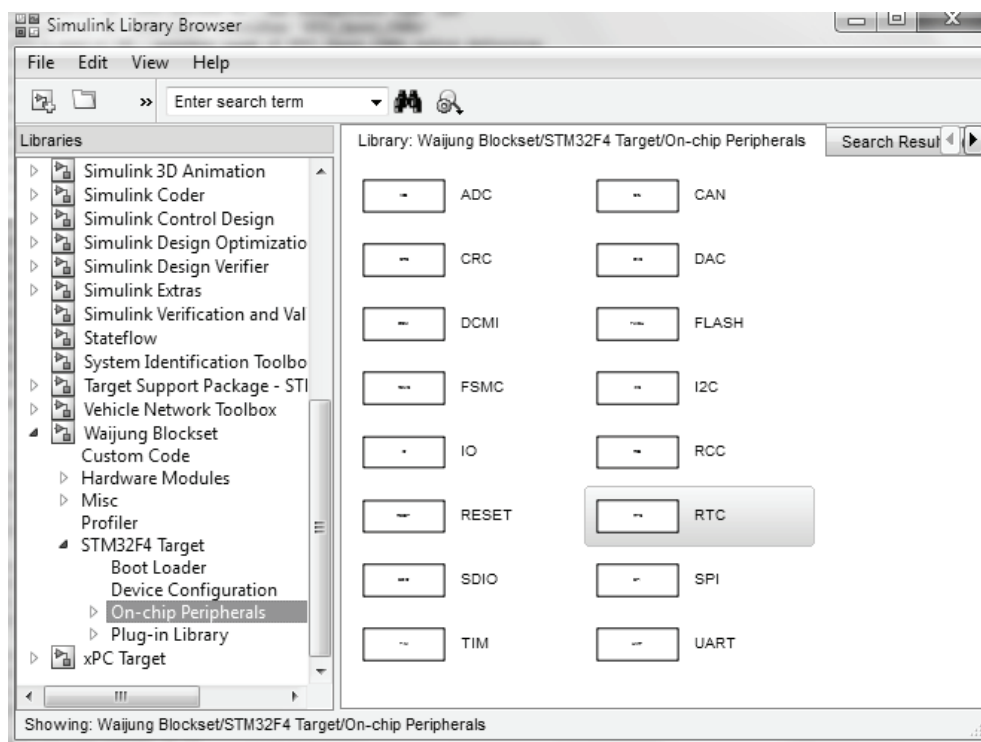


Rys. 2. Zawartość biblioteki STM32-F4xx Adapter w programie Matlab/Simulink

### 2.4. DEDYKOWANA BIBLIOTEKA WAIJUNG BLOCKSET

Oprócz oficjalnej biblioteki oferowanej przez producenta układów, na uwagę zasługują również biblioteki oferowane przez niezależnych producentów. Szczególnie przydatna okazuje się biblioteka Waijung Blockset, która obecnie oferuje szersze

wsparcie dla wykorzystywanych zasobów mikrokontrolera niż omówiona w p. 2.3 biblioteka STM32–Mat. Ponadto, należy zaznaczyć, że Waijung Blockset udostępniany jest również bezpłatnie dla programów testowych. Zawartość biblioteki z katalogu wbudowanych układów wewnętrznych mikrokontrolera pokazano na rysunku 3. Pozwala ona na obsługę z poziomu Simulinka modułów standardowych (porty we/wy, przetworniki ADC i DAC, podstawowe i zaawansowane układy czasowo-licznikowe, komunikacja szeregową UART, SPI, I2C), jak i dedykowanych (moduł obliczający sumę kontrolną CRC, moduł do obsługi pamięci – FMSC, kart SD, kamer – DCMI i inne). Elastyczność używania biblioteki jest zapewniona dzięki możliwości dodawania własnego kodu, w tym obsługi przerwań. Dodatkowo dostępne są bloki do obsługi zewnętrznych modułów – wybranych czujników, wyświetlacza LCD, itp.

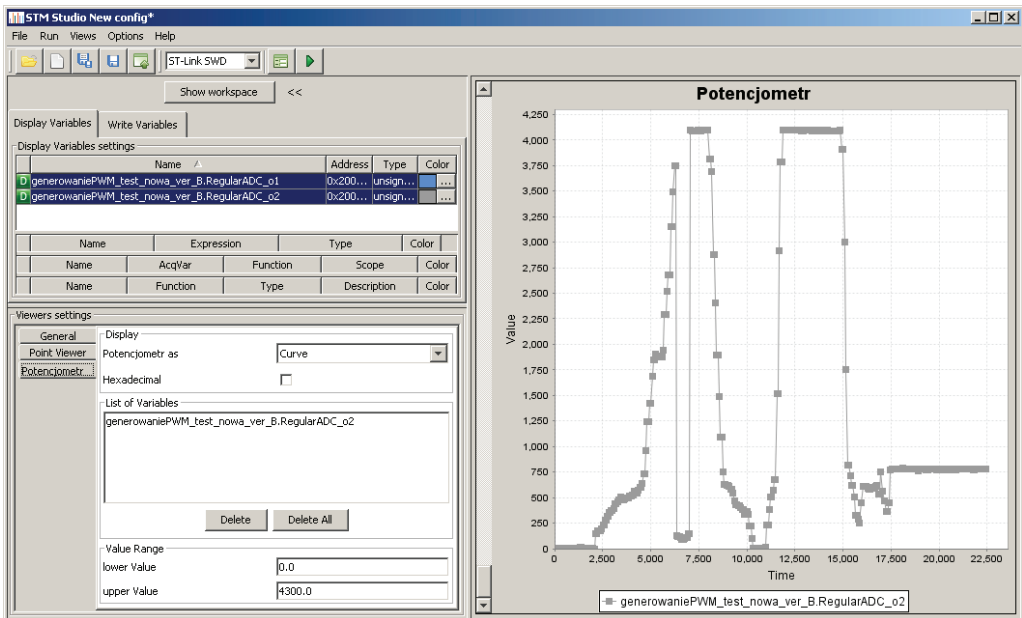


Rys. 3. Zawartość biblioteki Waijung Blockset

## 2.5. URUCHAMIANIE I DEBUGOWANIE PROGRAMU W ŚRODOWISKU STM STUDIO

Omówione wcześniej programy wspomagają programistów na wstępnym etapie konfiguracji zasobów mikrokontrolera, a także ułatwiają generowanie kodu źródłowego z graficznego modelu obiektu zapisanego w środowisku Simulinka. Ostatnim eta-

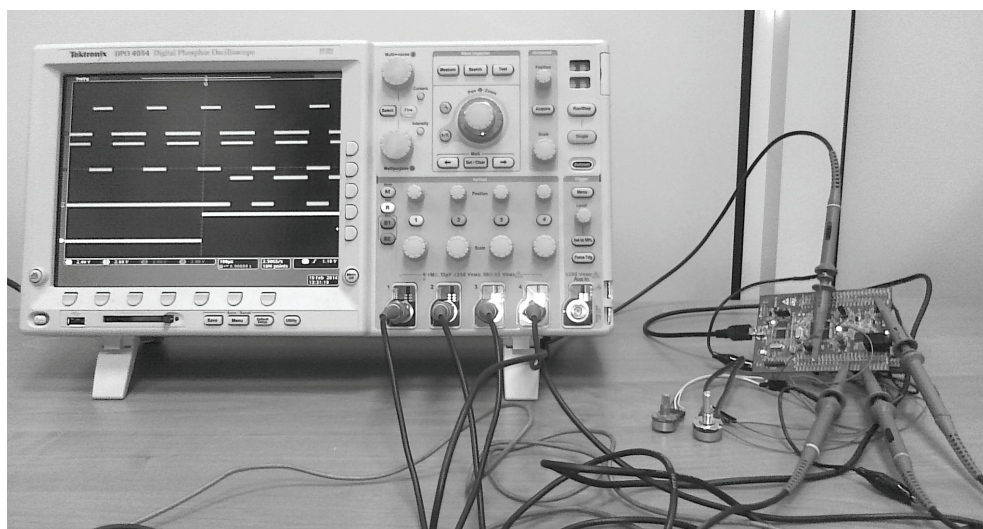
pem każdego projektu jest wgranie programu do pamięci mikrokontrolera i przetestowanie poprawności jego działania. Dotychczas niedostępne były żadne dedykowane narzędzia wspomagające ten etap procesu projektowania systemu, a tworzenie jakiegokolwiek interfejsu wymagało napisania odrębnego oprogramowania do śledzenia zawartości wybranych zmiennych np. z wykorzystaniem języka skryptowego LUA [5] lub pakietu Matlab/Simulink [6] z wykorzystaniem łącza szeregowego, co obarczone jest ograniczeniami związanymi z osiąganymi prędkościami transmisji Sytuacja zmieniła się po udostępnieniu przez firmę ST Microelectronics oprogramowania STM Studio. Jest to program narzędziowy, który dzięki wykorzystaniu odpowiedniego interfejsu umożliwia debugowanie i diagnostykę programów pisanych na mikrokontrolery STM8 i STM32 w czasie rzeczywistym, poprzez wyświetlanie wartości zmiennych z pamięci RAM. Program umożliwia komunikację z mikrokontrolerami STM8 za pomocą programatora/debugera ST-LINK, RLink lub STICE a z mikrokontrolerami STM32 – poprzez ST-LINK, zarówno w trybie JTAG, jak i SWD. Ważnymi zaletami programu są przede wszystkim: możliwość konfiguracji częstotliwości próbkowania przy akwizycji zmiennych, możliwość wyboru trybu akwizycji zsynchronizowanego z układem docelowym, a także możliwość akwizycji tylko tych danych, które są używane przez widoczne okna przeglądania. Wygląd głównego okna programu STM Studio z przykładowej aplikacji pokazano na rysunku 4.



Rys. 4. Przykładowe okno pakietu STM Studio

### 3. PRZYKŁADOWA APLIKACJA

Do realizacji przykładowej aplikacji wybrano płytkę startową STM32F4-Discovery z mikrokontrolerem ARM z rodziny Cortex-M4F, pokazaną na rysunku 5 w zestawie z sondami pomiarowymi i oscyloskopem. Celem projektu było wygenerowanie sygnałów PWM, parami komplementarnych względem siebie, z czasem martwym, z regulowanymi współczynnikami wypełnienia (np. za pomocą potencjometrów obsługiwanych przez wbudowane przetworniki analogowo-cyfrowe), z sygnałem zezwalającym na generację przebiegu. Dodatkowym wymaganiem była możliwość zdalnego rejestrowania stanu pracy układu, np. poprzez interfejs USART i sygnalizacja przekroczenia pewnych poziomów odczytywanego napięcia. program spełniający tak postawione wymagania może być użyty w realizacji układu sterowania prędkością silnika prądu stałego za pomocą mostka H, z elementami interfejsu użytkownika.



Rys. 5. Platforma testowa z oprzyrządowaniem

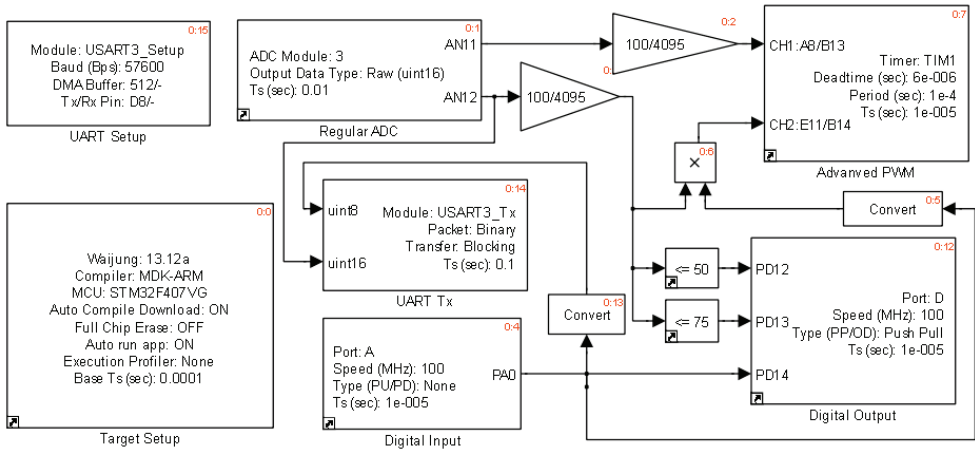
Rozwiązanie postawionego zadania wymagało doboru odpowiednich narzędzi programowych. Wybrane zostały:

1. kompilator MDK-ARM,
2. program STM32 ST-LINK Utility do programowania procesora,
3. pakiet Matlab/Simulink jako platforma szybkiego oprogramowania,
4. biblioteka Waijung Blockset w wersji 13.12a.

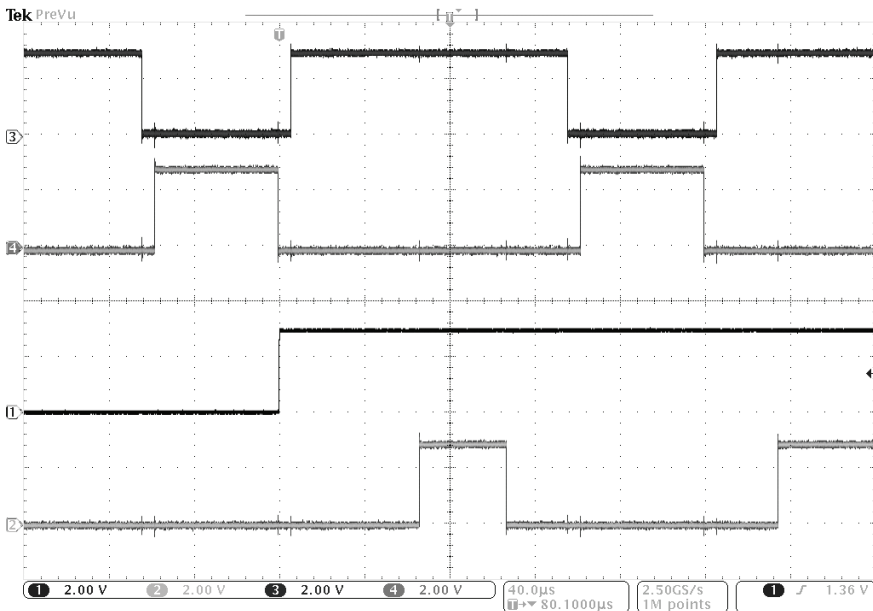
Na rysunku 6 przedstawiono schemat blokowy struktury zrealizowany w Simulinku. Warto zwrócić uwagę, że poszczególne bloki mogą wykonywać się z różną rozdzielczością czasową. Z kolei na rysunku 7 umieszczono przebiegi wygenerowane



przez program testowy, zarejestrowane oscyloskopem, przy czym: kanał 1 – sygnał zezwalający (wyprowadzenie PA0), kanał 2 – generowany sygnał PWM (wyprowadzenie PE11), kanały 3 i 4 – komplementarne sygnały PWM z czasem martwym 6  $\mu$ s (wyprowadzenia PA8, PB13).



Rys. 6. Przykład programu do generacji sygnału PWM i obsługi portów wejścia wyjścia (Wąjung Blockset i STM32F4 Target)



Rys. 7. Wygenerowane przebiegi komplementarnego sygnału PWM z czasem martwym 6  $\mu$ s

Przedstawiony program sterujący ma charakter demonstracyjny, jednak użyty dodatek do pakietu Matlab może być z powodzeniem wykorzystany do realizacji praktycznego, bardziej złożonego projektu. Przykładem profesjonalnego zastosowania biblioteki Waijung Blockset może być jedna z wersji oprogramowania prototypowego wózka inwalidzkiego [14], którego układ sterowania zbudowany został z wykorzystaniem 32-bitowego mikrokontrolera z rodziny Cortex-M4F firmy ST Microelectronics. Dla użytkownika takiego pojazdu nie miało znaczenia, w jaki sposób powstał program, natomiast dla programisty – wykorzystanie pakietu Matlab pozwoliło uprościć niektóre zadania – obliczenia, przeglądanie tablic ze stałymi, itp.

#### 4. WNIOSKI

W referacie opisano możliwości zastosowania tanich, nowoczesnych zestawów uruchomieniowych z 32-bitowymi mikrokontrolerami z rdzeniem ARM w szybkim prototypowaniu układów sterowania. Ze względu na dużą wydajność obliczeniową, mikrokontrolery o takiej architekturze z powodzeniem mogą być używane już nie tylko w najprostszych zastosowaniach, ale także w aplikacjach, które do niedawna możliwe były do zrealizowania jedynie przy zastosowaniu wydajnych obliczeniowo, lecz drogich zestawów z procesorami sygnałowymi lub PowerPC.

Omawiane rozwiązania wykorzystują pakiety programistyczne nowych generacji, które umożliwiają wydajne i szybkie tworzenie aplikacji (np. Keil  $\mu$ Vision, Atollic Studio, darmowe pakiety na licencji GNU, ST MicroXplorer, STM Studio) oraz posiadają możliwość programowania z wykorzystaniem bibliotek instalowanych w programie Matlab/Simulink.

#### LITERATURA

- [1] ABBOU A., NASSER T., MAHMOUDI H., AKHERRAZ M., ESSADKI A., *Induction Motor controls and Implementation using dSPACE*, WSEAS Transaction on Systems and Control, Issue 1, Vol. 7, 2012, 26–35.
- [2] AMAMRA S., BARAZANE L., BOUCHERIT M. S., CHERIFI A., *Inverse Fuzzy Model Control for a Speed control Induction Motor Based dSPACE Implementation*, Modern Electric Power Systems 2010, Wrocław, Poland, 1–5.
- [3] CHEN X.M., GONG X.L., ZHOU H.X., XU Z.B., XU Y.G., KANG C.J., *An Economical Rapid Control Prototyping System Design with Matlab/Simulink and TMS320F2812 DSP*, Proceedings of the International Multiconference of Engineers and Computer Scientists, 2010, Vol. II, IMECS 2010, Hong Kong, 1–6.
- [4] DASE C., FALCON J.S., MACCLEERY B., *Motorcycle Control Prototyping Using an FPGA-Based Embedded Control System*, EEE Control Systems Magazine, 09.2006, 11–17.
- [5] DYRCZ K. P., FAŚCISZEWSKI M., *Stanowisko do badania metod sterowania silnikami z magnesami trwałymi wykorzystujące mikrokontroler ARM*, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej, nr 64, 2010, 355–366.

- [6] GRYGIEL R., PACHOLCZYK M., *Prototypowanie algorytmów sterowania w systemie Matlab/Simulink*, Elektryka, 2010, Zeszyt nr 1, Wyd. Instytutu Automatyki, Politechnika Śląska w Gliwicach, 149–162.
- [7] GRZESZCZYK R., HOJKA A., MERKISZ J., BAJERLEIN M., FUC P., LIJEWSKI P., BOGUS P., *Estimation of In-Use Powertrain Parameters of Fully Electric Vehicle Using Advanced ARM Microcontrollers*, Advanced Microsystems for Automotive Applications 2012, G. Meyer (ed.), Springer, 2012, 157–164.
- [8] HERCOG D., CURKOVIC M., EDELBAHER G., URLEP E., *Programming of the DSP2 board with the Matlab/Simulink*, IEEE International Conference on Industrial Technology, 2003, 709–713.
- [9] LOVAS I., *Low Cost Universal Motor Drive Using Kinetis L family*, Freescale Semiconductor Document Number AN4609, Application Note, 10.2012, 1–11.
- [10] MOSTERMAN P.J., PRABHU S., DOWD A., GLASS J., ERKKINEN T., KLUZA J., SHENOY R., *Embedded Real-Time Control via MATLAB, Simulink, and xPC Target*, Handbook of Networked and Embedded Control Systems, Control Engineering, 2005, 419–446.
- [11] NOWOPOLSKI K., *Implementation of ball-and-beam control system as an instance of Simulink to 32-bit microcontroller interface*, Poznań University of Technology, Academic Journals, Electrical Engineering, Issue 76/2013, 31–38.
- [12] RUSU C., BARA A., ENIKO B., *Embedded target toolbox for DSP control applications of BLDC motor*, Journal of Computer Science and Control Systems, 01/2009, 115–119.
- [13] SIVACHANDRAN P., VENKATESH P., BALAMURUGAN S., *A new real time approach using dSPACE R&D Controller Board for reactive power control by SVC in autonomous wind-diesel hybrid power systems*, International Journal of Engineering, Science and Technology, Vol. 3, No. 5, 2011, 30–45.
- [14] SKÓRA M., PAWLAK M., *Zastosowanie czujników MEMS do sterowania napędu elektrycznego wózka inwalidzkiego*, Przegląd Elektrotechniczny, R. 89, nr 12/2013, 133–137.
- [15] VIJAYALAKSHMI S., *Vehicle control system implementation Using CAN protocol*, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 6, June 2013, 2532–2538.

#### RAPID PROTOTYPING OF CONTROL SYSTEMS USING MODERN EVALUATION TOOLS

The paper presents the possibilities of using modern kits with high-performance 32-bit ARM microcontrollers in the process of rapid prototyping control systems. Compactly describes the modern development tools, dedicated for use with these sets, enabling fast writing program code and run and test applications on the target system. PWM generator sample application, written using one of these programming techniques was presented.