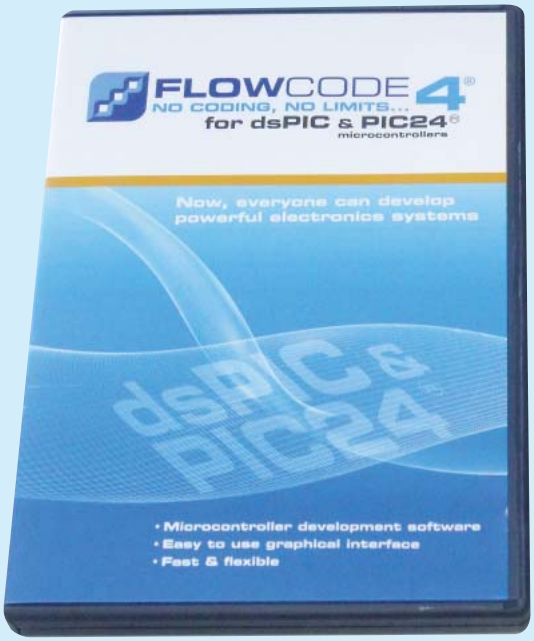
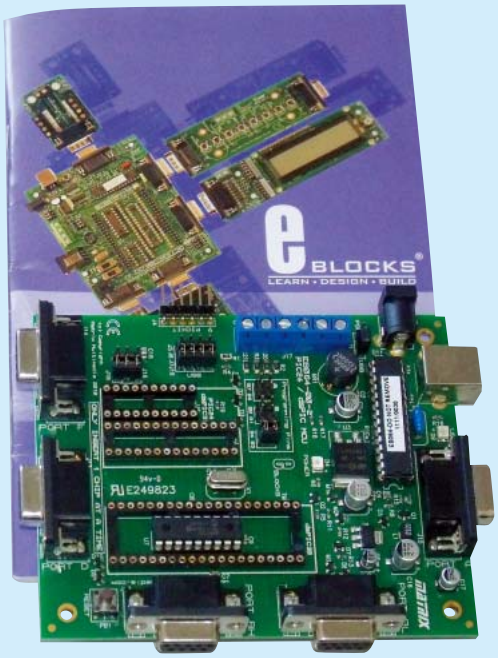




FlowCode i E-blocks (1)

Instalowanie komponentów, pierwszy program

Współczesne języki programowania ewoluują w stronę, która umożliwi korzystanie z kompilatora i mikrokontrolera nie tylko elicie programistów, ale dosłownie każdemu. Pozwolą na to języki do programowania graficznego, których współcześnie używają z powodzeniem zarówno profesjonaliści z różnych dziedzin, jak i… dzieci programujące klocki Lego Mindstorm. Przykładem środowiska programistycznego przeznaczonego zarówno dla profesjonalistów, jak i amatorów, służącego do programowania graficznego różnych rodzin mikrokontrolerów, jest produkt brytyjskiej firmy Matrix Multimedia – FlowCode. Nie sposób przejść obok niego obojętnie.

Oferta firmy Matrix Multimedia obejmuje nie tylko kompilator, ale również system dobrze przemyślanych „klocków" dla elektroników o nazwie handlowej *E-blocks*. Jest to zestaw zmontowanych płytek drukowanych – zestawów ewaluacyjnych innych niż wszystkie. Zwykle bowiem wyobrażamy sobie taki zestaw jako płytkę drukowaną z wlutowanym lub umieszczonym w podstawce mikrokontrolerem czy innym układem nadrzędnym, złączami goldpin lub polami lutowniczymi służącymi do dołączenia płytki do układów peryferyjnych oraz wlutowanego na stałe wyposażenia dodatkowego, takiego jak wyświetlacz, klawisze, diody LED itp. To dosyć wygodne rozwiązanie, stosowane przez wielu producentów, jednak niezbyt trwałe. Takie płytki po pewnym czasie najczęściej trafiają do zasobów elektronicznego złomu na skutek uszkodzenia szpilek goldpinów lub pól lutowniczych.

E-blocks

Firma Matrix Multimedia przyjęła inną koncepcję. Wyposażyła swoje *E-blocks* w praktycznie niezniszczalne złącza DB9, dzięki czemu płytki można łączyć ze sobą bokami tylko w taki sposób, w który do siebie pasują elektrycznie. Unika się w ten sposób ryzyka uszkodzenia, a sygnały trafiają odpowiednio zawsze tam, gdzie powinny.

Płytki *E-blocks* podzielono na dwie grupy, których nazwy trudno przetłumaczyć: *up-stream boards* oraz *down-stream boards*. Myślę, że w języku polskim dobrymi określeniami byłyby płytki *nadrzędne* (dla *up-stream*) oraz *podrzędne* (dla *down-stream*). Na płytce nadrzędnej montuje się mikrokontroler. Często jest ona również wyposażona w programator, jak na przykład *EB064 PIC24/dsPIC MCU*, na której można zamontować mikrokontroler PIC24 lub dsPIC i która jest dołączana do komputera PC za pomocą USB. Zwalnia to użytkownika z konieczności zakupu lub wykonania programatora PICkit. Płytki nadrzędne wyposażono w złącza żeńskie DSUB służące do przyłączania płytek z akcesoriami dodatkowymi. Sporadycznie trzeba wykonać połączenia za pomocą przewodów. Do tego celu służą złącza „pod

śrubkę", popularnie zwane terminatorami. Przykładową płytkę nadrzędną *PIC24/dsPIC Programmer Board* pokazano na **fotografii 1**.

Warto przy tej okazji wspomnieć, że mimo iż skupimy się w naszym cyklu na środowisku FlowCode 4 dla mikrokontrolerów PIC24 i dsPIC, to Matrix Multimedia oferuje również wersje kompilatorów i modułów *E-blocks* dla mikrokontrolerów PIC10, PIC12, PIC16, PIC18, 8-bitowych AVR i 32-bitowych ARM7. Dostępna jest również płytka *PLD Programmer Board* przeznaczona dla układów CPLD i FPGA.

Płytki podrzędne, *down-stream bards*, zawierają różne użyteczne układy peryferyjne. Są to peryferia typowo obsługiwane przez mikrokontroler, a więc dla przykładu: klawiatura, wyświetlacz, zasilacz silników prądu stałego, przekaźniki, modem GSM, moduły ZigBee, Bluetooth, GPS i RFID, interfejsy CAN, LIN, RS232, IrDA, MIDI, SPI, I²C, Ethernet i inne. Co ważne, dla płytek jest dostępna obszerna dokumentacja zawierająca schematy i przykłady programowania, nie tylko z użyciem FlowCode, ale również języka C i asemblera. Zazwyczaj płytki podrzędne nie wymagają zasilania lub są zasilone z płytki z mikrokontrolerem. W niektórych przypadkach może jednak zdarzyć się konieczność zastosowania zasilacza zewnętrznego. W ofercie Matrix'a jest dostępna również płyta montażowa z otworami i plastykowymi słupkami dystansującymi, dzięki której można „utrwalić" konfigurację, nad którą pracujemy, zasłonić ją przeźroczystymi osłonami. Nie jest ona jednak niezbędna przy korzystaniu z zestawu. Przykłady niektórych płytek podrzędnych pokazano na **fotografii 2**.

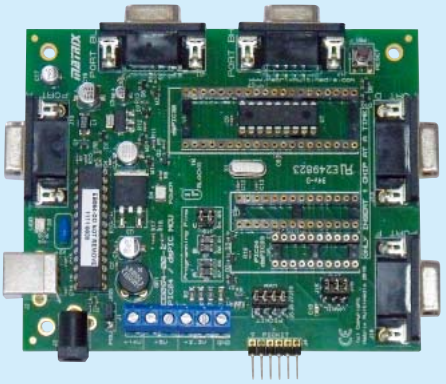
Mimo iż schemat połączeń elektrycznych w obrębie *E-blocks* jest ustalony przez producenta, to sygnały są doprowadzone do szpilek goldpin zwartych zworkami. Można je rozewrzeć i doprowadzić sygnały inaczej, zgodnie z własną potrzebą lub schematem budowanego prototypu. Z drugiej strony – programowej, elastyczne konfigurowanie makr również pozwala na dowolne przyporządkowanie wyprowadzeń mikrokontrolera sygnałom wejścia/wyjścia. Będzie o tym mowa dalej. W praktyce jednak, jeśli nie wykorzystujemy *E-blocks* do budowy prototypu a jedynie np. do nauki programowania, to nie ma potrzeby przełączania sygnałów. Warto jednak zauważyć, że jest taka możliwość dla większości *E-blocks*.

Takie ustandaryzowane doprowadzenie sygnałów do trwałych DSUB'ów ma jedną, bardzo ważną zaletę. Dzięki niemu *E-blocks* są kapitalną ofertą edukacyjną, ponieważ połączenia pomiędzy modułami zawsze wykonywane są w ten sam sposób, sygnały zawsze trafiają tam gdzie powinny, zminimalizowano możliwość pomyłki i wykonania błędnych połączeń, a złącza DSUB są trwałe i trudne do uszkodzenia nawet przez niewprawnego elektronika.

Moim zdaniem, *E-blocks* przydadzą się konstruktorowi – elektronikowi do szybkiego sprawdzenia rozwiązania układowego bardziej, niż do budowy prototypu urządzenia, ale przede wszystkim są po prostu kapitalną ofertą edukacyjną, na co wskazuje również obszerna, dosłownie „łopatologiczna" dokumentacja techniczna i przykłady programów użytkowych.

FlowCode

Zestawy *E-blocks* mogą być używane niezależnie od wybranego języka programowania, jednak ich doskonałym uzupełnieniem jest kompilator FlowCode. Umożliwia on wykonanie programu dla mikrokontrolera, bez znajomości języka programowania. Co prawda, w bloczkach FlowCode można umieszczać instrukcje języka C, jednak dla tych najprostszych zastosowań nie trzeba będzie tego robić.

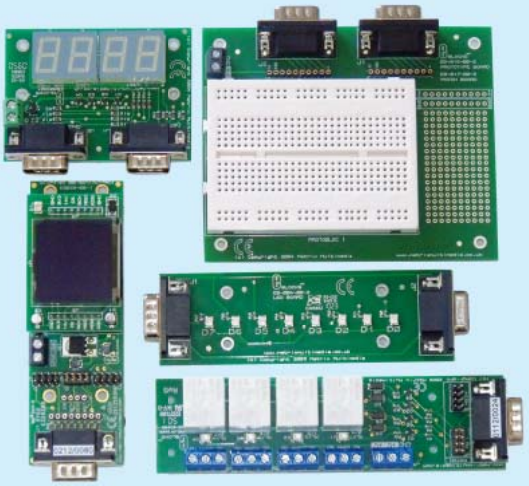


Fotografia 1. Płytka nadrzędna (up-stream bords) PIC24/dsPIC Programmer Board

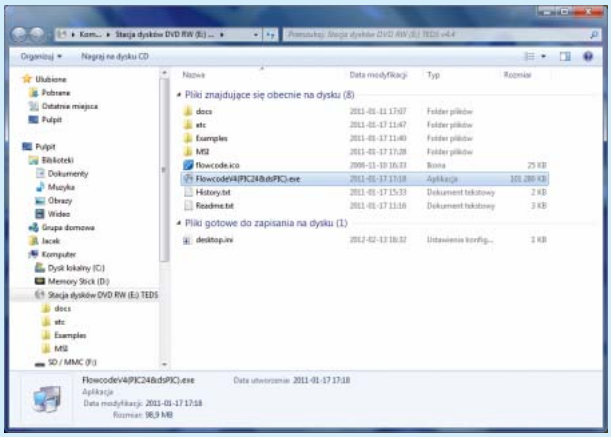
Jak wspomniano, istnieją wersje kompilatora przeznaczone dla mikrokontrolerów PIC10, 12,16 i 18; PIC24 i dsPIC; AVR (ATtiny, ATmega, ATxmega, AT90) i ARM (AT91SAM7), więc każdy znajdzie tu coś odpowiedniego dla stosowanego przez siebie sprzętu. FlowCode współpracuje z wieloma popularnymi programatorami (w wypadku Microchipa są to PICkit-2 oraz 3), ale najłatwiej używa się go z programatorem firmowym wbudowanym na niektórych płytkach nadrzędnych *E--blocks*. Moim zdaniem, FlowCode jest jeszcze łatwiejszy do użycia od popularnego języka Bascom AVR i ma szansę stać się dla niego silną konkurencją.

Kompilator FlowCode jest dostarczany na płycie CD w pudełku, wewnątrz którego znajduje się również indywidualny numer licencji. Po zainstalowaniu kompilatora jest wymagane zarejestrowanie się na stronie internetowej oraz aktywowanie programu. Bez tego, FlowCode będzie pracował bez ograniczeń, ale jedynie przez 30 dni. Prześledźmy jak wygląda proces instalacji i rejestracji.

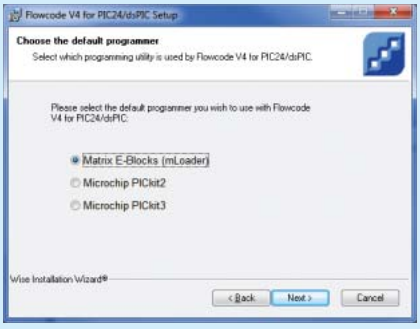
Kupiłem kompilator FlowCode w wersji *Professional* przeznaczonej dla mikrokontrolerów PIC24 i dsPIC. Instalator uruchamia się ręcznie, klikając na nazwie pliku *FlowcodeV4(PIC24&dsPIC).exe* dostępnego w głównym drzewie katalogów. (**rysunek 3**) na płycie CD. Instalacja w zasadzie przebiega standardowo – w kolejnych wyświetlanych przez program okienkach wpisuje się nazwę użytkownika i organizacji, akceptuje warunki



Fotografia 2. Przykłady „klocków" E-blocks



Rysunek 3. Zawartość płyty instalacyjnej FlowCode v4



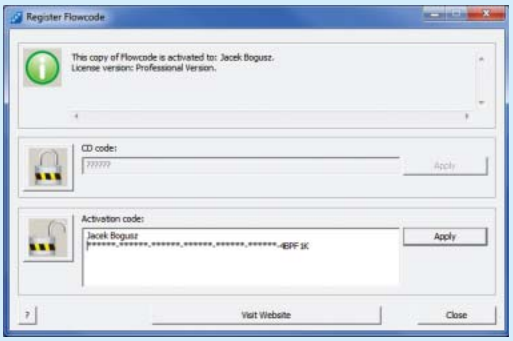
Rysunek 4. Wybór programatora podczas instalowania programu

Rysunek 5. Okienko z zapytaniem o numer licencji – umieszczono go w pudełku

licencji i wskazuje katalog docelowy, w którym zostaną umieszczone pliki programu. W związku z tym, że moje FlowCode jest przeznaczone dla mikrokontrolerów Microchip, to musiałem zdecydować się, z którym programatorem będę używał kompilatora (**rysunek 4**). To pozornie nic nieznaczące pytanie sprawiło mi później wiele kłopotów.

Z założenia chciałem użyć FlowCode do programowania mikrokontrolerów bez powiązania z żadnym zestawem. W związku z tym, że mam programator PICkit-3 wybrałem właśnie tę pozycję wychodząc z założenia, że przecież później można to łatwo zmienić. Tak jest w wielu znanych mi IDE, więc dlaczego tu miałoby być inaczej? Nie spodziewałem się żadnych problemów. Niestety, na próżno szukać w menu FlowCode v4 (być może poprawiono to wersji 5) opcji, która zrobi to automatycznie. Zmiana programatora odbywa

Rysunek 6. Formularz internetowy wypełniany w celu aktywowania pakietu



Rysunek 7. Wprowadzenie kodu aktywującego

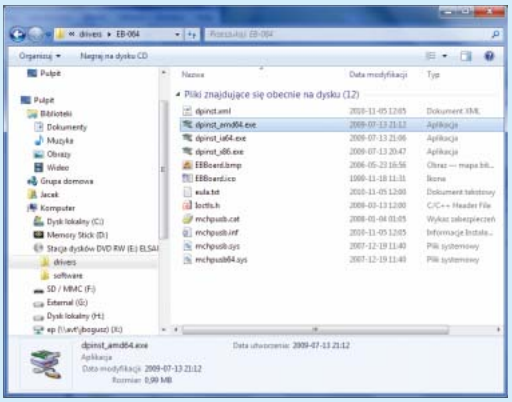
się poprzez wpisanie w linii komend, w oknie *Chip –> Compiler Options* odpowiedniego wywołania programu zewnętrznego. Co oczywiste, w linii komend trzeba podać parametry wywołania, ale jak? Odpowiedź na to pytanie wymagała przeszukania strony z najczęściej zadawanymi pytaniami (FAQ) odnośnie do kompilatora FlowCode v4. Mimo iż raczej rzadko zmienia się programatory, to w moim wypadku zaszła taka konieczność – niżej podam znalezione przeze mnie informacje, być może zaoszczędzi to komuś czas. Linie komend wpisujemy w polach *Programmer: Location* oraz *Parameters* we wspomnianym wcześniej oknie *Compiler Options:*

```
mLoader (E-blocks):
Location: C:\Program Files (x86)\Matrix
Multimedia\Flowcode PIC24&dsPIC V4\Tools\
mLoader\mLoaderv3.exe
Parametrers: -chip %p "%f.hex" -config
PICkit-2W
Location: C:\Program Files (x86)\Matrix
Multimedia\Flowcode PIC24&dsPIC V4\Tools\
PICkit2\pk2cmd.exe
Parameters: -BC:\Program Files (x86)\
Matrix Multimedia\Flowcode PIC24&dsPIC
V4\Tools\PICkit2\ -PPIC%p -F"%f.hex" -M -
A5 -R
PICkit-3
Location: C:\Program Files (x86)\Matrix
Multimedia\Flowcode PIC24&dsPIC V4\Tools\
PICkit2\pk3cmd.exe
Parameters: -P%p -F"%f.hex" -E -M -Y
```

Należy również zaznaczyć pole *Use external program to set configuration options*, a poniżej „ptaszka" wpisać w polu *Location: C:\Program Files (x86)\Matrix Multimedia\Flowcode PIC24&dsPIC V4\Tools\PPP\PPPv3.exe*, natomiast w polu *Parameters: –cs 2 –chip PIC%p –config*. Oczywiście, jeśli kompilator jest zainstalowany w innym katalogu, to w wywołaniu trzeba go zmienić – ja podałem catalog, w którym jest zainstalowany mój kompilator. Wróćmy jednak do instalatora.

Po wgraniu plików na dysk jest wyświetlane okienko z zapytaniem o numer licencji (**rysunek 5**). Jeśli to pierwsza instalacja programu, to wpisujemy w nim klucz dostarczony wraz z programem. Po potwierdzeniu nasz klucz zostaje zapamiętany. Następnie mamy możliwość uruchomienia przeglądarki internetowej i dokonania rejestracji. Po uzupełnieniu pól (**rysunek 6**) otrzymujemy e-mail z kodem aktywującym program, który należy wprowadzić w okienku dostępnym w menu *Help ->Register* (**rysunek 7**).

Po tych krokach FlowCode jest gotowy do pracy. Po uruchomieniu pozytywnie zaskoczyło mnie to, że menu



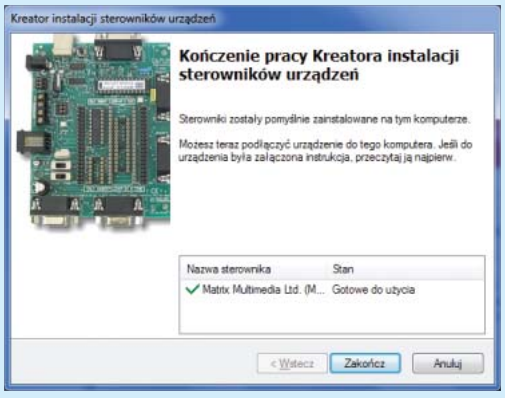
Rysunek 8. Plik instalatora sterownika programatora E-blocks

jest w języku polskim. Niestety, niektóre opcje nie zostały przetłumaczone i dlatego zmieniłem język kompilatora na angielski, który moim zdaniem lepiej pasuje do nazw parametrów mikrokontrolera, a ponadto unika się w ten sposób niejednoznaczności i błędów w tłumaczeniu.

E-blocks: PIC24 i dsPIC Programmer Board

Najwygodniej używa się FlowCode z firmowymi „klockami" *E-blocks*. Dla potrzeb demonstracji wybrałem płytkę oznaczoną jako EB064-00-2. Wyposażono ją w programator z interfejsem USB, zasilacz dostarczający napięć 3,3 V oraz 5 V, złącze programatora PICkit, podstawki dla układów PIC24 i dsPIC w obudowach DIP z 18, 20, 28 i 40 wyprowadzeniami. Płytka ma złącza DSUB9 żeńskie i jest nadrzędną (*up-stream board*). Jest dostarczana z płytą CD, na której można odnaleźć driver programatora oraz program *mLoader*. Moja płytka była dostarczona z mikrokontrolerem dsPIC30F2011.

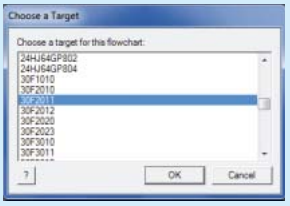
Sterownik programatora instaluje się „ręcznie" – jest on umieszczony w katalogu *drivers/EB-064*. Plik instalatora sterownika nosi nazwę *dpinst_amd64.exe* (**rysunek 8**). Pracuję z użyciem 64-bitowego systemu Windows 7 Home, który „broni się" przed oprogramowaniem nieautoryzowanym przez Microsoft. Dlatego na ekranie pojawiło się okienko z informacją o niezweryfikowanym producencie sterownika. Po kliknięciu na przycisk *Zainstaluj oprogramowanie sterownika mimo to* na dysku zostały umieszczone odpowiednie pliki, a na ekranie komputera pojawił się komunikat o pomyślnym przebiegu instalacji (**rysunek 9**).



Rysunek 9. Informacja o pomyślnym zainstalowaniu sterownika programatora

W pobliżu złącza zasilania umieszczono zworkę *J1*, która umożliwia wybór źródła zasilającego – przełączyłem ją w pozycję *„USB"*. Zgodnie z instrukcją, mój mikrokontroler wymagał umieszczenia w podstawce U7 oraz przełączenia zworki *Programming Pins* w pozycję „C" oraz zworek J16/J18 w pozycję *„I/O"*. Dodatkowo, w związku z tym, że wykorzystuję programator wbudowany na płytce, zworki *J5, J6, J7, J19* muszą być w pozycji *„USB"*.

Rysunek 10. Wybór tworzenia nowego programu



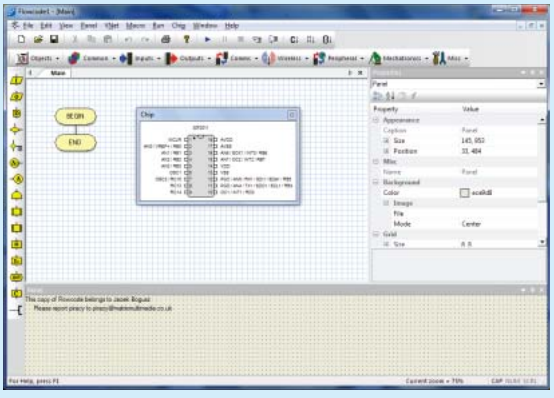
Rysunek 11. Okno wyboru mikrokontrolera

Teraz wolno dołączyć do komputera PC płytkę za pomocą kabla USB. Dzięki przełączeniu J1 jest ona zasilana z portu USB komputera PC. Po połączeniu z płytką, system Windows będzie poszukiwał sterowników. Aby zaoszczędzić nieco czasu, warto pominąć wyszukiwanie w witrynie Windows Update, w której instalator i tak niczego nie znajdzie.

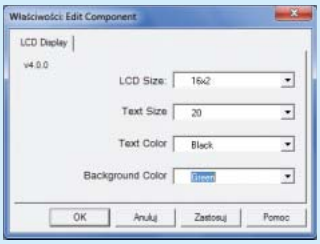
Pierwszy program

Chyba jak każdy programista, mając do dyspozycji kompilator, mikrokontroler oraz płytkę uruchomieniową nie mogłem doczekać się napisania pierwszego programu. W celu przetestowania zestawu postanowiłem utworzyć program do obsługi wyświetlacza LCD. Odpowiedni „klocek" dołączyłem do złącza *PORT BL*. Gdy wyjmowałem go z opakowania w ręce wpadła mi czerwona kartka informująca o konieczności zasilenia wyświetlacza ze źródła napięcia +5 V. Odpowiednie jest dostępne na złączu płytki nadrzędnej.

Po uruchomieniu FlowCode zaznaczyłem *Create a new FlowCode flowchart…* (**rysunek 10**). Z listy wybrałem mikrokontroler zainstalowany na płytce (**rysunek 11**). Po tym kroku pojawił się ekran roboczy, którego nieco pomniejszony pokazano na **rysunku 12**. Łatwo zauważyć, że kreator rozpoczął już za nas tworzenie programu i na ekranie mamy diagram z polami *BEGIN* oraz *END*, z których pierwsze zawiera funkcje inicjalizujące mikrokontroler, natomiast drugie – pętlę nieskończoną. Jak można się spodziewać, pomiędzy tymi polami będziemy umieszczali instrukcje.

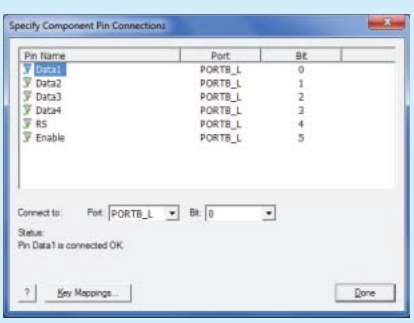


Rysunek 12. Ekran roboczy FlowCode v4

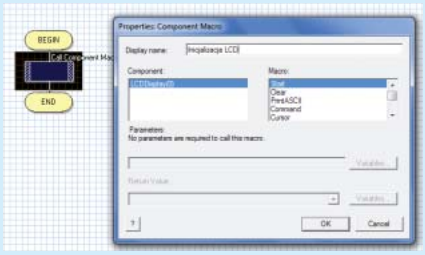


Rysunek 13. Konfigurowanie rodzaju wyświetlacza tekstowego LCD

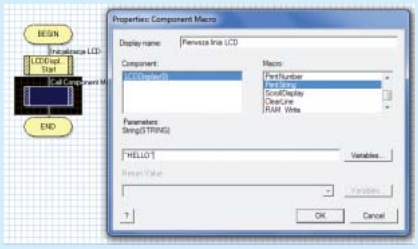
Do dyspozycji programisty oddano makra, których symbole są umieszczone na poziomym pasku znajdującym się nad obszarem roboczym. Są one związane z obsługą urządzeń wejścia – wyjścia, w tym wyświetlacza LCD. Po wybraniu odpowiedniego makra z paska na górze ekranu, symbol urządzenia zostaje umieszczony w oknie *Panel*. Do tego przykładu programu wybierzmy z grupy makr *Outputs* urządzenie *LCDDisplay*. Następnie wskazujemy symbol wyświetlacza w obszarze *Panel* i klikamy prawym klawiszem myszy. Z menu podręcznego wybieramy *Edit Component*. Okno pokazane na **rysunku 13** umożliwia wybranie liczby znaków na ekranie wyświetlacza LCD, długości wiersza tekstu oraz kolorów wyświetlacza w symulatorze. Po zatwierdzeniu ponownie wchodzimy do menu kontekstowego i tym razem wybieramy *Connections*. Dołączyłem wyświetlacz do portu *BL* – nastawy odpowiednie dla mojej konfiguracji można zobaczyć na **rysunku 14**. Każdą linię połączeń ustawia się indywidualnie wskazując ją, a następnie u dołu okienka wybierając odpowiedni port i numer bitu. Po kliknięciu OK wyświetlacz jest dołączony i można się spodziewać, że gotowy do użycia.



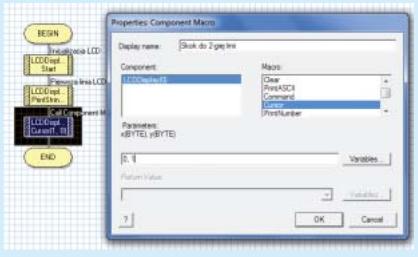
Rysunek 14. Konfigurowanie połączeń wyświetlacza LCD



Rysunek 15. Makro inicjalizujące wyświetlacz

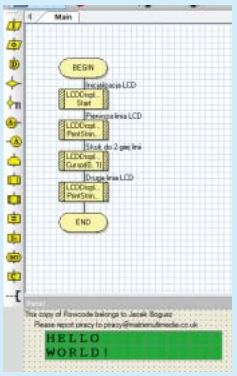


Rysunek 16. Makro wyświetlające – wpisanie komunikatu do wyświetlenia



Rysunek 17. Makro pozycjonujące kursor na ekranie LCD

Po lewej stronie ekranu jest pasek z żółtymi ikonami czynności. Do obsługi wyświetlacz służy makro, a więc można domyślić się, że odwołanie do niego następuje za pomocą pola *Component Macro* (żółty prostokąt z zakreskowanymi brzegami). Przeciągamy go na diagram pomiędzy pola *BEGIN* i *END*. Następnie dwukrotnie klikamy prawym klawiszem na symbolu na diagramie i w polu *Component* wybieramy *LCDDisplay(0)*. Teraz wybieramy czynność wymaganą dla wskazanego komponentu. Łatwo domyślić się, że wyświetlacz wymaga zainicjalizowania – z listy *Macro* wybieramy *Start*. W linii *Display name* można wpisać nazwę, która ułatwi późniejszą analizę programu, co pokazano na **rysunku 15**.



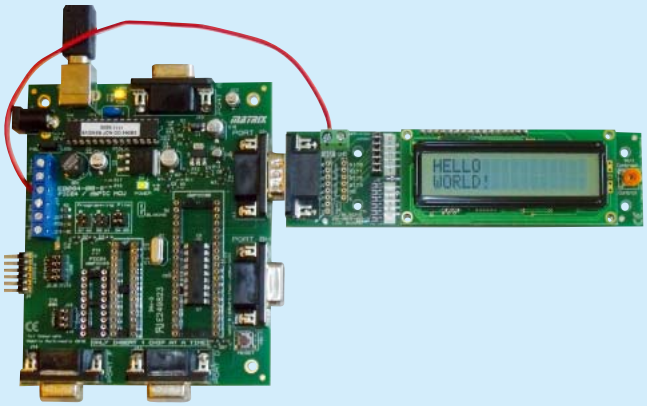
Rysunek 18. Efekt działania symulatora

Na początek postanowiłem w dwóch liniach wyświetlacza wyświetlić jakiś zwykły, stały komunikat. Do tego celu nie potrzeba zmiennych – wystarczą stałe. Po zainicjalizowaniu wyświetlacza można już wyświetlić napis. Opisaną wyżej metodą – *przeciągnij i upuść* – umieszczamy na diagramie kolejne pole *Component Macro*, ponownie wybieramy z listy *LCDDisplay(0)*, ale tym razem z listy *Macro* wybieramy *PrintString*, a w linii *Variables* w cudzysłowie wpisujemy komunikat. Wpisałem napis „HELLO" (**rysunek 16**). Wyświetlenie drugiej linii komunikatu wymaga przemieszczenia kursora. Ponownie robimy to za pomocą *Component Macro*, ale tym razem wybieramy z listy *Cursor* i wpisujemy odpowiednie współrzędne – u mnie to początek drugiej linijki, a więc x=*0*, y=*1* (**rysunek 17**). Po ustawieniu kursora, pozostaje wyświetlenie drugiego komunikatu – opisanym wcześniej sposobem ustaliłem go na „WORLD!". Wybierając z menu *Run* opcję *Go/Continue* sprawdziłem działanie programu za pomocą symulatora. Efekt jego pracy można zobaczyć na panelu, na wirtualnym wyświetlaczu pokazanym na **rysunku 18**. Następnie, wybierając z menu *Chip* opcję *Compile to Chip* przesłałem program do mikrokontrolera, co spowodowało efekt jak na **fotografii 19**.

Podsumowanie

Po tak łatwo osiągniętych efektach, bez żadnego żmudnego przeszukiwania Internetu i mozolnej nauki, ręce aż same palą się do tworzenia kolejnych programów, bo przecież równie łatwa jest obsługa klawiatury, serwomechanizmu, wyświetlacz LED i innych, co pokażę w kolejnych odcinkach kursu.

Jacek Bogusz, EP



Fotografia 19. Komunikat wyświetlany po zaprogramowaniu zestawu E-blocks

Redakcja Elektroniki Praktycznej dziękuje firmie TME z Łodzi, dystrybutorowi firmy Matrix Multimedia, za udostępnienie zestawu E-blocks oraz mikrokontrolerów używanych w kursie programowania FlowCode.