

# Vamgraf – cudze chwalicie, swego nie znacie...

*Mikrokontrolery to elementy o dużych możliwościach i niskiej cenie. Ich elastyczność powoduje, że mogą pracować w bardzo wielu zastosowaniach. Nie ma problemu, jeśli użytkownik zna języki programowania i jest w stanie dobrze taki układ zaaplikować. Ale jest szeroka grupa potencjalnych odbiorców na przykład: modelarzy, krótkofalowców czy profesjonalistów innych specjalności niż elektronika lub informatyka, którzy chętnie wykorzystaliby mikrokontrolery do swoich potrzeb. Jednak nauka programowania w assemblerze czy C wymaga pewnego wysiłku, a początkowy brak efektów często zniechęca do dalszych prób. Głównie z myślą o takich odbiorcach powstają środowiska projektowe pozwalające tworzyć programy w sposób bardziej intuicyjny.*

Idea takich narzędzi jest prosta: zamiast pracować wpisując w plikach tekstowych szereg przynajmniej na początku niezbyt zrozumiałych poleceń można program sobie narysować. Rysowanie polega na pobieraniu z przygotowanego magazynu elementów – klocków i łączenie ich tak by w końcowym efekcie tworzyły działające algorytmy. Praktyka pokazuje, że takiego projektowania można się nauczyć szybko i jeżeli nie będziemy potrzebować elastycznego zaawansowanego programowania, to w wielu prostych zastosowaniach zupełnie wystarczy.

Pierwszym takim narzędziem, o którym usłyszałem był opisywany w Elektronice Praktycznej ST6 Realizer przeznaczony do graficznego tworzenia programów dla mikrokontrolerów z rodziny ST6 firmy STM. Później poznałem angielski pakiet Flowcode. Firma produkująca ten pakiet zdecydowała się na „mocne uderzenie”. Po pierwsze, powstały środowiska dla najbardziej popularnych rodzin mikrokontrolerów: PIC Microchipa i AVR Atmela. Poza tym, jest dostępna wersja dla mikrokontrolerów z rdzeniem ARM z rodziny AT91SAM Atmela. Do środowiska projektowego jest dołączona spora oferta rozwiązań sprzętowych, od modułów bazowych z wbudowanym programatorem sterowanym przez USB z poziomu Flowcode, poprzez moduły rozszerzeń (wyświetlacze, przyciski, moduły Ethernet) po gotowe zamknięte urządzenia na przykład sterownik MIAC. Pakiet Flowcode był szeroko opisywany na łamach Elektroniki Praktycznej. Były też publikowane szczegółowo opisywane projekty i większość zainteresowanych może sobie wyrobić jakiś pogląd na temat przydatności tego środowiska. Sam używam Flowcode i mam o nim jak najlepsze zdanie. Jednak

ten artykuł nie będzie o Flowcode, tylko o innym graficznym środowisku – o Vamgrafie.

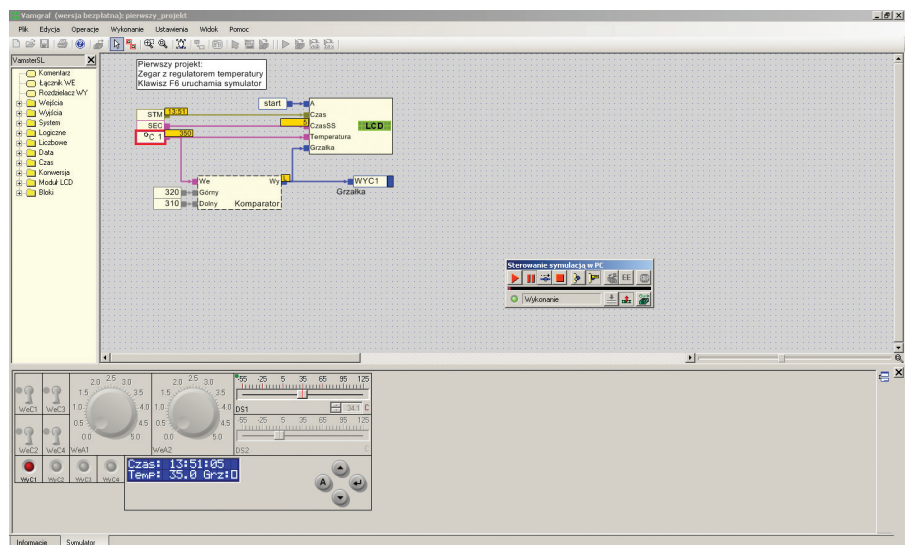
Vamgraf – podobnie jak Flowcode – jest narzędziem, które pozwala na tworzenie programów w sposób graficzny. Oprócz wielu ciekawych właściwości, o których napiszę, ma jedną, jakby nie mówić, zaskakującą cechę – jest polskim produktem. Przyznam, że opisywałem już czeski system modułów radiowych IQRF, używam słowackich programatorów firmy Elnec, znam serbski system środowiska projektowego dla mikrokontrolerów PIC firmy Mikroelektronika, ale z polskim środowiskiem projektowym mam do czynienia po raz pierwszy. Chociażby z tego powodu z wielkim zainteresowaniem zacząłem przyglądać się produktowi krakowskiej firmy Vamod.

Pakiet Vamgraf w wersji V2.49 można pobrać ze strony producenta spod adresu

[http://www.vamond.pl/download.php?plik=Vamgraf\\_2v49.zip](http://www.vamond.pl/download.php?plik=Vamgraf_2v49.zip). Vamgraf występuje w 2 wersjach: bezpłatnej i płatnej. Wersja bezpłatna, z której korzystałem, pozwala na zapoznanie się możliwościami i przetestowanie działania programu, ale bez zaprogramowania mikrokontrolera. Właściwie nie jest to precyzyjna definicja tej wersji – jest możliwość praktycznego przetestowania wyniku programowania w bezpłatnym Vamgrafie, ale pod warunkiem, że kupi się firmowy moduł VamsterB z mikrokontrolerem ATmega 188 lub VamsterD z mikrokontrolerem ATmega32. Jeżeli nie mamy tych modułów, to można Vamgrafa testować do woli, ale sprawdzenie działania rysowanych programów jest możliwe tylko w programowym symulatorze. Wersja bezpłatna staje się wersją płatną po zakupieniu w firmie kodu aktywacyjnego. W tej wersji Vamgraf generuje kod wynikowy .hex, którym można zaprogramować mikrokontroler ATmega32 za pomocą dowolnego programatora i bez konieczności kupowania firmowych modułów.

Po rozpakowaniu pliku 2v49.zip można uruchomić Vamgrafa przez kliknięcie na plik Vamgraf.exe umieszczony w katalogu Application files/Vamgraf\_2\_49\_0\_0. Otwiera się wtedy okno pokazane na **rysunku 1**. Nie różni się ono zbyt od okien podobnych programów tego typu i składa się z kilku głównych elementów:

- Menu głównego z sekcjami: *Plik, Edycja, Operacja, Wykonanie, Ustawienia, Widok i Pomoc*. Sekcje *Plik, Edycja, Widok i Po-*



Rysunek 1. Okno Vamgrafa

moc składają się ze standardowych elementów, poza kilkoma wyjątkami. Sekcja *Operacje* zawiera opcje, za pomocą których wykonuje się operacje edycyjne dla całego projektu. Sekcja *Wykonanie* jest przeznaczona do kompilowania i testowania projektu oraz do współpracy z firmowym modułem Vamster. Sekcja *Ustawienia* jest przeznaczona do modyfikowania ustawień *Vamgrafa* i *Vamstera*.

- Paska narzędzi z ikonami umożliwiającymi otwarcie nowego projektu, zapis projektu na dysku, otwarcie okna pomocy, ustawienie edycji elementów projektu w tryb wybieranie, zaznaczenie wszystkich elementów projektu itp.
- Panelu elementów. Ten panel zawiera pogrupowane elementy projektowe: komentarze, łącznik wejść, rozdzielacz wyjść, wejścia, wyjścia, system, logiczne, liczbowe, data, czas, konwersja, bloki. W zależności od rodzaju wybranego Vamstera, oraz modułów sprzętowych z nim współpracujących może się zmieniać ilość i rodzaj grup elementów.
- Panelu edycji. To w tym panelu jest wykonywane rysowanie schematu działania programu, czyli stawianie elementów z panelu elementów i wykonywanie połączeń pomiędzy nimi.
- Panelu informacji i symulacji. Dolny panel spełnia funkcję panelu informacji wypisywanych przez kompilator w tym informacji o błędach i jednocześnie jest panelem symulatora. Przełączanie pomiędzy tymi funkcjami odbywa się po kliknięciu na umieszczone na dole zakładki „Informacje” i Symulator”.

## Działanie systemu

System jest zbudowany w oparciu o mechanizm generowania zdarzeń. Przyjmuje się, że każdy system sterowania składa się z określonej liczby wejść i wyjść. Wejściami mogą być typowe, dwustanowe wejścia cyfrowe, wejścia analogowe oraz wejścia przypisane do zdefiniowanych w systemie układów, na przykład wejście do odczytywania temperatury z układu DS18B20. Każda zmiana na wejściu generuje zdarzenie, które wywołuje określone działanie projektowanej aplikacji. Ponieważ każdy system mikroprocesorowy ma ograniczone zasoby, to musi sobie jakoś radzić w sytuacjach, kiedy występuje wiele zdarzeń lub są one jednoczesne albo prawie jednoczesne. Dlatego zarejestrowane zdarzenie jest zapisywane do kolejki FIFO (pierwszy zapisany, pierwszy obsługiwany). Potem zdarzenia są sukcesywnie obsługiwane przez program – egzekutor zdarzeń. W wyniku działania egzekutora zmienia się stan wyjść i obsługiwane zdarzenia jest usuwane z kolejki. Oprócz opisanych elementów, czyli wejść/wyjść, kolejki i egzekutora, system wyposażono w stos wywołań z zabezpieczeniem przed pętlami, mechanizmy zarządzania

ekranami wyświetlania, obsługę pamięci EEPROM oraz obsługę zegara RTC.

Jak już wspominałem, zdarzenia są generowane przez zmianę na wejściach sprzętowych: cyfrowych, analogowych, pomiaru temperatury itp. To nie jest jedyne ich źródło. Oprócz tego mogą je zgłaszać wewnętrzne układy peryferyjne np. timery, układ RTC, oraz program użytkownika i system (start lub wygenerowany błąd).

## Elementy systemu

Wszystkie elementy systemu – a nie jest ich mało – są dokładnie opisane w pliku pomocy programu i nie będę tego opisu powielał. Skupię się tylko na kilku wybranych, tak aby mieć ogólne pojęcie z czego można budować program.

**Wejścia.** Wejścia cyfrowe są przeznaczone do odczytywania poziomów logicznych: wysokiego lub niskiego. W systemach mikroprocesorowych takie wejścia są szeroko stosowane na przykład do odczytywania stanu klawiszy, a ale też do implementacji linii interfejsów komunikacyjnych. Zaprogramowany sterownik Vamster zawsze odczytuje wejścia cyfrowe, jeśli tylko nie wykonuje obsługi zdarzeń, ale nie szybciej, niż co 1 ms. Jeżeli stan wejścia się zmieni, to system generuje zdarzenie i wpisuje je do kolejki. Liczba dostępnych wejść jest zależna od typu wybranego Vamstera. Na przykład VamsterD jest sterownikiem przeznaczonym do uruchomienia na płycie zaprojektowanej przez użytkownika. Do dyspozycji są wtedy 24 wejścia cyfrowe odpowiadające możliwościom mikrokontrolera ATmega32.

Vamstery mogą mieć zdefiniowany dodatkowy typ wejścia: klawisze. Są to 4 elementy stykowe oznaczone jako klawisz góra, dół, enter i esc. System domyślnie odczytuje je co 200 ms, ale kiedy mikrokontroler jest mocno obciążony obsługą zdarzeń, to ten czas może znacznie wydłużyć się. Jak się łatwo domyśleć, te elementy są głównie przeznaczone do obsługi menu kontekstowego. Każdy klawisz może występować w programie wielokrotnie, ale jego przyciśnięcie powoduje wygenerowanie jednego zdarzenia. Jeżeli kolejka jest pełna przed odczytaniem klawiszy, to odczytanie zmiany ich stanu jest gubione (nie jest wprowadzane do kolejki). Może zostać zarejestrowane kolejne przyciśnięcie, ale pod warunkiem, że i w tym wypadku kolejka nie jest zapełniona.

Wejścia analogowe konwertują napięcie z zakresu 0...5 V na wartość 10-bitową. Napięcie referencyjne przetwornika jest równe napięciu zasilającemu. Można w części sterowników Vamster dołączyć zewnętrzne źródło napięcia referencyjnego. Przy  $V_{ref}=5\text{ V}$  wynik konwersji napięcia  $V_{in}$  ma wartość  $V_{in} \times 1024/5$ .

**Wyjścia.** Elementy wyjściowe to cyfrowe wyjścia dwustanowe, wyjścia PWM, buzzer,

diody LED, wyświetlacze (LCD i 4-cyfrowy LED) oraz wyjścia wirtualne (testowe). Do każdego z wyjść są przypisane bufory. W trakcie obsługi zdarzenia te bufory są zapisywane i po zakończeniu obsługi przepisywane do wyjść sprzętowych. Na wyjściach dwustanowych i PWM poziom wysoki to +5 V, a niski to 0 V. Sygnał PWM może być generowany sprzętowo – wyjścia PWMS1 i PWMS2. Ma wtedy rozdzielczość 10-bitów i częstotliwość 900 Hz. Programowa realizacja PWM dostępna na wyjściach PWMP1 do PWMP8 ma rozdzielczość 8-bitów i częstotliwość 75 Hz. Możliwe jest zdefiniowanie aktywnego poziomu napięcia dla wyjść dwustanowych i PWM.

**Wyświetlacze.** Takie elementy wyjściowe jak diody LED i wyświetlacze są definiowane dla konkretnych typów sterowników. Dla konstruktora – szczególnie początkującego – bardzo atrakcyjna jest możliwość obsługi standardowego wyświetlacza LCD ze sterownikiem HD44780 i wyświetlacza LED złożonego z 4 cyfr.

Żeby prawidłowo używać elementów wyświetlaczy trzeba wiedzieć jak działają w systemie i jak się je konfiguruje. Z pojęciem wyświetlacza LED czy LCD związane są tzw. ekrany. Ekran to element, który określa wygląd i zawartość wyświetlanej informacji. Ekran jest aktywowany i dezaktywowany poziomem logicznym na swoim wejściu A. Ekrany są podzielone na ekrany wyświetlania, edycji, edycji daty i czasu itp.

Na wyświetlaczu może być widoczny tylko jeden ekran informacji, ale w programie może być ich więcej i wszystkie mogą być aktywowane w jednym czasie. Skoro tak, to powstają konflikty, z którymi Vamgraf radzi sobie tworząc listę aktywnych ekranów. Użytkownik nie ma dostępu do tej listy, a zarządzanie nią przez system wygląda następująco: po zerowaniu mikrokontrolera lista jest pusta. Kiedy program uaktywni ekran wymagając poziom aktywny na wejściu A, to ekran jest umieszczany na końcu listy i jest wyświetlany (ekran aktywny). Po dezaktywacji ekran jest usuwany z listy i zaczyna być wyświetlany ekran ze środka listy, czyli aktywowany wcześniej i niedezaktywowany. Inaczej mówiąc – jest wyświetlany ekran ostatnio aktywowany. Ekran przed nim jest przesuwany do środka listy, jeżeli pozostał aktywny. Po dezaktywacji ostatnio aktywowanego, ten ze środka na powrót staje się aktywny i wyświetlany. Jeżeli wszystkie ekrany informacji przestają być aktywne, to lista się opróżnia i na ekranie nic się nie wyświetla.

Ekranów informacji może być wiele i mogą być aktywne wszystkie. Ekranów edycji również może być wiele, ale aktywny może być tylko jeden. Założmy, że żaden ekran nie jest aktywny. Po uaktywnieniu ekranów wyświetlania przejmują one kontrolę nad wyświetlaczem, zgodnie z regułami

opisanymi przed chwilą. Jeżeli uaktywni się ekran edycji, to przejmuje kontrolę nad wyświetlaczem. Ekran wyświetlania informacji są nadal zarządzane (wpisywane na listę), ale nie są wyświetlane. Jeżeli uaktywni się kolejny ekran edycji, to jest ignorowany, aż aktywny ekran edycji nie zostanie dezaktywowany. Przy czym nie wystarczy wymuszenie poziomu nieaktywnego na wejściu A.

Z ekranem edycji jest powiązane działanie opisywanych już 4 klawiszy: góra, dół, enter i esc. Ten ekran przejmuje wszystkie zdarzenia od klawiszy. Żeby dezaktywować ekran edycji, trzeba nacisnąć i puścić klawisz enter lub esc. Może to się stać również po przekroczeniu timeoutu, kiedy żaden klawisz nie jest naciskany przez określony czas. Edycja parametrów wymaga zmian ich wartości. Jest to wykonywane przez przyciskanie klawiszy góra/dół. Enter akceptuje zmiany, nowa wartość jest zapisywana w pamięci EEPROM i menu przechodzi do ustawiania kolejnej wartości. Naciśnięcie klawisza esc powoduje przejście do poprzedniej edycji lub dezaktywowanie ekranu edycji.

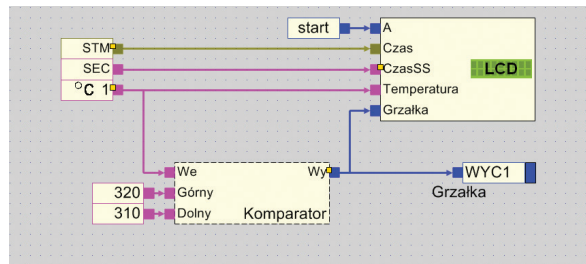
W trakcie edycji w tle działają wszystkie niezbędne procesy Vamstera, poza aktywowaniem innych ekranów edycji. Edytowane wartości będą odczytywane z pamięci EEPROM i zapamiętywane w niej po zmodyfikowaniu. Ekran edycji może mieć dodatkowe wyjścia: W, Enter, Esc. Ich działanie jest dokładnie opisane w instrukcji Vamgrafa.

**Projekt przykładowy – zegar z regulatorem temperatury**

Wiemy już jak działają ekrany. Popatrzmy teraz, jak można je wykorzystać w praktyce. Do tego celu wykorzystamy przykładowy (przygotowany przez producenta) projekt zegara z regulatorem temperatury przeznaczony dla sterownika Vamster SL. Narysowany projekt pokazano na **rysunku 2**. W jego skład wchodzi następujące elementy

- STM – czas w Vamsterze umieszczony w katalogu *Elementy\System\Data, czas*. Ten element generuje zdarzenie co 1 s.
- °C1 – element umieszczony w katalogu *Elementy>Wejście\Temperatura* i przeznaczony do pomiaru temperatury z czujnika DS1820. Ma jedno wyjście, na którym pojawiają się wartości z zakresu -550...1250. Odpowiada to mierzonej temperaturze z zakresu -55°C...125°C pomnożonej przez 10. Na przykład, temperaturze 13,5°C będzie odpowiadała wartość 135 na wyjściu elementu. Pomiar odbywa się co 1 s, a zdarzenie jest generowane tylko wtedy, kiedy odczytywana wartość uległa zmianie od ostatniego pomiaru. Jeżeli pomiar zostanie wykonany, ale kolejka zdarzeń jest pełna, to nie zostanie obsłużony (na przykład wyświetlony). Do sterownika można przyłączyć 2 czujniki DS1820. Po umieszczeniu na schemacie, system Vamstera sprawdza poprawność przyłączenia układu czujnika do mikrokontrolera i w wypadku błędów zgłasza stosowny komunikat.
- Element Start umieszczony jest w katalogu *Elementy\System\Start*, który generuje jedno zdarzenie po włączeniu zasilania lub zerowaniu mikrokontrolera. Zdarzenie polega na ustawieniu stanu wysokiego H.
- Element Blok. Jest to wcześniej zdefiniowany blok komparatora pozwalający na porównanie wartości na wyjściu pomiaru temperatury (wejście We) z dwoma stałymi wartościami 320 i 310 podawanymi na wejścia *Górny* i *Dolny*. Wyjście komparatora jest podawane na moduł wyświetlacza i jednocześnie na wyjście WYC1 do sterowania grzałką. Na **rysunku 4** pokazano rozwinięcie bloku komparatora

Ważnym elementem projektu jest bez wątpienia element wyświetlacza LCD. Może on mieć zdefiniowanych wiele wejść. W naszym przykładzie są to wejścia *A*, *Czas*, *CzasSS*, *temperatura* i *grzałka*. Na **rysunku 5** pokazano okno konfiguracji elementu. Po kliknięciu na przycisk *Dodaj*, w głównym menu dodaje się wejścia z ich nazwami i konfiguracją położenia na ekranie. Każdemu elementowi z listy wejść przypisuje się parametry położenia wyświetlania: numer linii i położenie początku tekstu. Przy wyświetlaniu czasu początek komunikatu jest ustalony na 7-my znak górnej linijki. Format jest ustalany w kolumnie *Typ Wyśw.* Wy-



Rysunek 2. Przykładowy projekt

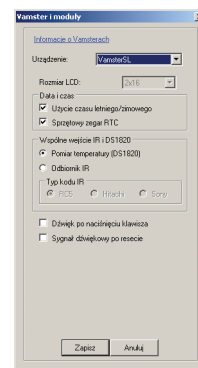
świetlanie czasu ma swój ustalony format i nie musimy się już troszczyć o dodatkowe konwersje, wyświetlanie dwukropka itp.

Ponieważ element *STM* odczytuje czas z rozdzielczością 1 minuty, to żeby zegar zliczał też sekundy jest potrzebne dodatkowe odczytywanie i wyświetlanie sekund. Wykonuje to element *SEC* połączony z wejściem *CzasS*. Wejście *CzasS* ma format *Liczbowe* i jest wyświetlane za minutami w górnej linijce.

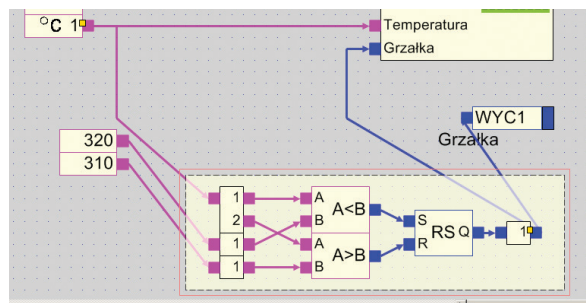
Temperatura jest mierzona przez element *°C*. Jego wyjście jest połączone z wejściem *Temperatura*. Dane z tego wejścia mają format liczbowy, czyli można określić liczbę znaków zarezerwowanych do wyświetlania i położenie kropki dziesiętnej.

Wejście *grzałka* ma format *logiczne*. Można tu zdefiniować znak przypisany do poziomów niskiego i wysokiego oraz położenie tego znaku na ekranie. Dodatkowe teksty niezwiązane z danymi z wejścia, czyli etykiety *Czas.*, *Temp* i *Grz* są edytowane litera po literce. Klikamy na edytowaną pozycję i z menu mającego obrazować generator znaków wyświetlacza wybieramy potrzebny. Tak zdefiniowany element LCD potrafi wyświetlać wszystkie dane otrzymane z zegara i termometru.

Program można testować symulatorem uruchamianym na komputerze PC. Ekran symulatora pokazano na **rysunku 6**. Jest na nim umieszczony widok ekranu wyświetlacza LCD z informacjami zdefiniowanymi

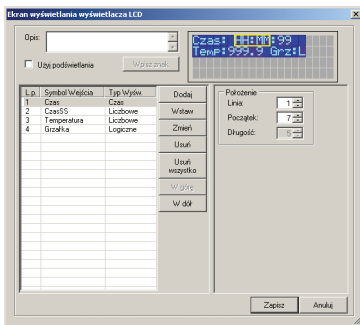


Rysunek 3. Okno Vamstera i moduły

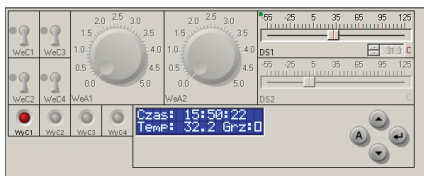


Rysunek 4. Rozwinięcie bloku komparatora.





Rysunek 5. Okno Edycji elementu LCD



Rysunek 6. Ekran symulatora dla testowanego programu

w oknie edycji elementu LCD. Symulowany czas jest pobierany z czasu komputera PC. Temperaturę w zakresie -55...125 °C zadajemy suwakiem oznaczonym DS1. Zmieniana temperatura jest wyświetlana na ekranie LCD. Można też sprawdzić działanie komparatora temperatury. Wyjście *WyC1* jest aktywne, gdy temperatura wzrasta i jest wyższa niż 32°C, a staje się nieaktywne, gdy temperatura maleje i jest niższa niż 31°C (histereza 1°C).

### Podsumowanie

Program Vamgraf razem z ofertą zestawu różnych sterowników z rodziny Vamster mógłby być ciekawą propozycją dla konstruktorów pragnących wykorzystywać mikrokontrolery w swoich projektach. Graficz-

na edycja programów, bardzo dobrze przygotowany plik pomocy i sporo programów przykładowych, to niewątpliwe zalety tego środowiska. Po teście muszę przyznać, że całość sprawia wrażenie narzędzia przemysłowego i dopracowanego, szczególnie pod względem graficznym. Ktoś włożył w nie dużo pracy i efekty tej pracy widać. Nie miałem co prawda możliwości sprawdzenia poprawności działania w rzeczywistym układzie, bo dysponowałem tylko bezpłatną wersją dostępną w Internecie. Jednak programowy debugger działał bardzo dobrze. Widać, że autorzy starali się zbudować system, w którym można szybko zaprogramować uniwersalny sterownik mierzący na przykład temperaturę i czas oraz wyświetlający to na ekranie. W programach przykładowych jest ciekawy projekt woltomierza z wyświetlaczem LED, sterownika akwariowego, sterownika silnika krokowego itp. To sugeruje, że grupą docelową są albo początkujący programiści, albo użytkownicy mikrokontrolerów, którzy potrzebują sterownika i nie chcą zbyt wniknąć w tajniki programowania. Dla nich silnym wsparciem będą pewnie gotowe sterowniki z rodziny Vamster.

Przy rysowaniu programów trzeba się przyzwyczaić koncepcji budowy algorytmów. Vamgraf przypomina trochę koncepcję programu Lab View. Trudno się oprzeć porównaniu z Flowcode. Kompilator Flowcode jest oparty na graficznym tworzeniu algorytmów bardzo bliskim algorytmom wykorzystywanym przy projektowaniu programów pisanych w językach programowania. Tam pętle, skoki warunkowe, sprawdzanie warunków itp. jest bardzo bliskie konstrukcjom programowym w C. Ponadto, można robić wstawki napisane w C, chociaż nie jest to

niezbędne. W przypadku Vamgrafa niezbędną jest umiejętność korzystania z układów logicznych: bramek, przerzutników, multiplexerów. Niestety sporym ograniczeniem jest możliwość rysowania programów prawie wyłącznie dla firmowych sterowników z rodziny Vamster. Można co prawda wykorzystać projekt z VamsterD i w wersji płatnej rysować program dla własnego sterownika, ale ten sterownik może być zbudowany tylko w oparciu o ATmega32. Flowcode oferuje znacznie więcej przy niższej cenie. Na przykład w wersji dla Microchipa można korzystać z wielu mikrokontrolerów rodziny PIC16 i PIC18, a generowany kod nie jest związany z firmowym sprzętem. Podejście twórców Vamgrafa sprowadza się do tego, że jest on dodatkiem do sterowników Vamster, a nie uniwersalnym narzędziem programowym.

Vamgraf jest również narzędziem bardzo mało znanym. Dowiedzieliśmy się o nim z kilku wpisów na forach internetowych. Bez większej reklamy i akcji promocyjnej nie można sprawić, by stał się bardziej popularny, a według mnie zdecydowanie na to zasługuje i to nie tylko dlatego, że jest polski. Być może gdyby twórcy zdecydowali się na rozszerzenie funkcjonalności polegające na przykład na zwiększeniu listy obsługiwanych mikrokontrolerów, nie wzięli tak mocno możliwości programowych z własnymi sterownikami Vamster, obniżyli i cenę i mocniej zaczęli promować swój produkt, to mógłby się stać bardziej popularny. W obecnej sytuacji obawiam się, że niestety, nie doczekamy się polskiego narzędzia na miarę Bascoma czy Flowcode.

Tomasz Jabłoński, EP

**ZAJRZYJ NA TE STRONY**

**RENEX**  
NARZĘDZIA DLA ELEKTRONIKÓW  
[www.renex.com.pl](http://www.renex.com.pl)

**ZAJRZYJ NA TE STRONY**

**www.piekarz.pl**  
Hurtownia części elektronicznych  
firma@piekarz.pl tel. 022-835-50-37 fax 022-213-92-82

sklep. **FERYSTER**.pl  
ELEMENTY INDUKCYJNE

**HUMA Co.**  
[www.humasklep.pl](http://www.humasklep.pl)  
KONTAKT

**WO BIT**  
[www.wobit.com.pl](http://www.wobit.com.pl)  
silniki.pl  
silniki.com  
enkodery.pl  
Czujemy i poruszamy

**GAMMA**  
[www.gamma.pl](http://www.gamma.pl)  
info@gamma.pl  
PODZESPOŁY ELEKTRONICZNE