

# Kurs Arduino (2)

## Oprogramowanie Arduino IDE



W EP 4/2011 rozpoczęliśmy kurs programowania Arduino. Omówiliśmy wtedy elementy języka. W tym artykule zajmiemy się Arduino IDE oraz utworzymy pierwszy program, który jeszcze nie będzie użyteczny, ale będzie takim arduinowym „Hello World”.

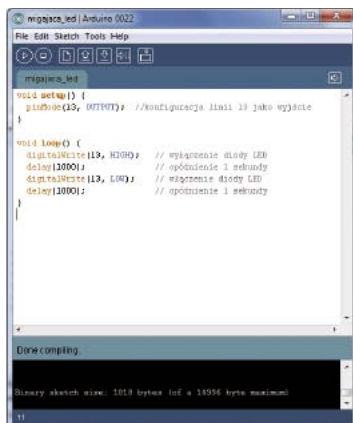
**Dodatkowe materiały na CD/FTP:**  
<ftp://ep.com.pl>, user: 12147, pass: 2e7u6a2a  
 • pierwsza część kursu

Do programowania w systemie Arduino jest przeznaczony program Arduino IDE, który jest dostępny na stronach <http://arduino.cc/en/Main/Software>. Można pracować pod kontrolą systemów operacyjnych Windows, Linux oraz MAC OS X.

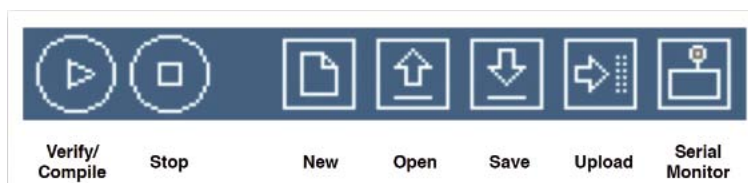
Po zainstalowaniu oprogramowania Arduino uruchamia się za pomocą skrótów lub przez dwukrotne kliknięcie na plik „Arduino.exe”. Wygląd głównego okna pokazano na **rysunku 1**. Okno programu można podzielić na kilka części. Dostępny jest górny pasek narzędzi, okno na przygotowywany program oraz obszar na dole, w którym będą wyświetlane komunikaty związane z działaniem programu Arduino IDE: o pracy kompilatora, o błędach i programowaniu mikrokontrolera. Pliki z przygotowanym programem dla mikrokontrolera są w Arduino zapisywane z rozszerzeniem *.pde*.

Pasek narzędzi składa się z 7 przycisków. Dostępny jest jeden przycisk (ze strzałką w prawo) przy wyborze kartotek, którego użycie umożliwia dostęp do poleceń zarządzania kartotekami z plikami programu. Dostępne jest również menu podzielone na grupy *File*, *Edit*, *Sketch*, *Tools* oraz *Help*. Dodatkowe przyciski (pokazane na **rysunku 2** i opisane w **tabeli 1**) umożliwiają szybki dostęp do najczęściej używanych poleceń (zapis pliku na dysku, programowanie itp.).

Po wybraniu ikony *Verify/Compile* kompilator sprawdza składnię programu, a na



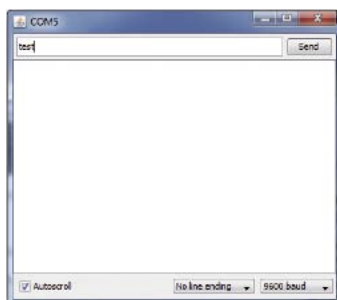
Rysunek 1.



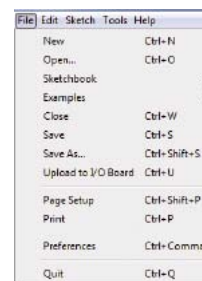
Rysunek 2.

stepnie jest on poddawany kompilacji. Po jej prawidłowym zakończeniu program jest gotowy do wysłania do mikrokontrolera. W przypadku nieprawidłowości w kodzie zostaną w dolnej części okienka systemu Arduino wyświetlone znalezione błędy.

Ikona przycisku *Stop* zatrzymuje działanie *Serial Monitor* (monitor komunikacji szeregowej). Jest to pomocne, gdy przesyłane informacje przez interfejs szeregowy RS232 pojawiają się szybciej, niż można je zaobserwować. Przycisk *New* umożliwia utworzenie nowego, pustego pliku dla programu. Należy podać nazwę nowego pliku i jego lokalizację na dysku. Przycisk *Open* umożliwia otwarcie pliku z programem z dostępnej listy plików w wybranym katalogu. Przycisk *Save* umożliwia zapisanie przygotowanego programu do pliku o podanej nazwie i w wybranym ka-



Rysunek 3.



Rysunek 4.

talogu. Przycisk *Upload* umożliwia przesłanie programu do mikrokontrolera a dokładnie do zestawu Arduino. Wcześniej należy przygotowany program poddać weryfikacji i kompilacji.

Przed wysłaniem programu do mikrokontrolera należy skonfigurować typ zestawu Arduino oraz numer portu w komputerze, do którego jest dołączony zestaw Arduino. Przycisk *Serial Monitor* uruchamia okno monitora komunikacji (**rysunek 3**) przez interfejs RS232. W jego oknie pojawiają się informacje wysyłane przez interfejs RS232 mikrokontrolera (zestaw Arduino). Za jego pomocą jest również możliwość wysyłania danych do mikrokontrolera. W oknie monitora są dostępne opcje automatycznego przewijania otrzymanych znaków, możliwość wyboru prędkości transmisji czy opcji związanych ze znakami końca linii. Monitor będzie po-

Tabela 1. Pasek przycisków

Przycisk	Funkcja
Verify/Compile	Sprawdza i poddaje kompilacji napisany kod programu
Stop	Zatrzymuje działanie monitora interfejsu RS232
New	Tworzy nową pustą zakładkę na program
Open	Otwiera plik z programem
Save	Zapisuje plik z programem
Upload	Umożliwia wysłanie programu do mikrokontrolera z wykorzystaniem szeregowego interfejsu RS232
Serial Monitor	Wyświetla okno monitora interfejsu RS232

# Oferta dla prenumeratorów Elektroniki Praktycznej

## Avtduino specjalnie z myślą o elektronikach-praktykach!

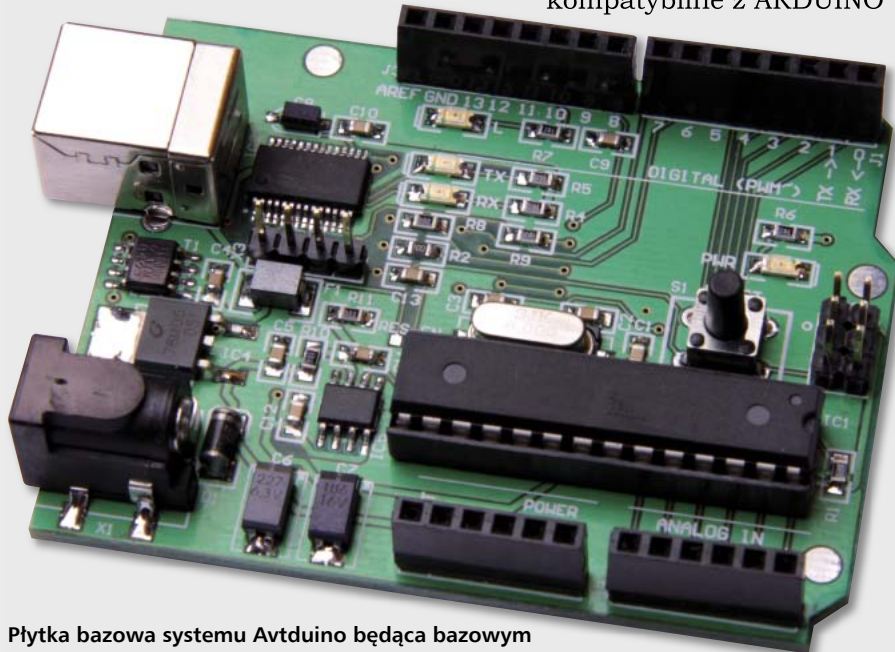
Od numeru EP 04/2011 rozpoczęliśmy kurs programowania mikrokontrolerów AVR z użyciem bezpłatnego środowiska programistycznego Arduino. Kurs będzie się opierał na przykładach przygotowanych dla płytek rozszerzających do bazy (kompatybilnej z systemem modułów Arduino) wyposażonej m.in. w mikrokontroler ATmega, opisanej w EP1/2011 (odpowiednik Arduino Duemilanove, AVT-5272).

Dla prenumeratorów Elektroniki Praktycznej przygotowaliśmy niespodziankę: wszystkim prenumeratorom papierowej wersji miesięcznika w grudniu 2011 zaoferujemy za darmo jedną, wybraną płytkę drukowaną modułu rozszerzenia dla zestawu **Avtduino** (zgodne z Arduino), dla których przykłady aplikacji przedstawimy w ramach kursu publikowanego na łamach czasopisma.

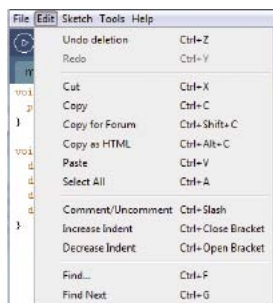
Pierwsze artykuły kursowe o Arduino opublikowaliśmy w EP 4/2011 na stronach: 96 i 98.

Opis pierwszego modułu rozszerzającego do płyty bazowej **Avtduino** opublikowaliśmy w Elektronice Praktycznej 4/2011 na stronie 47 (AVT-1615), kolejnego w bieżącym numerze na stronie 55 (AVT-1616).

**AVTduino**  
kompatybilne z ARDUINO



Płytkę bazową systemu Avtduino będącą bazowym rozwiązaniem dla uczestników kursu



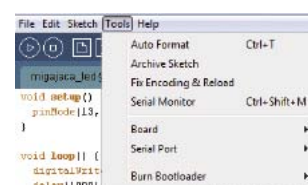
Rysunek 5.

mocny podczas sprawdzania pracy programu i wyszukiwania w nim błędów. Przycisk



Rysunek 6.

*Send* umożliwia wysłanie danych do mikrokontrolera w zestawie Arduino. Działanie monitora transmisji szeregowej można zatrzymać przyciskiem *Stop*. Aby ponownie uruchomić monitor wystarczy przycisnąć przycisk *Serial Monitor*. Korzystanie z monitora szeregowej transmisji będzie pokazane podczas praktycznych przykładów odczytu



Rysunek 7.

danych z czujników, gdy do zestawu nie będzie dołączony wyświetlacz.

U góry okna Arduino IDE znajduje się menu składające się z przycisków *File*, *Edit*, *Sketch*, *Tools* oraz *Help*. W menu *File*, które pokazano na **rysunku 4** umieszczono funk-

REKLAMA

**EAE ELEKTRONIK**

od pomysłu  
po wyrób

- Montaż SMT i THT zgodny z normą IPC-A-610D
- Produkcja wiązek kablowych
- Doradztwo techniczne, projektowanie
- Kompleksowa obsługa zamówień
- Testy EMC, badania środowiskowe

atrakcyjne ceny  
szybka realizacja

EAE Elektronik Spółka z o. o.  
ul. Przemyska 24d, 38-500 Sanok  
www.eae-elektronik.pl, tel.: +48 13 463 3773

Oficyjalny  
dystrybutor  
Arduino

**Nettigo**

Odkrywamy tajniki elektroniki

Oferta specjalna dla czytelników  
Elektroniki Praktycznej

<http://nettigo.pl/ep>

Starter Kit dla Arduino UNO - zestaw 20 części  
i 10 przykładów dla chcących rozpocząć.

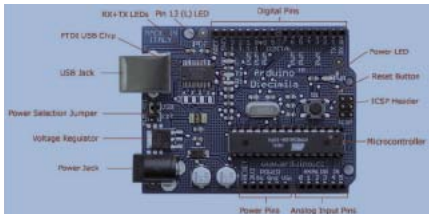


Rysunek 8.

**Listing 1. Program powodujący miganie diody LED**

```
void setup() {
  pinMode(13, OUTPUT); //konfiguracja linii 13 jako wyjście
}

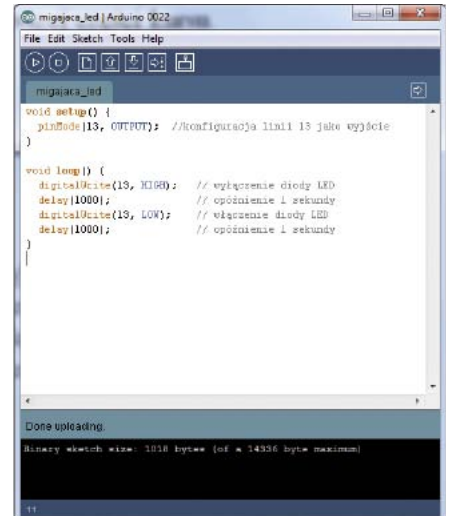
void loop() {
  digitalWrite(13, HIGH); // wyłączenie diody LED
  delay(1000); // opóźnienie 1 sekundy
  digitalWrite(13, LOW); // włączenie diody LED
  delay(1000); // opóźnienie 1 sekundy
}
```



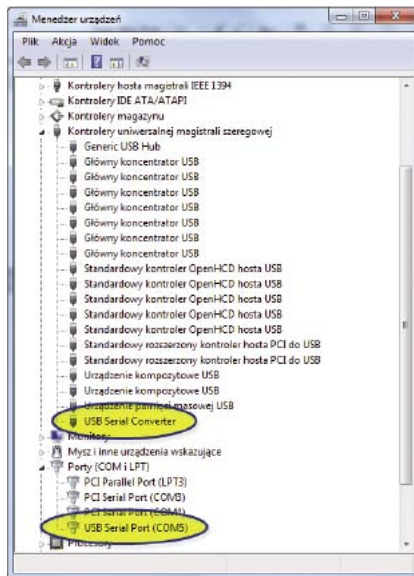
Rysunek 9.



Rysunek 12.



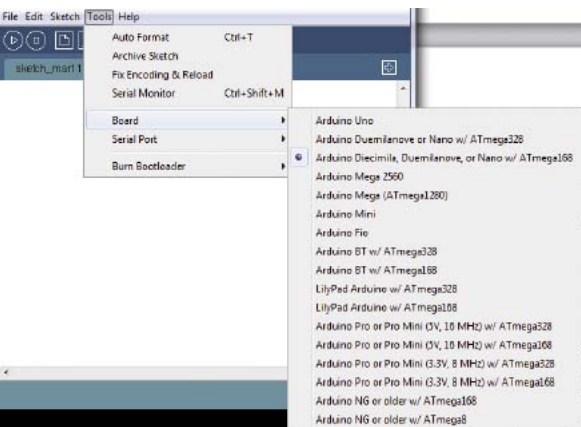
Rysunek 13.



Rysunek 10.

cje tworzenia nowego programu, jego zapisu oraz zamknięcia. W opcji *Example* znajdują się przykładowe programy dla Arduino. Można w nich znaleźć dużo przykładów podzielonych na grupy. Są to programy do obsługi silników, wyświetlaczy czy czujników. Dostępnych jest kilkadziesiąt przykładowych programów. Opcja *Upload to I/O Board* wysłała program do mikrokontrolera. Opcja *Page Setup* umożliwia ustawienie opcji

strony, natomiast w opcji *Preferences* jest możliwość zmiany opcji konfiguracyjnych Arduino IDE. Oczywiście dostępna jest również opcja drukowania. W menu *Edit* pokazanym na **rysunek 5** dostępne są opcje cofania zmian w programie, kopiowania, wklejania oraz wycinania. Są również opcje wstawiania komentarzy do programu, zaznaczania oraz opcje wyszukiwania *Find* i *Find Next*. W menu *Sketch* (**rysunek 6**) znajdują się ważne opcje związane z weryfikacją i kompilacją programu (*Verify/Compile*). Ponadto dostępna jest opcja *Stop* zatrzymująca monitor komunikacji szeregowej. Opcja *Show Sketch Folder* pokazuje folder z programem, natomiast opcja *Import Library* umożliwia import biblioteki z której będzie korzystał przygotowywany program (do programu wstawiany jest odnośnik do wybranej biblioteki). Program wyświetla biblioteki znajdujące się folderze *libraries* oprogramowania Arduino IDE. Domyślnie dostępnych jest kilkanaście bibliotek do których można również dołączyć biblioteki dostępne w Internecie. Opcja *Add File* umożliwiła dodanie kolejnego pliku do przygotowywanego programu. W menu *Tools* pokazanym na **rysunku 7** znajdują się narzędzia dzięki którym będzie możliwa komunikacja z zestawami Arduino. Opcja *Auto Format* umożliwia z formatowanie napisanego programu (wprowadza wcięcia sprawiające że program będzie bardziej czytelny).



Rysunek 11.

Opcja *Archive Sketch* umożliwia archiwizację przygotowanego oprogramowania do formatu ZIP. Dostępna jest również opcja *Serial Monitor* włączająca monitor komunikacji szeregowej. W opcji *Board* jest możliwość wyboru zestawu Arduino z którym będzie się komunikowało oprogramowanie Arduino IDE. Natomiast w opcji *Serial Port* wybiera się numer portu szeregowego z którym będzie odbywała się komunikacja z zestawem Arduino. W opcji

*Burn Bootloader* znajdują się obsługiwane przez Arduino programatory zewnętrzne za pomocą których jest możliwość przesłania do mikrokontrolera tak zwanego programu *Bootloadera* za pomocą którego będzie możliwe programowanie mikrokontrolera bez potrzeby wykorzystywania dodatkowego zewnętrznego programatora. W menu *Help* (**rysunek 8**) znajdują się opcje związane z odnośnikami do stron internetowych o Arduino, jego obsłudze, sposobu programowania czy dostępnych instrukcji.

**Podstawowe elementy zestawu Arduino IDE**

Odpowiednik zestawu Arduino UNO (Avtduino) z mikrokontrolerem ATmega168, który będzie używany podczas kursu, był opisywany w styczniowej Elektronice Praktycznej. Na **rysunku 9** pokazano elementy płytki zestawu Arduino UNO. Zestaw ma gniazdo USB, za pomocą którego będzie się odbywać komunikacja komputera PC z mikrokontrolerem zestawu. Jest ona przeprowadzana z użyciem konwertera USB-RS232. Zestaw może być zasilany z interfejsu USB lub z wykorzystaniem zewnętrznego zasilania.

W zestawie dostępne są linie *DIGITAL* (cyfrowe) mikrokontrolera oznaczone 0...13 przy czym linie 0 i 1 są liniami interfejsu RS232. Dostępna jest również linia *AREF*, do której można dołączyć zewnętrzne napięcie odniesienia dla przetwornika A/C mikrokontrolera. Do gniazda *ANALOG IN* (analogowe) dołączone zostały linie analogowe mikrokontrolera A0...A5, które również mogą pracować jako linie cyfrowe. Do gniazda *Power*

doprowadzono linie masy, napięcia zasilające 5 V i 3,3 V oraz linię zerującą *RESET*. Dostępne jest również gniazdo *ICSP*, do którego można dołączyć zewnętrzny programator. Umożliwia on załadowanie do pamięci mikrokontrolera programu Bootloadera lub zaprogramowania go dowolnym innym programem. Ponadto zestaw wyposażono w diodę sygnalizującą zasilanie, diody sygnalizujące transmisję z wykorzystaniem interfejsu RS232 oraz diodę „L” sygnalizującą stan linii 13 mikrokontrolera. Do dostępnych gniazd z liniami portów i zasilających będzie możliwość dołączania własnych lub dostępnych modułów Avtduino. Więcej informacji o zestawie Arduino UNO można znaleźć z Elektronice Praktycznej 01/2011.

### Uruchomienie zestawu

Zestaw może być zasilany z użyciem zewnętrznego zasilacza lub z interfejsu USB. Po połączeniu zestawu Arduino UNO z komputerem za pomocą przewodu USB należy w pierwszej kolejności zainstalować sterowniki USB wirtualnego portu COM. Sterowniki te znajdują się w pakiecie Arduino w katalogu *Drivers*. Po prawidłowym zainstalowaniu w menedżerze urządzeń powinny się pojawić dwa urządzenia zaznaczone na **rysunku 10**. W przykładzie został zainstalowany wirtualny port *COM5*, za pomocą któ-

rego będzie się odbywała komunikacja z zestawem Arduino UNO.

Należy jeszcze odpowiednio skonfigurować oprogramowanie Arduino IDE. W menu *Tools->Board* należy wybrać zestaw pokazany na **rysunku 11**. Następnie ustawić numer portu, po którym będzie się odbywała komunikacja. Prawidłową konfigurację dla zainstalowanego portu (w tym przypadku port *COM5*) pokazano na **rysunku 12**.

Po wykonaniu opisanych nastaw oprogramowanie Arduino IDE może się już komunikować z zestawem Arduino UNO, który został już wyposażony z odpowiedni Bootloader. Prawidłowa komunikacja będzie sygnalizowana za pomocą diod TX oraz RX. Aby przesłać do zestawu przygotowany program, po jego weryfikacji i kompilacji wystarczy przycisnąć przycisk *Upload*. Jeśli będą problemy z komunikacją może to być wina sprzętu lub nieprawidłowej konfiguracji portu komunikacyjnego. Zainstalowany numer portu powinien być zgodny z wybranym portem w oprogramowaniu Arduino IDE.

### Przykładowy program

Aby sprawdzić prawidłową współpracę oprogramowania Arduino IDE z zestawem Arduino UNO należy przepisać program z **listingu 1**. Powoduje on miganie diody „L” dołączonej do wyprowadzenia 13 mikrokon-

trolera. Dioda miga w jednosekundowych odstępach.

Po przepisaniu przykładowego programu (**rysunek 13**) należy w pierwszej kolejności wykonać weryfikację oraz kompilację klikając na ikonie *Verify/Compile*. Po bezbłędnej kompilacji (gdy jest brak komunikatów o znalezionych błędach) program można przesłać do mikrokontrolera naciskając przycisk *Upload*. Po przesłaniu programu do mikrokontrolera powinna zacząć migać dioda oznaczona jako „L”. Można zmienić wartości opóźnień w instrukcji *Delay* i zobaczyć jak zacznie się zachowywać dioda LED L

### Podsumowanie

W ramach tej części kursu pokazano funkcje oprogramowania Arduino IDE oraz pokazano sposób konfigurowania i komunikację z zestawem Arduino UNO wraz z przykładowym programem testowym. W następnych częściach kursu będą przedstawiane przykłady obsługi różnych peryferiów mikrokontrolera i dołączonych do niego modułów wraz ze szczegółowym opisem ich działania. Przykłady będą pomocne podczas przygotowywania własnych programów, będzie je można szybko zmodyfikować i dostosować do własnych potrzeb.

Marcin Wiązania  
marcin.wiazania@ep.com.pl

REKLAMA



**ul. Grabiszyńska 240**  
53-235 Wrocław

tel. (0-71) 339 00 29  
339 00 30  
faks (0-71) 339 05 01  
lemibis@lemi.pl

**złącza HDC**



**złączki listwowe**



**przyciski sterownicze**



**przełączniki elektromagnetyczne**



**SSR**



**przełączniki czasowe**



**czujniki indukcyjne i pojemnościowe**



**czujniki fotoelektryczne**



**regulatory temperatury PID**



**impulsowe zasilacze przemysłowe**





**www.lemi.pl**

SKLEP INTERNETOWY 24h

SPRZEDAŻ PEŁNEGO ASORTYMENTU Z MAGAZYNU ♦ NAJLEPSZE CENY NA RYNKU

♦ POSZUKUJEMY DYSTRYBUTORÓW LOKALNYCH  
♦ DOSKONAŁE WARUNKI HANDLOWE  
♦ DUŻE RABATY

**Energooszczędne komputery jednopłytkowe do zastosowań wbudowanych**

2010  
Certified SW Partner  
**ADVANTECH**

- > Wbudowana pamięć RAM o pojemności 256 lub 512 MB
- > Wyjście wideo VGA, 24-bit TTL oraz 24-bit LVDS (praca w trybie VGA + TTL, VGA + LVDS)
- > Możliwość pracy w zakresie temperatur od -40 do +85°C (tylko PCM-3343Z2)
- > Złącze PC/104 (pełna ISA), podstawa pod kartę CompactFlash
- > Dwa porty Ethernetowe 10/100 Mbps, 4 x COM
- > Wbudowany Watchdog Timer
- > Wsparcie dla następujących systemów operacyjnych:
  - DOS, FreeDOS
  - Windows CE 5 / 6
  - Windows XP / XPE / WES2009
  - QNX 6.3.2
  - Linux 2.4 / 2.6 (np. Debian 4, Ubuntu 8.04)

Sercem modułu jest procesor SoC DM&P Vortex86DX pracujący z częstotliwością 1 GHz. Vortex86DX jest układem typu System-on-a-Chip wypełni zgodnym z architekturą x86. Pod względem wydajności jednostka centralna jest nieznacznie słabsza od popularnego na rynku komputerów embedded procesora AMD Geode LX800 500 MHz. Największą zaletą procesora jest jego niska cena oraz niewielki pobór mocy, dzięki czemu moc pobierana przez moduł mieści się w granicach 4-6W.

**www.elmark.com.pl**

EL MARK Automatyka sp. z o.o.  
05-075 Warszawa-Wesoła ul. Niemcewicza 76  
Tel. (022) 773-79-37; Fax (022) 773-79-36  
elmark@elmark.com.pl

