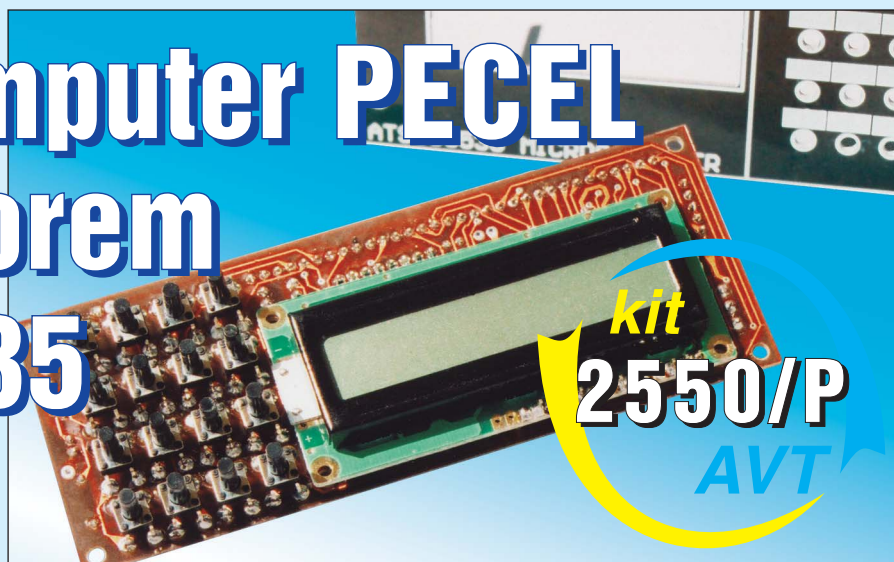


Mikrokomputer PECEL z procesorem AT90S8535

Część 4



Oczywiście, program przedstawiony w poprzednim odcinku ma tylko jedno zastosowanie: pokazanie że nawiązanie łączności pomiędzy dwoma komputerami jest możliwe i sprawdzenie poprawności konfiguracji używanych do transmisji narzędzi. Poza tym nie służy on do niczego. Aby jednak móc napisać bardziej rozbudowany program, musimy wreszcie zapoznać się choćby z podstawowymi poleceniami programowymi języka MCS BASIC służącymi przekazywaniu informacji poprzez łącze RS232.

Podstawowymi poleceniami języka MCS BASIC stosowanymi podczas wymiany danych poprzez interfejs RS232 są:

PRINT [zmienna, wartość, zmienna tekstowa lub tekst]

pozwalające wysłać do portu szeregowego komputera dowolną wartość, zmienną liczbową lub tekstową, oraz

INPUT [opcjonalny tekst zachęty], [zmienna liczbowa lub tekstowa]

które umożliwią „ręczne” przesłanie informacji poprzez port szeregowy do systemu mikroprocesorowego. Jeżeli polecenie INPUT zostanie zastosowane łącznie z tekstem zachęty podanym w cudzysłowie, to tekst ten zostanie wyświetlony na ekranie terminala komputera.

Czy jednak to drugie polecenie naprawdę działa? Aby to sprawdzić, napiszmy sobie prosty programik:

```

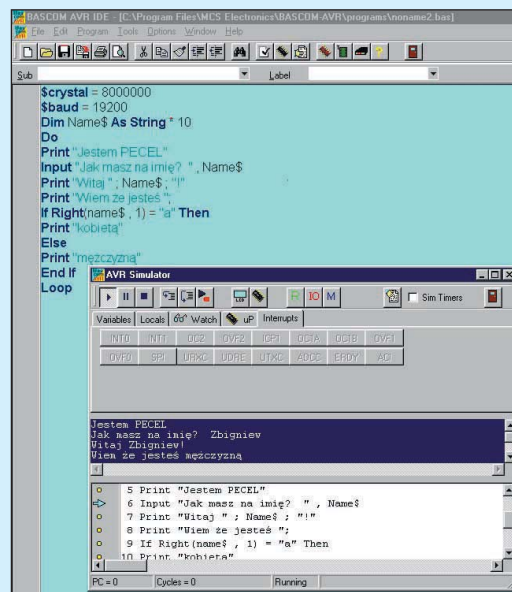
$crystal = 8000000
$baud = 19200
Dim Name$ As String * 10
Do
Print " Jestem PECEL"
Input "Jak masz na imię? ", Name$
Print "Witaj "; Name$; "!"
Print "Wiem że jesteś ";
If Right(name$, 1) = "a" Then
Print "kobieta"
Else
Print "mężczyznę"
End If
Loop
    
```

Napisany program kompilujemy i na wszelki wypadek testujemy w symulacji programowej (rysunek 20). Uruchamianie emulatora programowego niczym się nie różni od podobnej operacji dokonywanej w środowisku BACOM-a 8051 i którą opisywaliśmy w BASCOM College. Jeżeli nie popełniliśmy błędu, to powinniśmy nawiązać z PECEL-em dwustronną łączność.

Aby się upewnić, czy nasz program działa poprawnie otwieramy okienko monitora portu RS232 i odpowiadamy na pytanie zadane przez PECEL-a. Tekst wprowadzamy z klawiatury komputera, a następnie potwierdzamy podanie imienia za pomocą klawisza ENTER (rysunek 21). Czytelnikom pozostawiam odpowiedź na pytanie, w jaki sposób procesor określa płeć rozmówcy i jakie trzy męskie imiona mogą wprowadzić go w błąd.

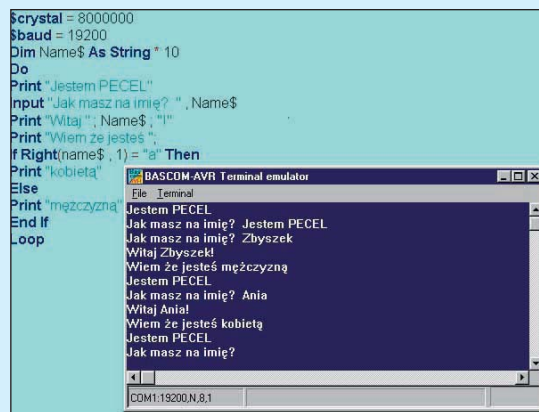
No tak, przełomowy moment mamy już za sobą: PECEL potrafi porozumiewać się z PECEL-em! Pozostaje jednak otwarte pytanie, do czego to można wykorzystać? Przecież chyba nie do pisania prostych, zabawkowych programików? Otóż, z pewnością komunikacja pomiędzy dwoma komputerami nie będzie wykorzystywana tylko do błahych spraw. Wprost przeciwnie, wykorzystując narzędzia, z którymi zapoznaliśmy się przed chwilą, będziemy mogli zbudować, a właściwie zaprogramować wiele „bardzo poważnych” urządzeń, a przede wszystkim zestaw przyrządów laboratoryjnych o ogromnych możliwościach. Tematowi temu poświęcona będzie „większa połowa” kolejnej części tego artykułu, na razie, pamiętając

że musimy jeszcze poruszyć temat obiecanej niespodzianki, podam Wam tylko jeden przykład. Obiecuję, będzie to przykład wyjątkowo spektakularny!



Rys. 20

Rys. 21



Mam nadzieję, że dysponujecie choćby jednym egzemplarzem popularnego termometru cyfrowego typu DS1820? Jeżeli nie, to warto zakupić nawet kilka sztuk tych tanich i niezwykle użytecznych elementów. Przydadzą się nam wielokrotnie, nie tylko podczas nauki programowania PECEL-a. Jeden taki termometr podłączamy do złącza CON6 minikomputera, dokładnie tak, jak pokazano na rysunku 22. Następnie piszemy kolejny program, kompilujemy go i programujemy nim procesor. Programu tego nie będę komentował, ponieważ procedury odczytu danych z termometrów DS1820 zostały już opisane w ramach kursu BASCOM College.

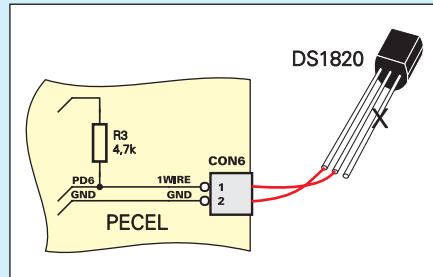
W tym memencie muszę wspomnieć o jednej, dość wstydliwej sprawie. Moje, legendarne już rozartgnienie dało jeszcze raz o sobie, tym razem owocując przeoczeniem pewnego elementu, którego umieszczenie w konstrukcji PECEL-a byłoby jak najbardziej wskazane. Mam tu na myśli przycisk służący do ręcznego resetowania procesora. W warunkach normalnej eksploatacji taki element nie byłby specjalnie użyteczny, ale podczas prowadzenia eksperymentów jego zastosowanie może znacznie usprawnić pracę. Wyłączanie i ponowne włączanie zasilania w celu rozpoczęcia pracy programu jest dość uciążliwe,

a dodanie przycisku RESET, niekoniecznie umieszczonego na płycie czołowej będzie czynnością bardzo prostą. Jako taki element możemy wykorzystać zwykły microswitch lub dowolny inny przycisk monostabilny o niewielkich wymiarach, dołączony do PECEL-a zgodnie ze schematem pokazanym na rysunku 23.

Otwieramy teraz po raz kolejny okienko terminala RS232 i albo naciskamy dodany do układu przycisk RESET, albo wyłączamy i ponownie włączamy zasilanie PECEL-a. Na ekranie monitora ukaże się napis zachęcający do naciśnięcia klawisza ENTER, co też bez obaw możemy natychmiast uczynić.

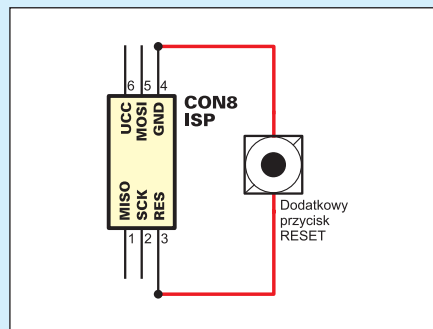
Zgodnie z przewidywaniami na ekranie rozpoczęło się cykliczne wyświetlanie zmierzanej przez DS1820 temperatury. Ponieważ w programie zabrakło poleceń przeliczają-

cych wynik pomiaru, wyświetlane wartości są dziesięciokrotnie zawyżone, tj. zamiast np. 28,3 stopni wyświetlane jest 283. Nie przejmujemy się tym jednak, za chwilę okaże się, że w niczym nie będzie nam to przeszkadzać. Pobawmy się teraz chwilę tak wykonanym termometrem, sprawdzając czy prawidłowo reaguje na podgrzanie i ochładzanie czujnika.



Rys. 22

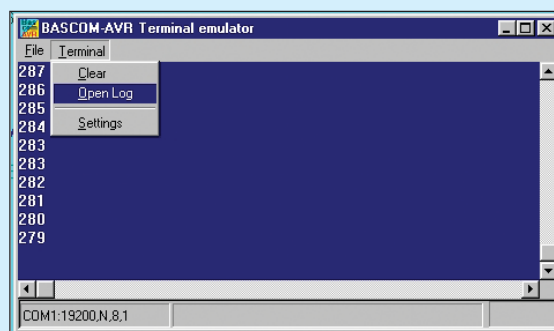
Rys. 23



Myślę, że część Czytelników jest nieco rozczarowana: tyle zachodu, aby zbudować prosty termometr! Poczekaście jednak chwilę, a już teraz mogę Wam przypomnieć, że zbudowaliśmy termometr, ale nie taki znowu byle jaki: do PECEL-a możemy przecież dołączyć absolutnie dowolną ilość czujników DS1820, pracujących na jednym, wspólnym przewodzie. To tego tematu powrócimy jeszcze w przyszłości, a na razie chciałbym pokazać Wam coś innego.

Otwórzmy teraz po raz kolejny okienko terminala RS232 i zajmijmy się dodatkowymi przyciskami umieszczonymi na jego górnej krawędzi (rysunek 24). Najbardziej powinien nas zainteresować przycisk OPEN

Rys. 24



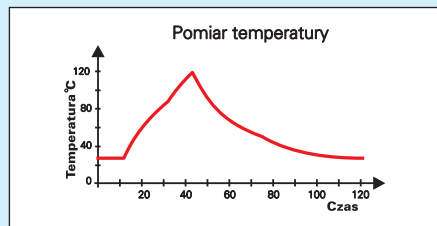
LOG, ponieważ otwiera on drogę do nieznanych dotąd, rewelacyjnych możliwości BASCOM-a. Wykonajmy kolejno następujące czynności:

1. Zresetujmy minikomputer bądź za pomocą dodanego przycisku, bądź za pomocą wyłączenia i włączenia zasilania.
2. Kliknijmy na przycisk OPEN LOG. Spowoduje to pojawienie się na ekranie okienka, w którym musimy podać nazwę pliku, do którego zapisywane będą wszelkie dane przechodzące przez monitorowany port COM.
3. Po nadaniu nazwy pliku zamykamy okienko OPEN LOG i naciskamy klawisz ENTER. Od tego momentu wszystkie informacje ukazujące się na ekranie monitora będą także zapisywane w pliku o podanej przez nas nazwie.
4. Pomęczmy teraz trochę nasz termometr. Osobiście polecałbym serię sadystycznych eksperymentów polegających na przypiekaniu go lutownicą lub innym gorącym przedmiotem. Zwracamy jednak uwagę, aby temperatura czujnika nie przekroczyła 120 stopni (na ekranie liczba 1200!).
5. Po upływie 1 ... 2 minut kliknijmy ponownie na przycisk na krawędzi terminala i tym razem wybierzmy opcję CLOSE LOG.

Możemy teraz zapoznać się z treścią pliku, w którym zapisaliśmy wyniki pomiarów temperatury. To już zaczyna być interesujące: mamy tam dokładny zapis zmian temperatury odbywającego się na określonym odcinku czasu, a pomiary dokonywane były mniej więcej co 1 sekundę. Oczywiście, ten skromny sposób zapisu możemy metodami programistycznymi dowolnie rozbudować. Możemy do każdego pomiaru dodać informację o czasie jego dokonania, możemy dowolnie zmieniać częstotliwość dokonywanych pomiarów, możemy też wreszcie zastosować dowolną ilość czujników i ich wyniki zapisywać w oddzielnych kolumnach. Teraz chyba mogliście zorientować się, jakie możliwości daje transmisja danych z PECEL-a do komputera PC. W najbliższej przyszłości zaprogramujemy PECEL-a tak, aby stał się wszechstronnym laboratoryjnym przyrządem pomiarowym. Połączenie z komputerem da nam wtedy możliwość zapisywania wyników wszelkich pomiarów i archiwizowania ich w celu późniejszego wykorzystania. Ale czy tylko archiwizowania? Przecież dane uzyskane z PECEL-a możemy poddać dalszej obróbce, wykorzystując w tym celu arkusze kalkulacyjne czy też edytory graficzne pracujące pod kontrolą MS WINDOWS. Nie mogę się po prostu powstrzymać, aby nie zaprezentować Wam kolejnego, spektakularnego pokazu możliwości, jakie dostaliśmy do rąk.

Dane opisujące barbarzyński eksperyment z przypiekaniem nieszczęsnego czujnika lutowanicą przeniosłem jako plik ASCII do arkusza kalkulacyjnego MS EXCEL. Następnie wyniki pomiarów zostały podzielone przez 10. Wykonanie z tak przetworzonych danych wykresu sprowadziło się już tylko do kilku kliknięć myszką, a efekt wszystkich tych operacji nie trwających dłużej niż minutę został pokazany na **rysunku 25**. Wygląda ciekawie, prawda? Na wykresie widać nawet krótki moment wahania, w którym chciałem dać już spokój dręczonemu czujnikowi, ale ostatecznie postanowiłem torturować go nadal.

Rys. 25



Moi Drodzy, to tylko prosty, najprostszy przykład możliwości PECEL-a używanego w roli inteligentnego terminala komputerowego. W najbliższym czasie zajmiemy się szerzej tym tematem, ale dopiero po omówieniu wszystkich (lub prawie wszystkich) metod programistycznych stosowanych przy pisaniu programów na nasz minikomputer.

Do tej pory mówiliśmy o dwukierunkowej transmisji danych pomiędzy komputerem a procesorem wyłącznie w kontekście ewentualnej budowy urządzeń wykorzystujących taką wymianę informacji. Istnieje jednak jeszcze jedno zastosowanie łącza RS232, genialnie upraszczające odpluskwanie i testowanie pisanego oprogramowania. Transmisję szeregową możemy wykorzystać jako narzędzie do „podglądania” pracującego programu, i to „na żywo”, w jego naturalnym środowisku. Wystarczy nieraz, w punkcie programu, którego działania nie jesteśmy pewni, dopisać instrukcje wysyłające na ekran terminala np. informacje o wartości pewnych zmiennych, od których w decydujący sposób zależy działanie programu. Z kolei, jeżeli chcemy przetestować program zmieniając „zdalnie” parametry jego pracy, to stosując instrukcję INPUT możemy w wybranych momentach zmieniać wartości wybranych zmiennych.

Omawianie podstawowych zagadnień związanych z transmisją danych poprzez złącze RS232 zajęło nam tyle miejsca, że już niewiele go pozostało na opisanie niespodzianki, jaką dla Was przygotowałem. Tak więc z konieczności omówimy tę sprawę w największym skrócie, pozostawiając resztę do następnego numeru EdW.

Opis PECEL-a nie jest BASCOM College i nie mam prawa zadawać Wam jakichkol-

wiek ćwiczeń czy prac domowych do odrobienia. Mam jednak prośbę: może zechcielibyście w wolnej chwili przepisać do edytora BASCOM-a program, którego listing został zamieszczony poniżej.

Program ten nie jest mojego autorstwa, nie mogę więc go ani zmieniać ani komentować. Tak więc, po prostu przepiszcie go, nie wnikając na razie w jego treść. Następnie poproszę Was o skompilowanie tego programu i wpisanie go do pamięci procesora. Uprezdam, że bezpośrednio po wykonaniu tej czynności spotka Was przykre rozczarowanie: PECEL nie będzie dawał żadnych widocznych z zewnątrz oznak „życia”. Następnie połączcie ponownie PECEL-a z komputerem za pomocą złącza RS232 i na wszelki wypadek odłączmy od niego kabel programatora.

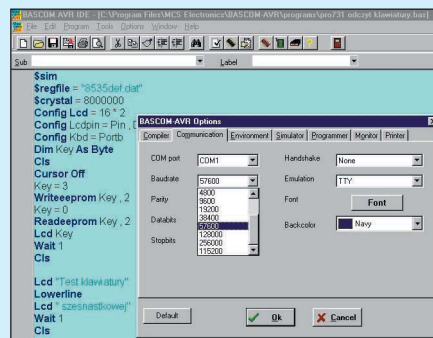
```

'MONITOR SYMULACJI
SPRZĘTOWEJ
$regfile = "8535def.dat"
$crystal = 8000000
$baud = 57600
Dim Krk As Byte
Dim Adr As Word
Dim Adrl As Byte , Adrh As Byte
Dim VI As Byte
Do
  Krk = Inkey()
  If Krk = "T" Then
    Print Chr(13);
  Elseif Krk = "W" Then
    Adr = Waitkey()
    Out Adr , VI
    Print Chr(13);
  Elseif Krk = "R" Then
    Adr = Waitkey()
    VI = Inp(adr)
    Print Chr(vi);
  Elseif Krk = "O" Then
    Adrl = Waitkey()
    Adrh = Waitkey()
    VI = Waitkey()
    Adr = Adrh * 256
    Adr = Adr + Adrl
    Out Adr , VI
    Print Chr(13);
  End If
Loop
    
```

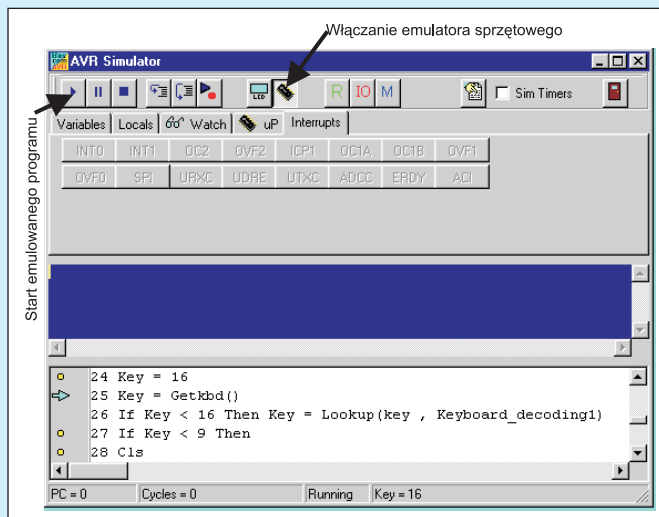
Kolejną czynnością będzie napisanie prostego programu, wykorzystującego np. wyświetlacz alfanumeryczny LCD i klawiaturę szesnastkową. Nie musi to być jakiś nowy i skomplikowany program, na początek zupełnie wystarczy prosty programik, który już wykorzystywaliśmy do demonstracji obsługi wyświetlacza LCD i klawiatury. **Tu bardzo ważna uwaga: na samym początku programu napiszcie "\$SIM"!** Następnie skompilujcie ten program, ale **w żadnym wypadku nie ładujcie go do pamięci procesora!**

Kolejną czynnością będzie wykonanie drobnej zmiany w konfiguracji BASCOM-a. Musimy zmienić uprzednio ustawioną prędkość transmisji danych poprzez złącze RS232 z 19200 na 57600 baud (**rysunek 26**). No i wreszcie dochodzimy do końca: pozostało nam już tylko otwarcie okienka symulatora (klawisz F2), kliknięcie na ikonę włączania symulacji sprzętowej (**rysunek 27**)

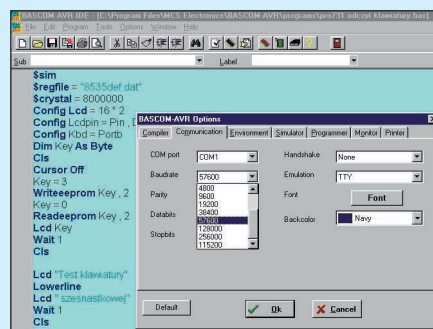
i naciśnięcie na strzałkę włączającą symulację. Następnie możemy już przeprowadzić testy klawiatury za pomocą uniwersalnego emulatora sprzętowego, który w tym momencie dostaliśmy do dyspozycji. Nie będzie już potrzebne wielokrotne przeprogramowywanie procesora w celu przetestowania drobnych zmian w programie. Wszystkie testy będziemy mogli przeprowadzić w „real world”



Rys. 26



Rys. 27



Rys. 28

wyłącznie w symulacji sprzętowej. Poruszony temat jest tak obszerny, że jego kontynuację odkładamy do dalszej części artykułu o minikomputerze PECEL.

Zbigniew Raabe,
zbigniew.raabe@edw.com.pl