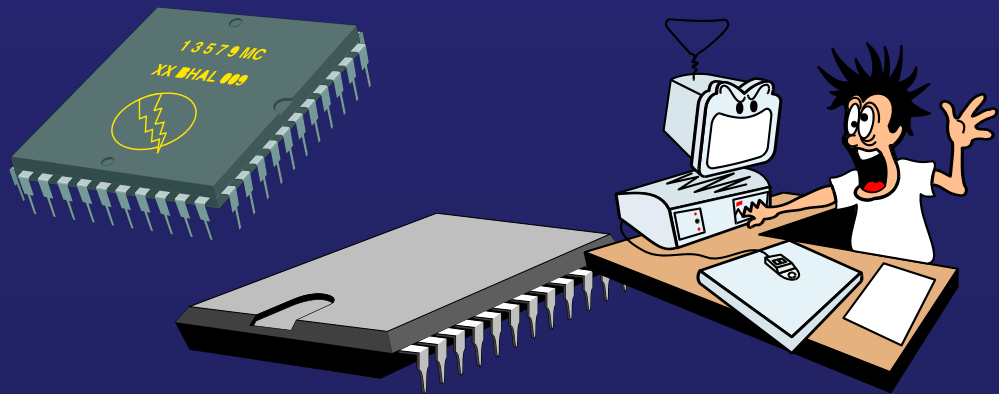
WYDZIAŁ FIZYKI
i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Systemy wbudowane

Witold Kozłowski

Zakład Fizyki i Technologii Struktur Nanometrowych
90-236 Łódź, Pomorska 149/153

https://std2.phys.uni.lodz.pl/mikroprocesory/



Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Uwaga !!!

Proszę o wyłączenie telefonów komórkowych

na wykładzie i laboratorium

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Wykład 1.

Wprowadzenie do mikroprocesorów

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

System wbudowany (ang. *Embedded system*) – system komputerowy specjalnego przeznaczenia, który staje się integralną częścią obsługiwanego przez niego sprzętu.

System wbudowany spełnia określone wymagania, zdefiniowane do zadań, które ma wykonywać. Nie można nim więc nazywać typowego wielofunkcyjnego komputera osobistego. Każdy system wbudowany oparty jest na mikroprocesorze (lub mikrokontrolerze), zaprogramowanym do wykonywania ograniczonej ilości zadań, lub nawet do jednego.

Zależnie od złożoności wykonywanych zadań, może zawierać oprogramowanie dedykowane wyłącznie temu urządzeniu (firmware), lub może to być system operacyjny wraz ze specjalizowanym oprogramowaniem. Może o tym decydować też stopień niezawodności jakie ma oferować dany system wbudowany. Ogólną zasadą jest, iż im mniej złożone i specjalizowane jest oprogramowanie, tym bardziej system jest niezawodny, oraz może szybciej reagować na zdarzenia krytyczne.
Niezawodność systemu może być zwiększona, przez rozdzielenie zadań na mniejsze podsystemy, a także przez redundancję. Polegać to może np. na zastosowaniu do jednego zadania dwóch identycznych urządzeń, które mogą przejąć zadania drugiego, w przypadku awarii jednego z nich.

Ciekawostka:

Za pierwszy komputer wbudowany uznaje się ten, który sterował amerykańskim statkiem kosmicznym Apollo. Pierwszy komputer wbudowany produkowany masowo sterował rakietą LGM-30 Minuteman.

Systemy wbudowane

Zastosowania

Tego typu systemy znajdują zastosowanie we wszystkich dziedzinach życia. Przykłady obszarów i urządzeń technicznych, w których stosuje się systemy wbudowane:

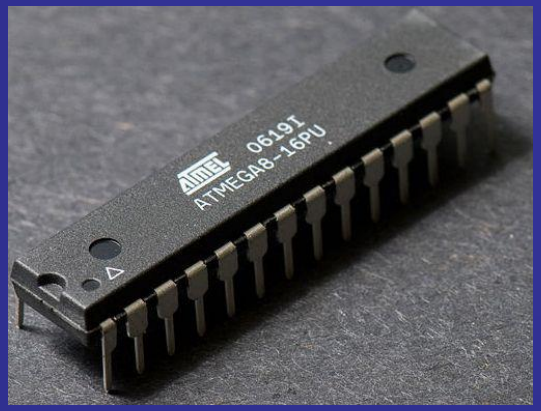
- układy sterujące pracą silnika samochodowego i ABS, komputery pokładowe
- sprzęt sterujący samolotami, rakietami, pociskami rakietowymi, bomby inteligentne
- sprzęt medyczny
- sprzęt pomiarowy, w tym między innymi: oscyloskopy, analizatory widma
- bankomaty
- termostaty, klimatyzatory,
- kuchenki mikrofalowe, zmywarki
- sterowniki PLC stosowane w przemyśle do sterowania i kontroli procesów i maszyn produkcyjnych
- sterowniki do wszelkiego rodzaju robotów mechanicznych
- systemy alarmowe służące do ochrony osób i mienia np. antywłamaniowe, przeciwpożarowe, i inne
- telefony komórkowe i centrale telefoniczne,
- drukarki, kserokopiarki
- kalkulatory
- sprzęt komputerowy, w tym między innymi: dyski twarde, napędy optyczne, routery, serwery, firewalle
- systemy rozrywki multimedialnej i interaktywnej:
    - konsole do gier: stacjonarne i mobilne
    - automaty np. do gier oraz inne
- telewizory, odtwarzacze DVD, kamery cyfrowe, magnetowidy

Systemy wbudowane

Zastosowania

**Tego typu systemy znajdują zastosowanie we wszystkich dziedzinach,
gdyż w obecnych czasach dąży się aby wszystkie urządzenia były inteligentne i zdolne do pracy autonomicznej oraz wykonywały coraz bardziej złożone zadania.**

Mikrokontrolera ATmega8



Budowa mikrokontroler AVR opiera się na architekturze harwardzkiej:

Mikroprocesory AVR należą do grupy układów o architekturze RISC (Reduced Instruction Set Computer)

- **rozdzielono przestrzeń adresowej pamięci programu i przestrzeni adresowej pamięci danych**
- **uzyskano to dzięki zastosowaniu oddzielnych magistral adresowych**
- **dzięki temu możliwe było zastosowanie słowa o różnej szerokości dla pamięci programu i pamięci danych**
- **chroni to przed przypadkowym odczytaniem danej i interpretowanie jej jako instrukcji**

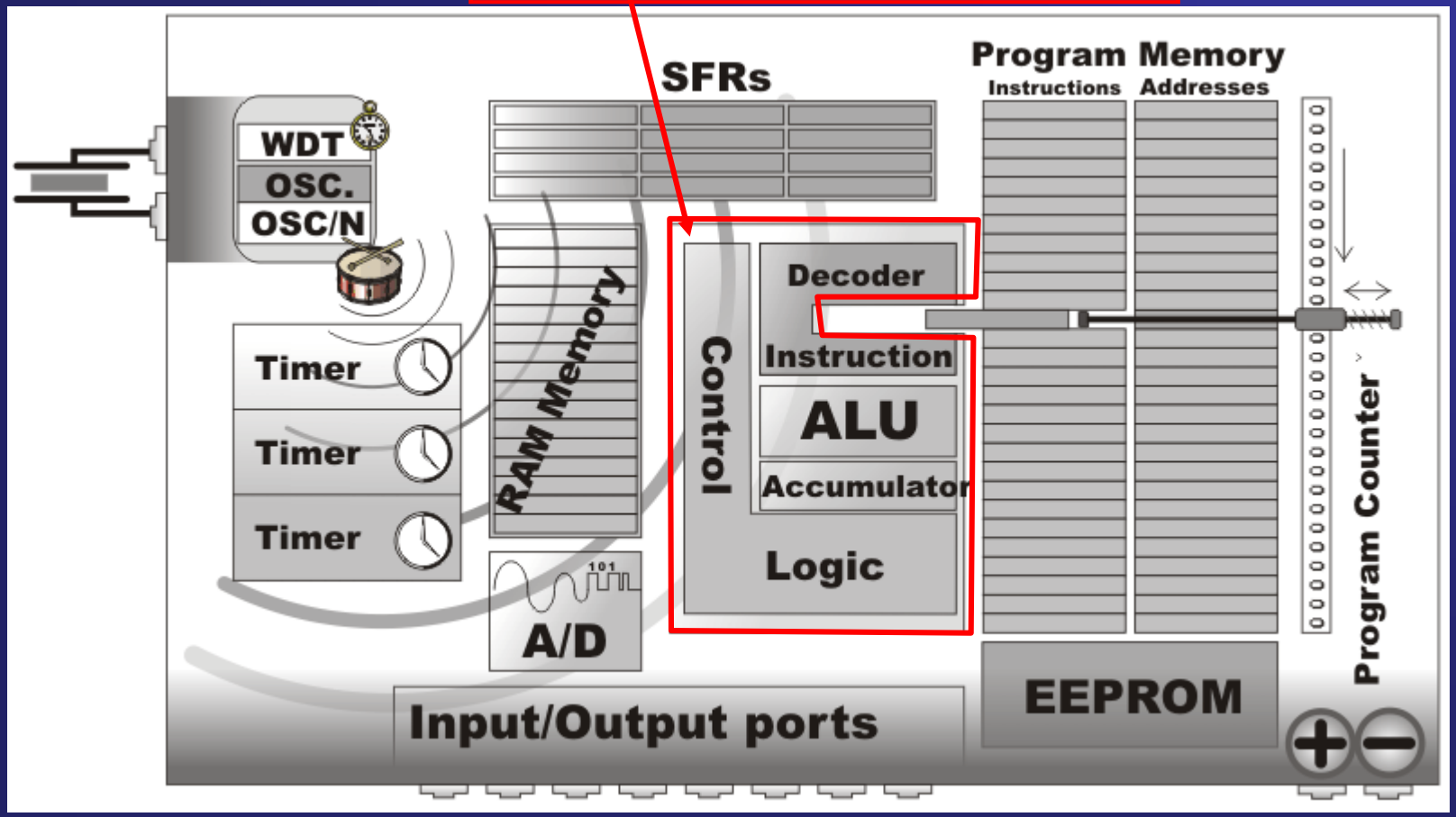
- **Większość rozkazów RISC jest realizowana w jednym takcie zegara co zapewnia szybsze wykonanie programu**
- **programy pisane dla procesorów RISC charakteryzują się większą spójnością, a co za tym idzie mniejszym kodem wynikowym**

(Architektura 8051 jest określona nazwą CISC (Complex Instruction Set Computer) wykonanie jednego rozkazu CISC wymaga zazwyczaj wykonania wielu operacji, co zwykle trwa kilka taktów zegara)

Cechą wyróżniającą mikrokontrolery AVR jest również
- **Zaimplementowanie wielu rejestrów wewnętrznych, z których każdy może pełnić rolę akumulatora podczas wykonywania operacji arytmetycznych i logicznych**
- **Minimalizuje to liczbę przesłań między rejestrowych, co korzystnie wpływa na szybkość wykonywania programu**

Schemat blokowy mikrokontrolera ATmega8

Sercem mikrokontrolera jest centralna jednostka sterująca
(CPU - Central Processing Unit)

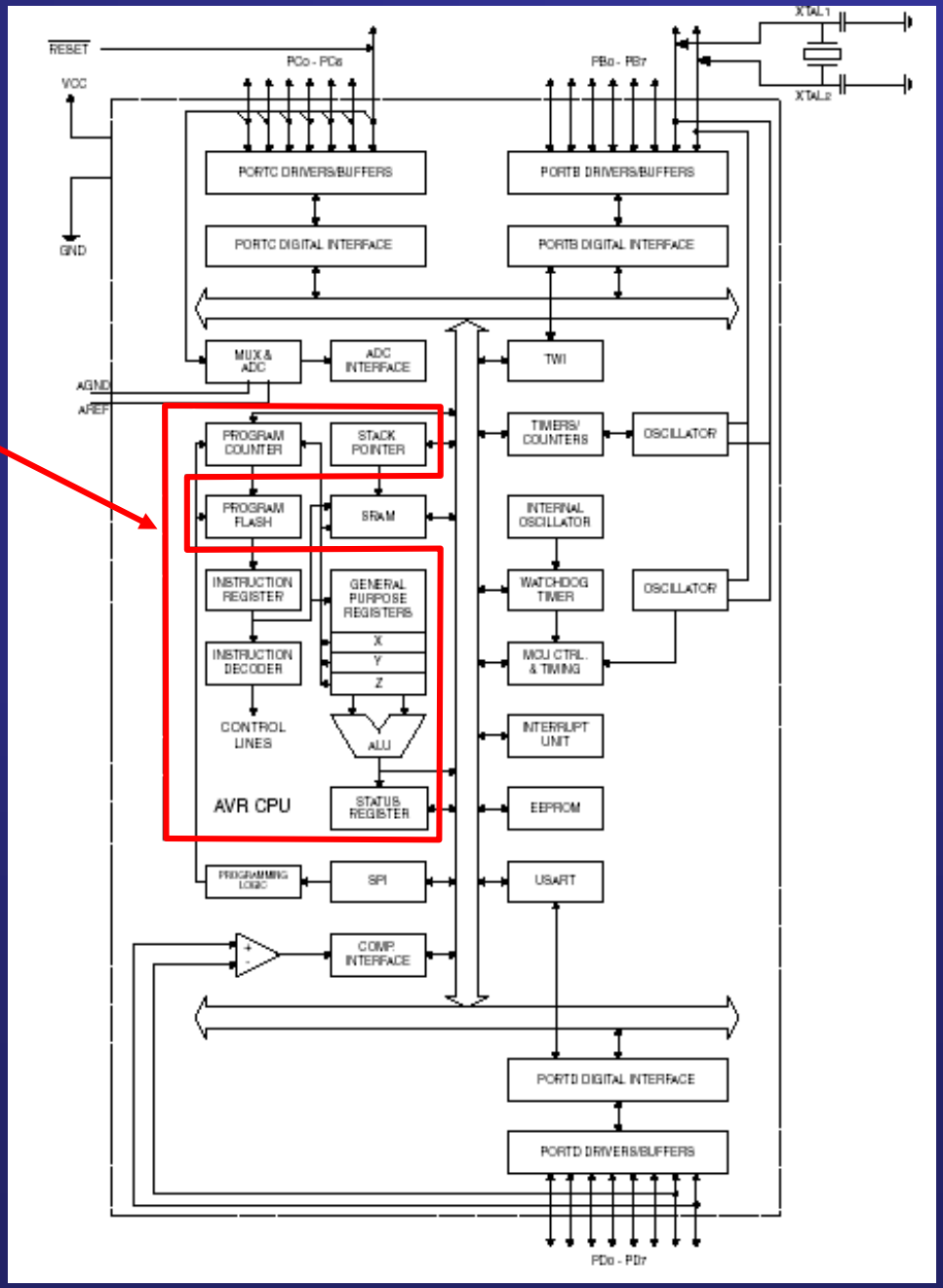


Jej zadaniem jest sterowanie procesem pobierania rozkazów, ich rozpoznawania i wykonywania.

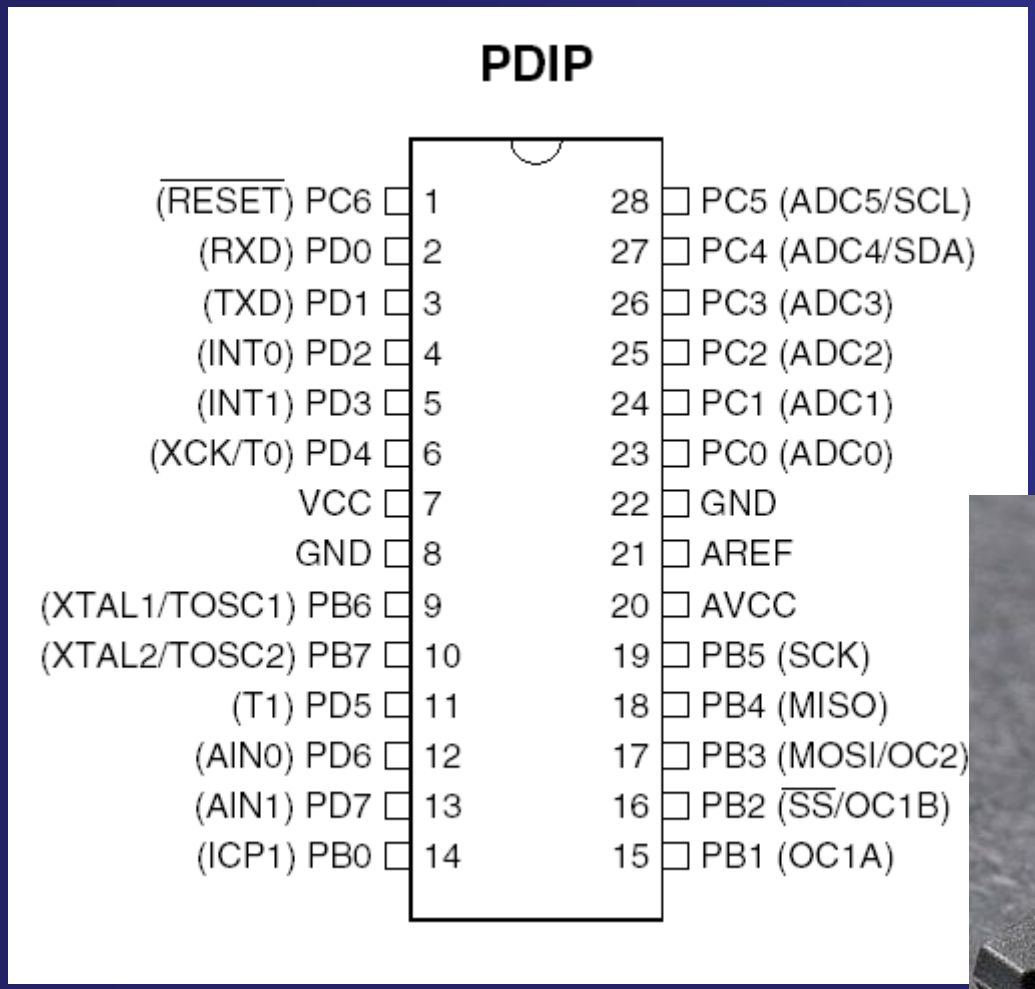
Schemat blokowy mikrokontrolera ATmega8

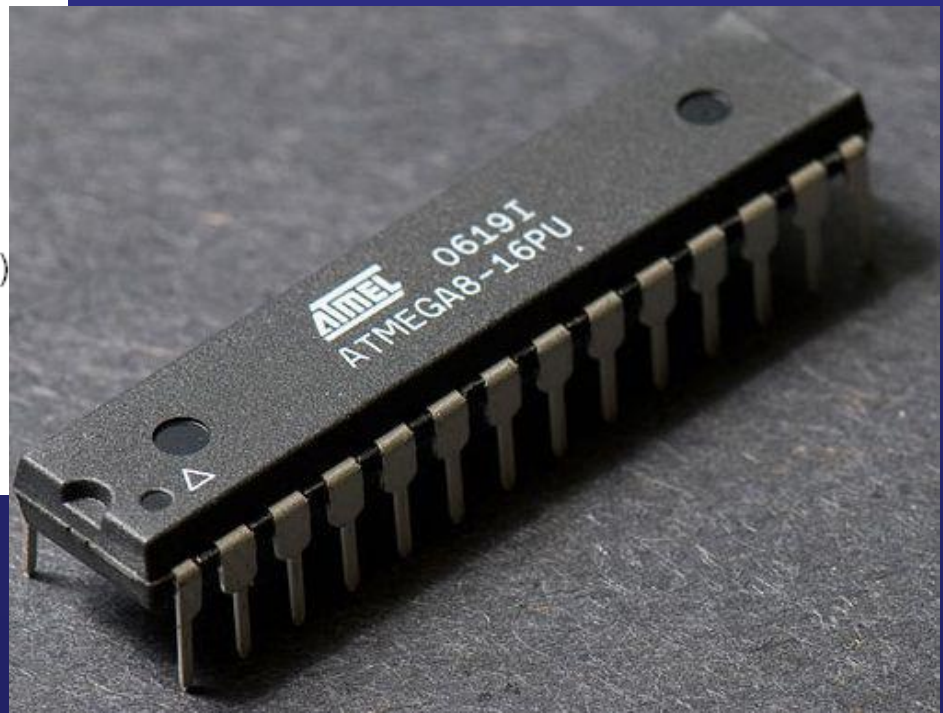
Sercem mikrokontrolera jest centralna jednostka sterująca
(CPU - Central Processing Unit)

Jej zadaniem jest sterowanie procesem pobierania rozkazów, ich rozpoznawania i wykonywania.



Rozmieszczenie wyprowadzeń Mikrokontrolera - ATmega8 w obudowie DIP 28





Mikrokontrolera ATmega8

Mikrokontroler ATmega8 ma następujące parametry oraz cechy funkcjonalne:

- **Mały pobór mocy**
- **Zaawansowana architektura RISC (Reduced Instruction Computer) charakteryzująca się:**
  - **130 instrukcjami, z których większość jest wykonywana w jednym cyklu maszynowym**
  - **32 rejestrami 8-bitowymi ogólnego przeznaczenia**
  - **Dużą wydajnością 16 MIPS (miliona operacji na sekundę) przy częstotliwości zegara 16 MHz**
- **Pamięć:**
  - **8 kB nielotnej pamięci Flash o trwałości 10 000 zapisów/kasowań,**
  - **Ma opcjonalne miejsce na Bootloader**
  - **512 B pamięci EEPROM o trwałości 100 000 zapisów/kasowań (może przechowywać dane do 10 lat)**
  - **1 kB pamięci SRAM,**
  - **Ma programowalne zabezpieczenia pamięci programu przed odczytem**
- **Układy peryferyjne:**
  - **dwa 8 – bitowe czasomierze/liczniki z oddzielnymi preskalerami,**
  - **jeden 16 – bitowy czasomierz/licznik z oddzielnym preskalerem, możliwość pracy z w trybie Capture oraz Compare**
  - **zegar czasu rzeczywistego z oddzielnym oscylatorem**
  - **trzy kanały PWM (OC1, OC1B, OC2)**
  - **6-kanałowy przetwornik A/C – cztery kanały o rozdzielczości 10 bitów i dwa o rozdzielczości 8 bitów**
  - **programowalny USART do transmisji przez RS232**
  - **szeregowy interfejs Master/Slave SPI**
  - **programowalny Watchdog z oddzielnym oscylatorem**
  - **wbudowany komparator analogowy**

Mikrokontrolera ATmega8

Mikrokontroler ATmega8 ma następujące parametry oraz cechy funkcjonalne:

- **Specjalne wyposażenie mikrokontrolera:**
  - **zerowanie po włączeniu mikrokontrolera**
  - **wewnętrzny kalibrowany oscylator RC**
  - **wewnętrzne oraz zewnętrzne źródła sygnałów przerwań**
  - **pięć trybów uśpienia mikrokontrolera**
  - **23 linie I/O dowolnego wykorzystania**
- **Zakresy napięć zasilania:**
  - **2,7 V…5,5 V (ATmega 8L)**
  - **4,5 V…5,5 V (ATmega8)**
- **Zakres częstotliwości sygnału taktującego mikroprocesor:**
  - **0…8 MHz (ATmega 8L)**
  - **0…16 MHz (ATmega8)**
- **Pobierany prąd przy częstotliwości sygnału taktowania 4 MHz i przy napięciu zaśilania 3V (ATmega8L):**
  - **w stanie aktywnym: 3,6 mA**
  - **w trybie *Idle*: 1,0 mA**
  - **w trybie *Power-down*: 0,5 uA**

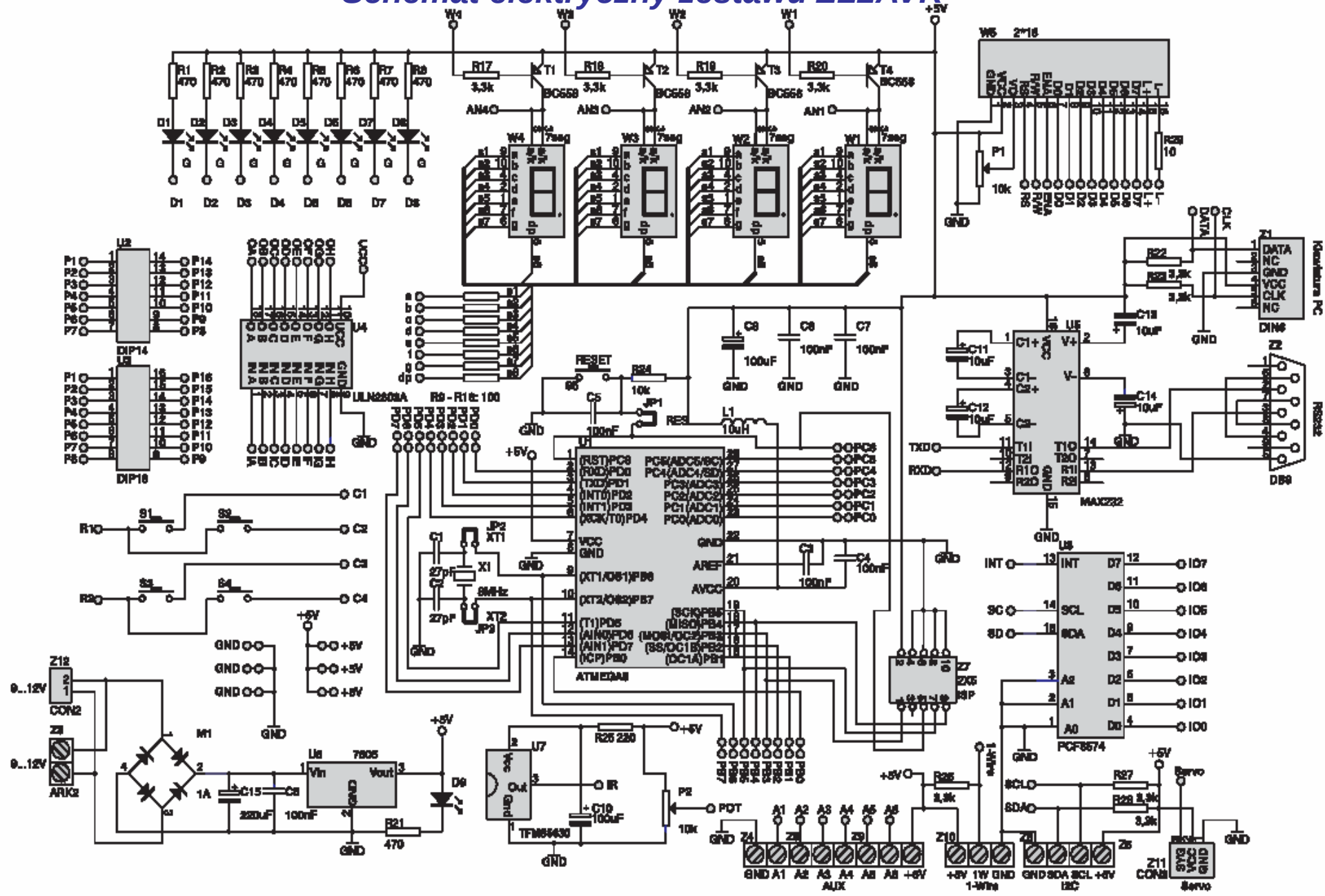
Pamięć stałamiała pojemność 74 kB. Co istotne, pojęcie bajt ***nie pojawia się w dokumentacji tego sprzętu*** *– AGC posługiwał się słowami maszynowymi mającymi długość 16 bitów. Pamięć kasowalna, czyli odpowiednik dzisiejszego RAM-u, miała pojemność 2048 16-bitowych słów, czyli 4 kB, a jednostka obliczeniowa Appolo Guidance Computer pracowała z częstotliwością 2,048 MHz.*

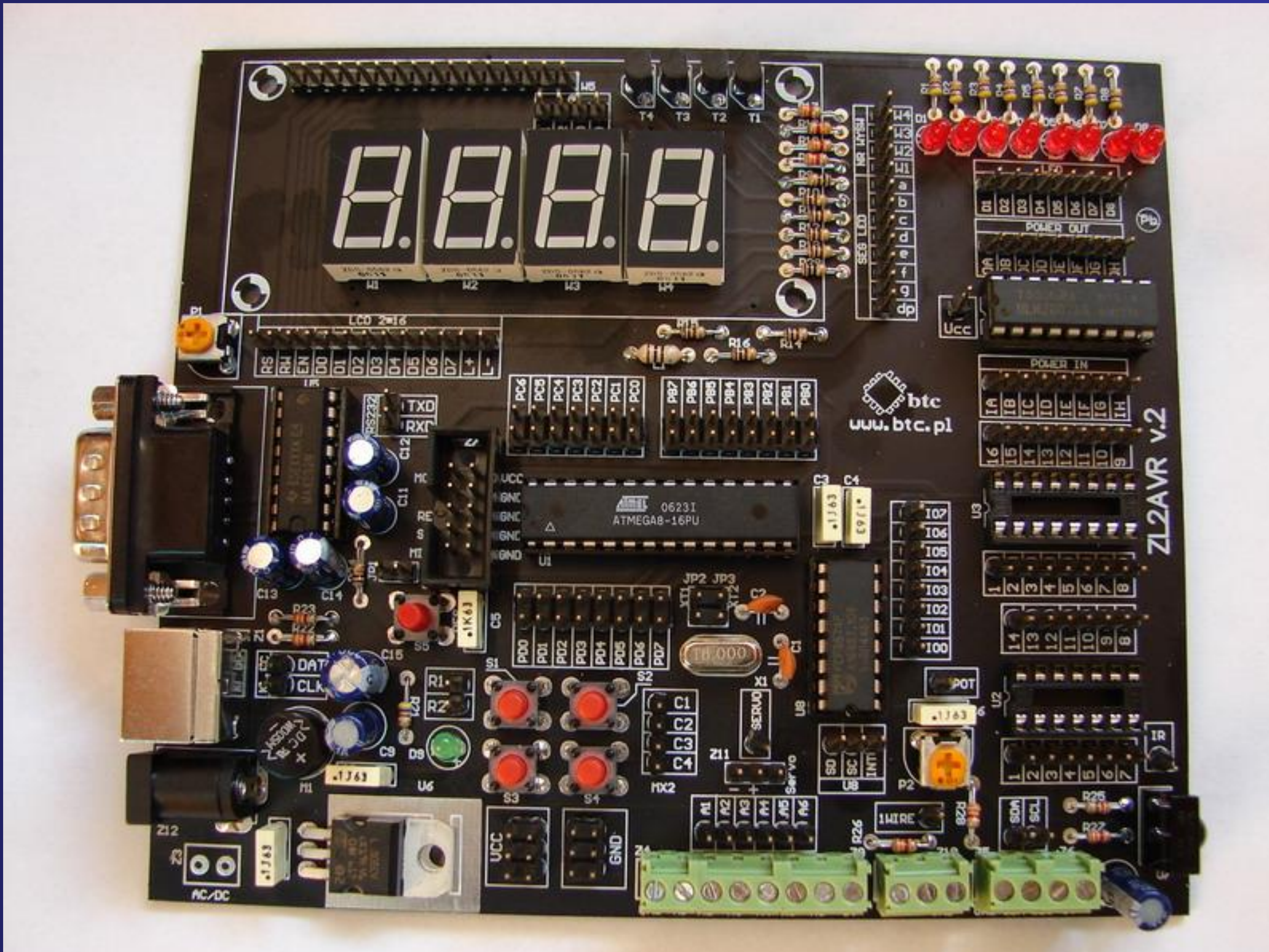

Program Apollo
Człowiek na Księżycu

ZL2AVR

Zestaw uruchomieniowy dla mikrokontrolerów AVR ATmega8

Schemat elektryczny zestawu ZL2AVR




ZL2AVR v.2
btc
www.btc.pl
ATMEGA8-16PU
LCD 2*16
POWER OUT
POWER IN
SEG LED
AC/DC
UCC
GND
SERVO
1WIRE

ZL2AVR - Zestaw uruchomieniowy dla mikrokontrolerów AVR ATmega8

1. Mikrokontroler ATmega8 (U1), który może być taktowany wewnętrznym sygnałem zegarowym lub z zastosowaniem zewnętrznego rezonatora kwarcowego X1, dołączanego za pomocą zworek JP2, JP3.

2. S5, R24, C5, JP1 – zapewniające zerowanie mikrokontrolera. Zerowanie można przeprowadzić przyciskiem S5. Zworka JP1 umożliwia dołączenie zewnętrznego obwodu zerującego do mikrokontrolera.

3. Złącze Z7, które jest złączem programatora ISP.

4. Alfanumeryczny wyświetlacz LCD o organizacji 2x16 znaków (potencjometr P1 służy do ustawienia kontrastu wyświetlacza).

5. Cztery wyświetlacze 7-segmentowe LED ze wspólną anodą. Na płytce układu uruchomieniowego można wykorzystać jeden wyświetlacz LED sterowany statycznie lub wszystkie cztery przy sterowaniu multipleksowym.

6. Interfejs szeregowy RS232, który zrealizowano z wykorzystaniem konwertera poziomów U5.

7. Port wyjściowy dużej mocy zbudowany z wykorzystaniem układu ULN2803A (U4).

8. Blok ośmiu diod LED (D1...D8).

9. Blok przycisków (S1...S4).

10. Odbiornik transmisji danych w podczerwieni U7.

ZL2AVR - Zestaw uruchomieniowy dla mikrokontrolerów AVR ATmega8

11. **Konwerter I^2C na 8-bitowy port I/O, który zrealizowano na układzie PCF8574. Adresy konwertera ustalono na stałe: adresem zapisu jest 64(dec), a odczytu 65(dec).**

12. **Złącza Z5 i Z6, które umożliwiają dołączanie zewnętrznych urządzeń sterowanych magistralą I^2C.**

13. **Złącze Z10 (1-Wire) umożliwia dołączanie elementów sterowanych za pomocą interfejsu 1-Wire.**

14. **Złącze Z11 (Servo) umożliwia np. dołączenie serwomechanizmu modelarskiego.**

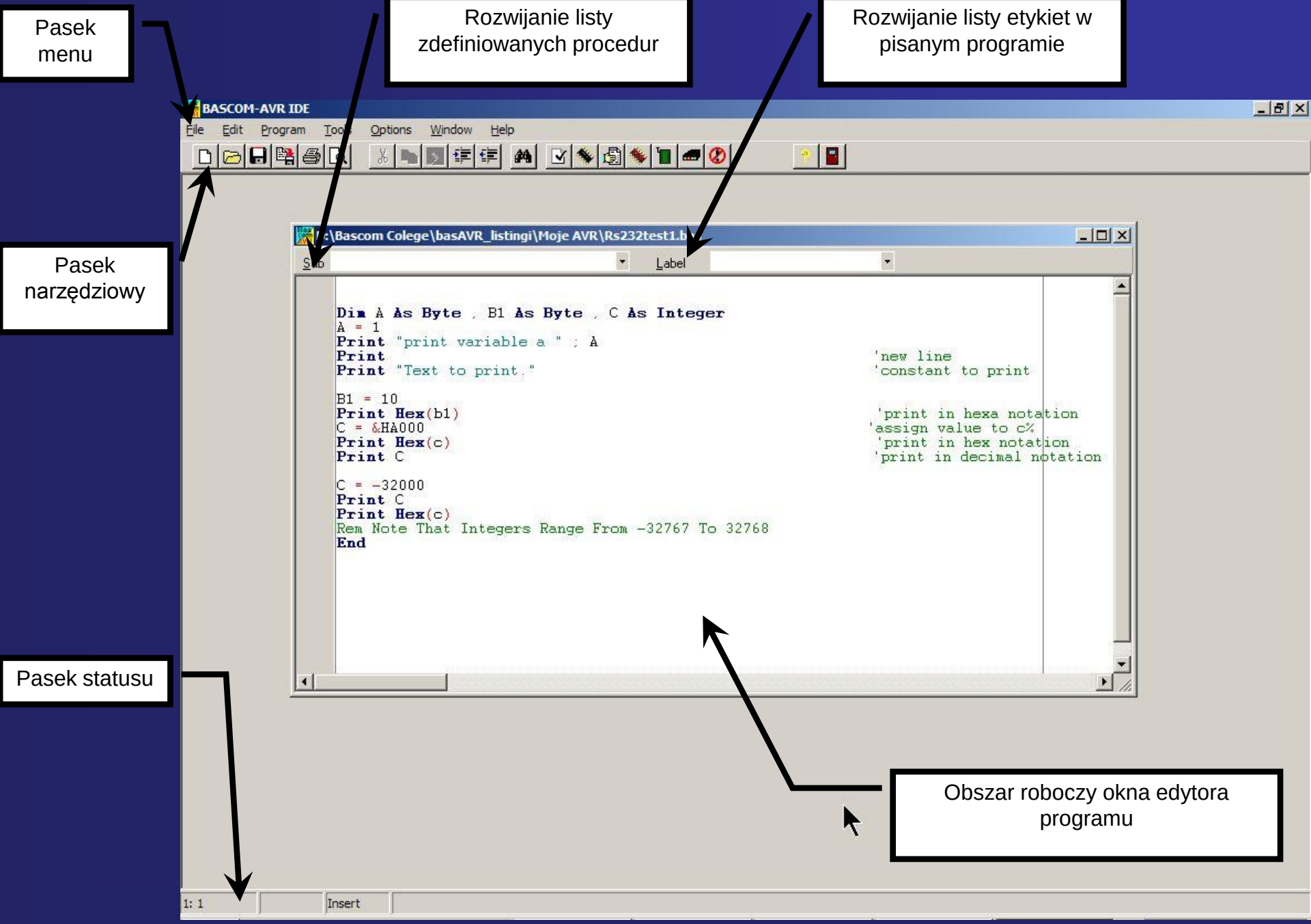
15. **Złącze Z1 umożliwia dołączenie klawiatury PS2.**

16. **Na złącza Z4, Z8 i Z9 (AUX) zostały wyprowadzone napięcia zasilające (masa i +5 V) oraz linie uniwersalne oznaczone A1...A6, które umożliwiają dołączanie elementów zewnętrznych, jak przyciski, przekaźniki itp.**

17. **Potencjometr P2 umożliwia zmienianie napięcia (w zakresie od 0 do 5 V) podawanego na wejście przetwornika A/C zawartego w mikrokontrolerze ATmega8.**

Programowanie mikrokontrolerów AVR w języku

BASCOM


Pasek menu
Rozwijanie listy zdefiniowanych procedur
Rozwijanie listy etykiet w pisanym programie
BASCOM-AVR IDE
File Edit Program Tools Options Window Help
C:\Bascom Colege\basAVR_listingi\Moje AVR\Rs232test1.b
Sub
Label
Dim A As Byte , B1 As Byte , C As Integer
A = 1
Print "print variable a " ; A
Print 'new line
Print "Text to print." 'constant to print
B1 = 10
Print Hex(b1) 'print in hexa notation
C = &HA000 'assign value to c%
Print Hex(c) 'print in hex notation
Print C 'print in decimal notation
C = -32000
Print C
Print Hex(c)
Rem Note That Integers Range From -32767 To 32768
End
Pasek narzędziowy
Pasek statusu
Obszar roboczy okna edytora programu
1: 1
Insert

Pasek narzędziowy programu AVR

Nowy program

Otwieranie programu

Wycina zaznaczony tekst

Kopiuje zaznaczony tekst

Wkleja zaznaczony tekst

Akapit tekstu

Uruchamia emulator terminala

Zeruje układ

zapis

Zapis w nowym pliku

Drukuje zawartość pliku

Podgląd wydruku

Znajduje fragment tekst

Sprawdza składnie programy

Kompiluje program

Pokazuje raport kompilacji

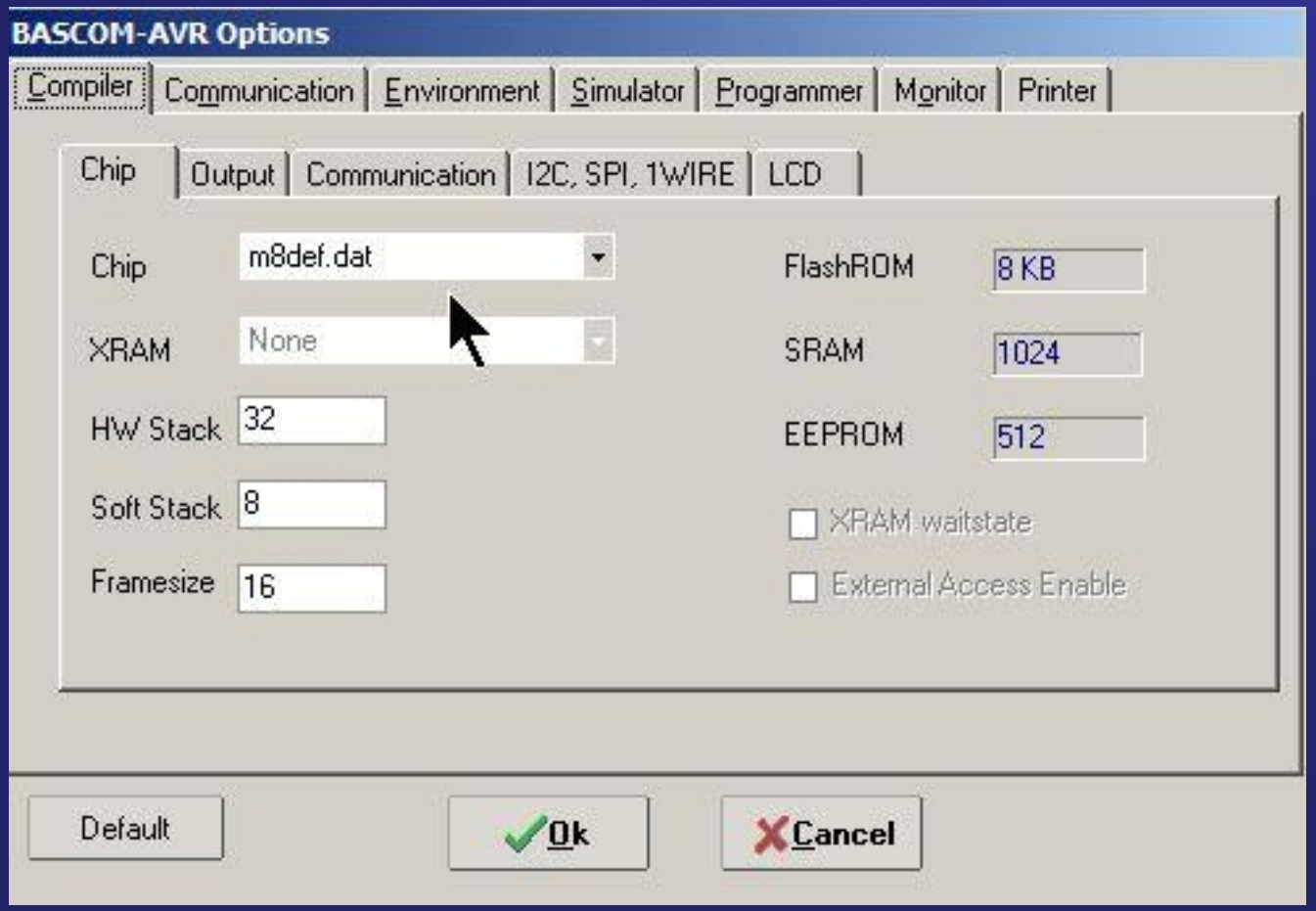
Uruchamia symulacje programu

Uruchamia programator

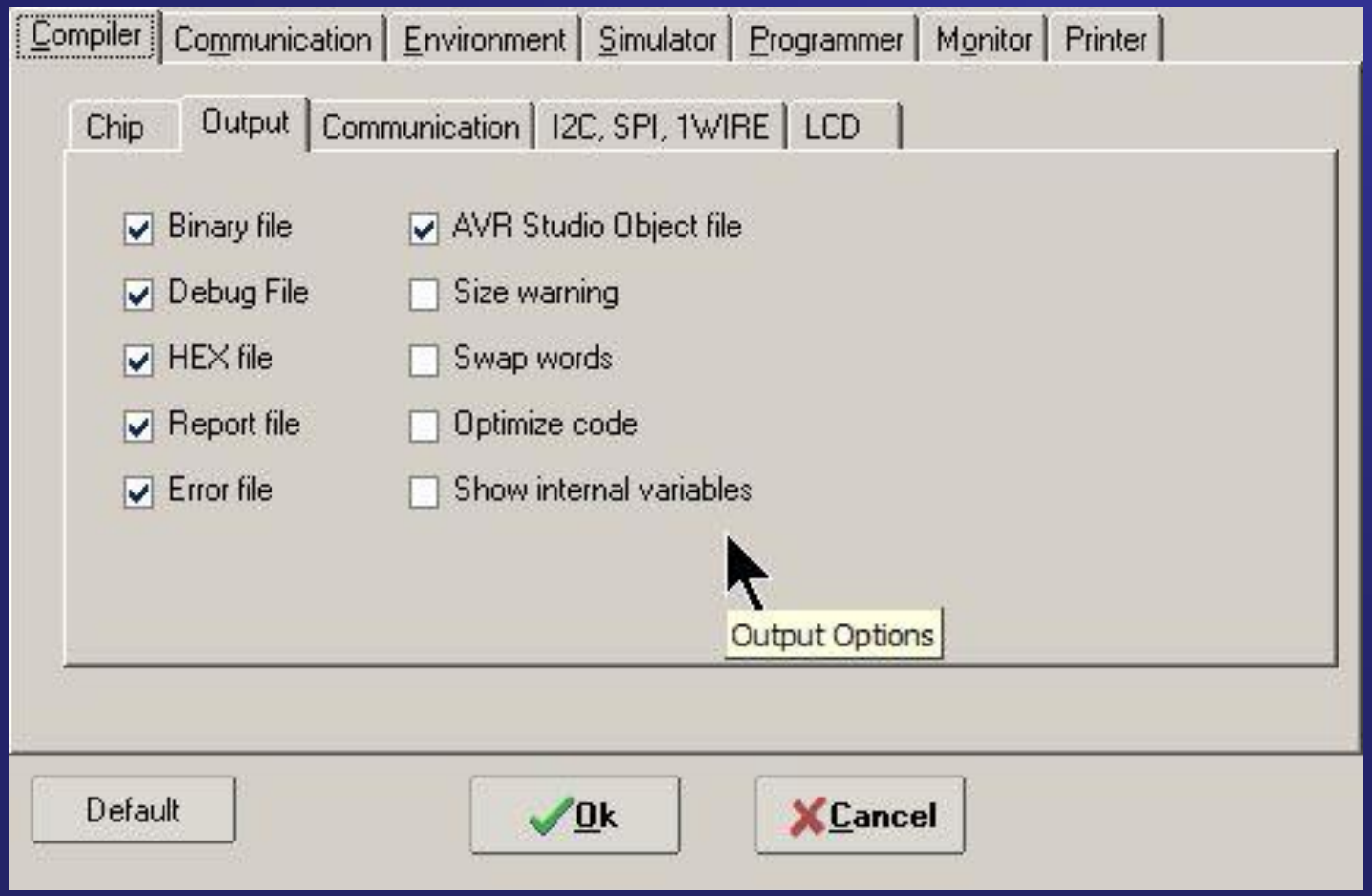
Kończy prace programu

Pomoc programu

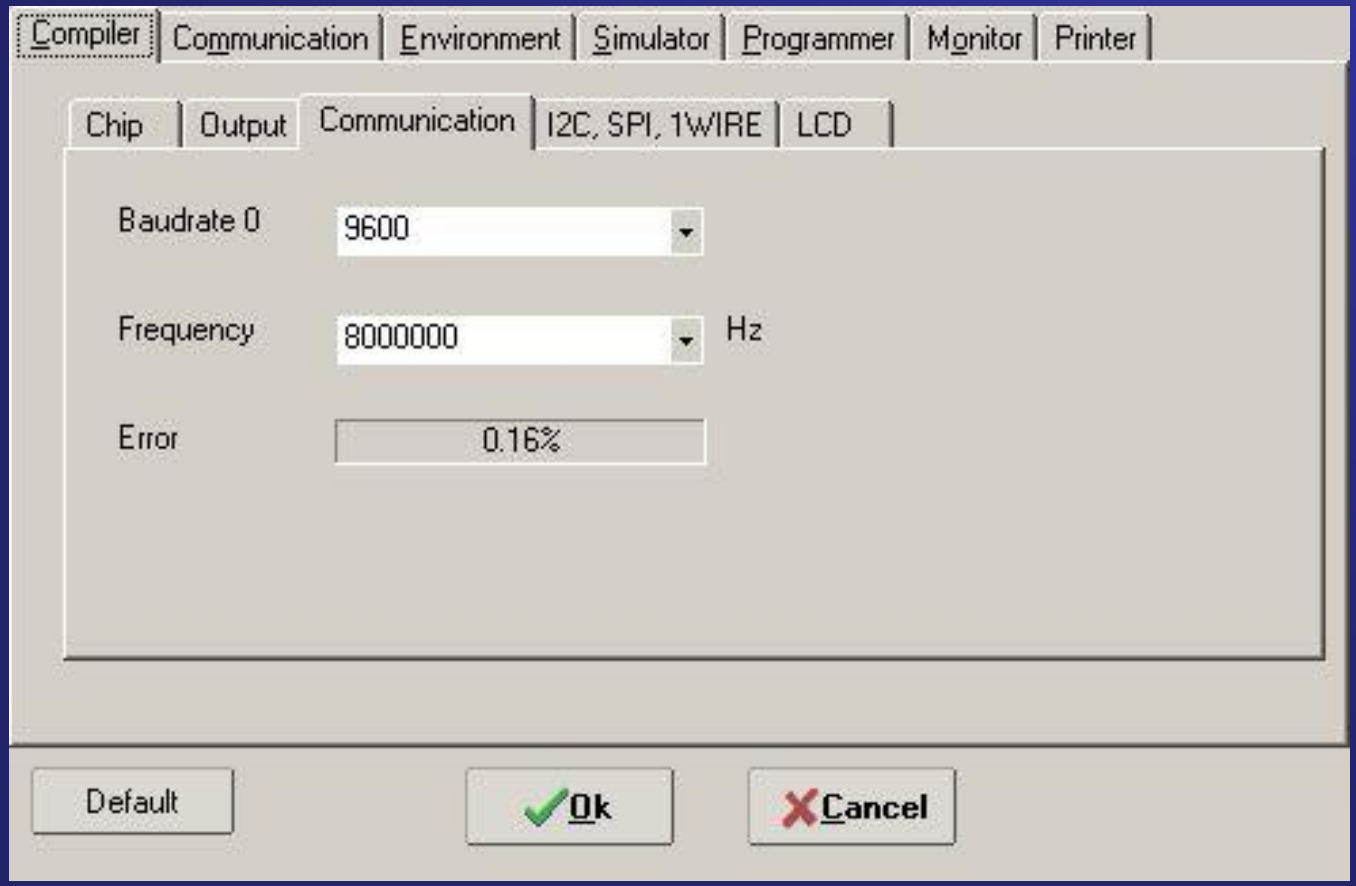
Konfiguracja programu Bascom AVR



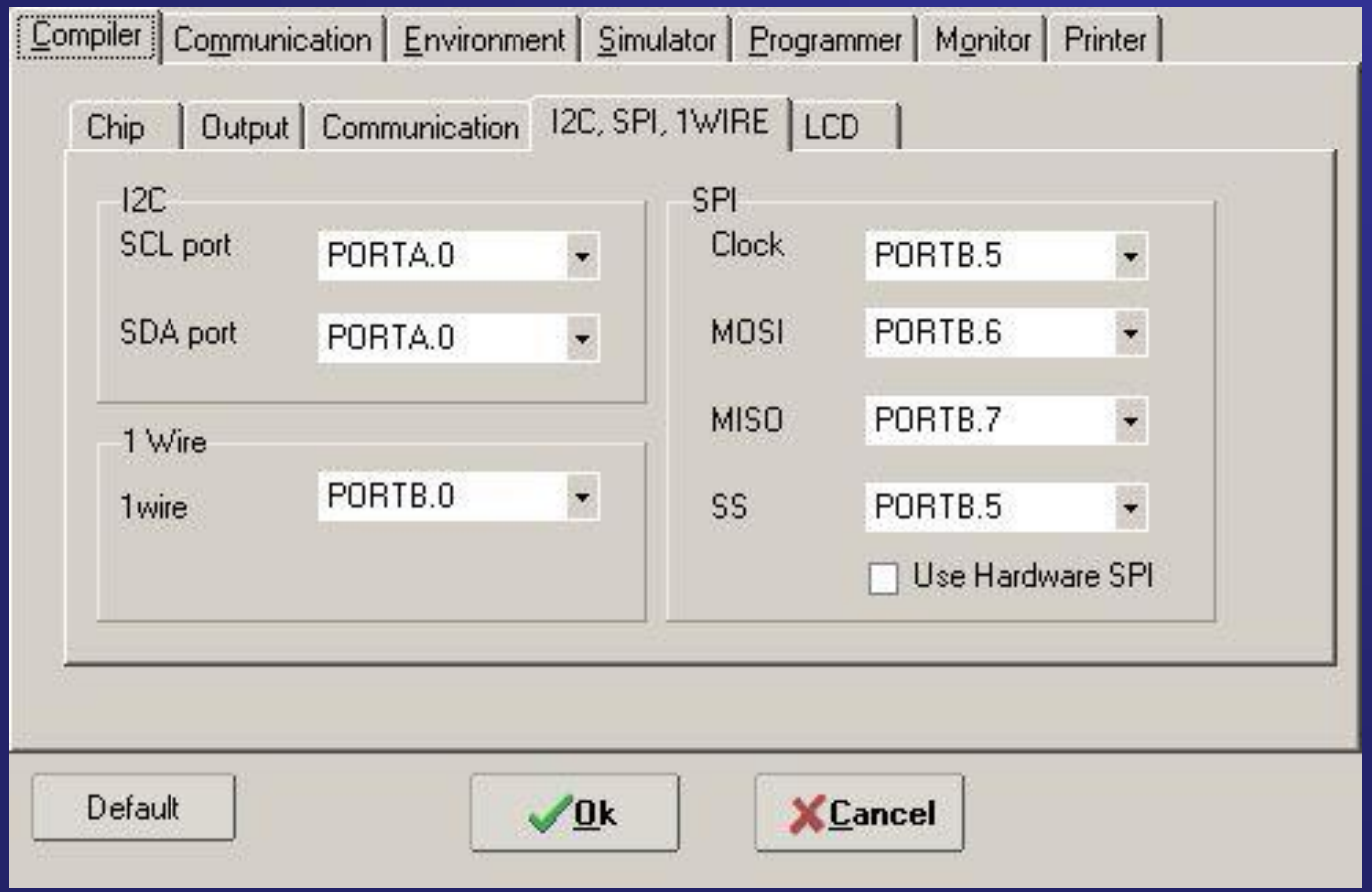
Konfiguracja programu Bascom AVR



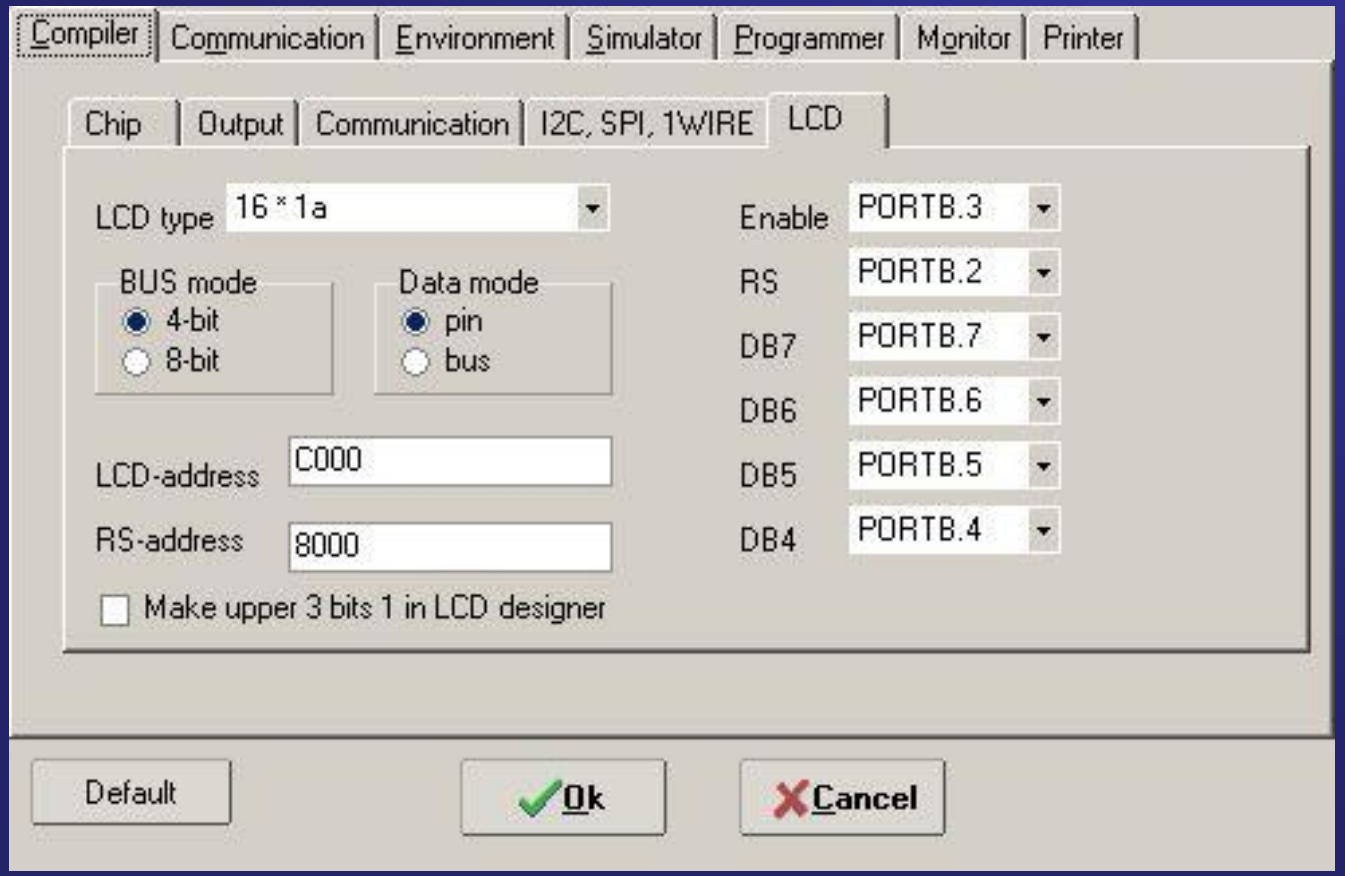
Konfiguracja programu Bascom AVR



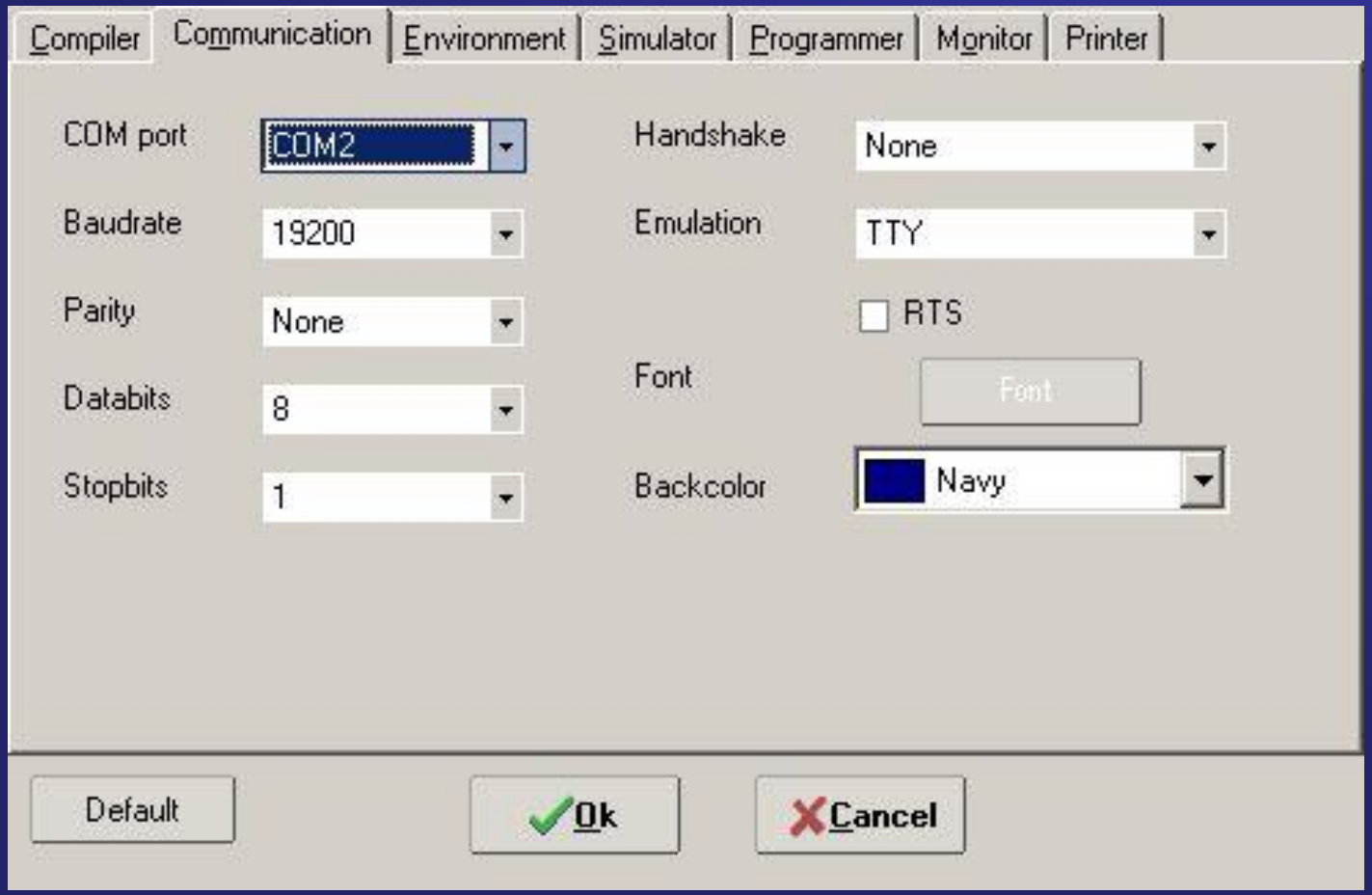
Konfiguracja programu Bascom AVR



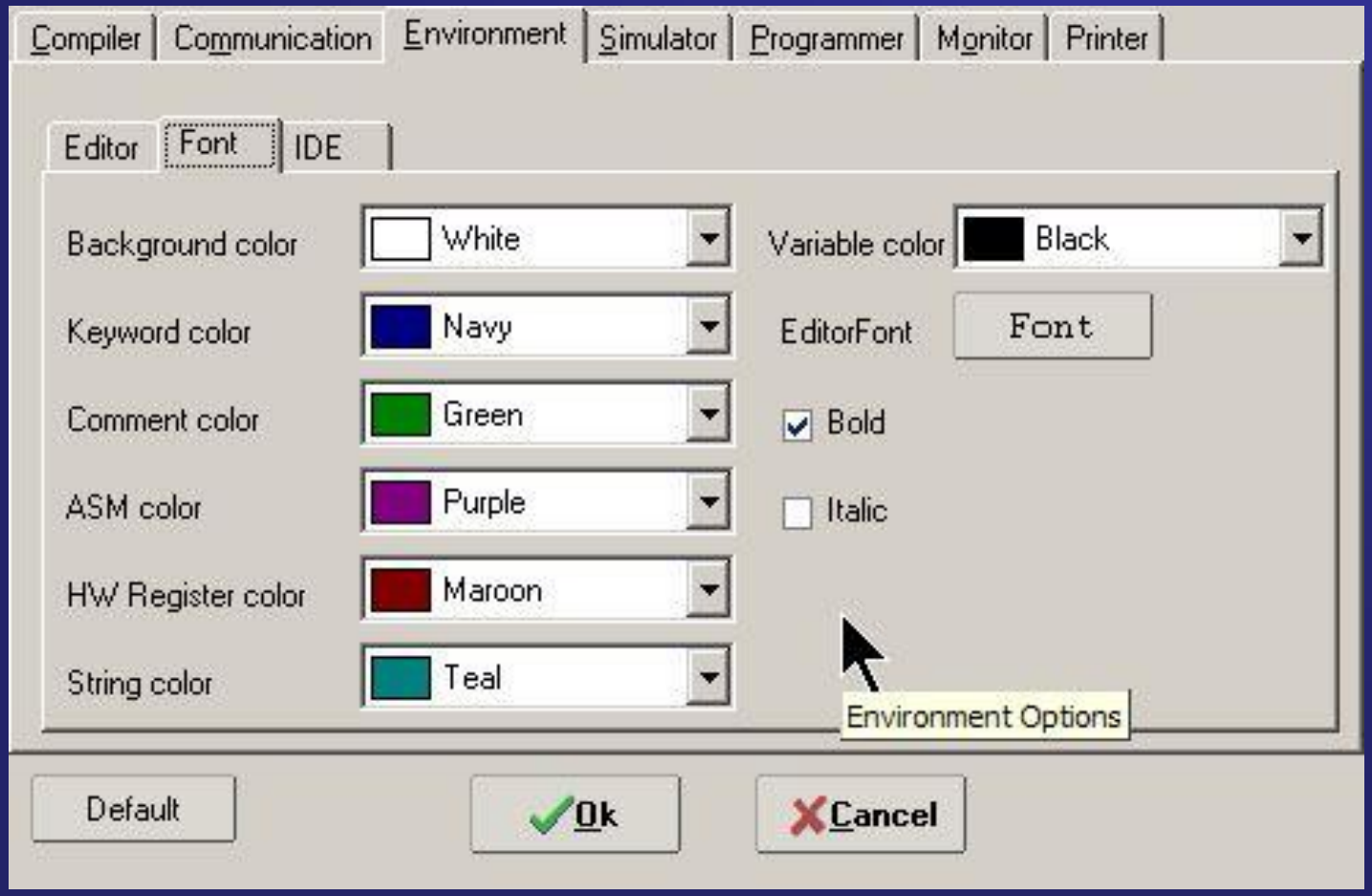
Konfiguracja programu Bascom AVR



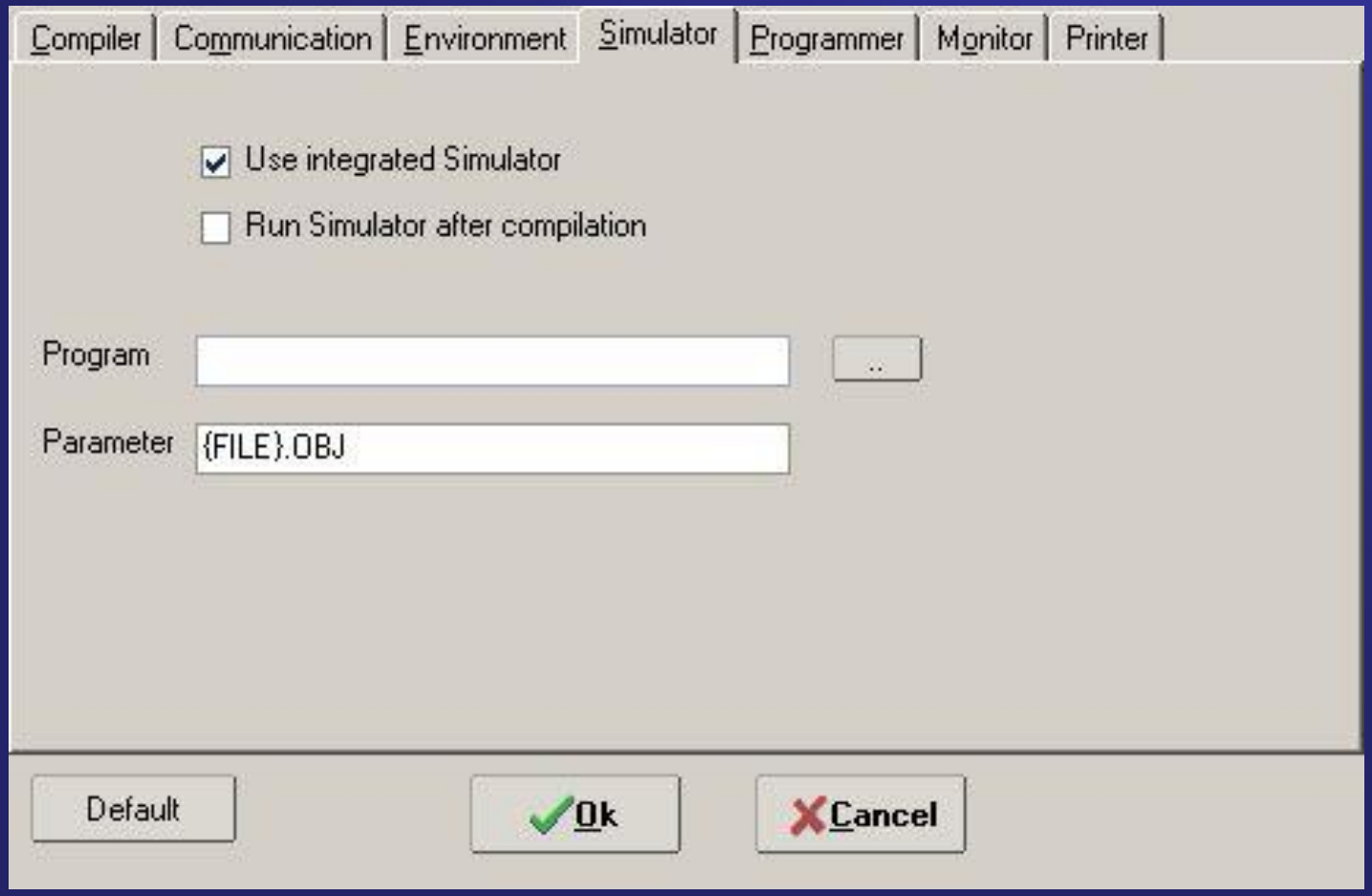
Konfiguracja programu Bascom AVR



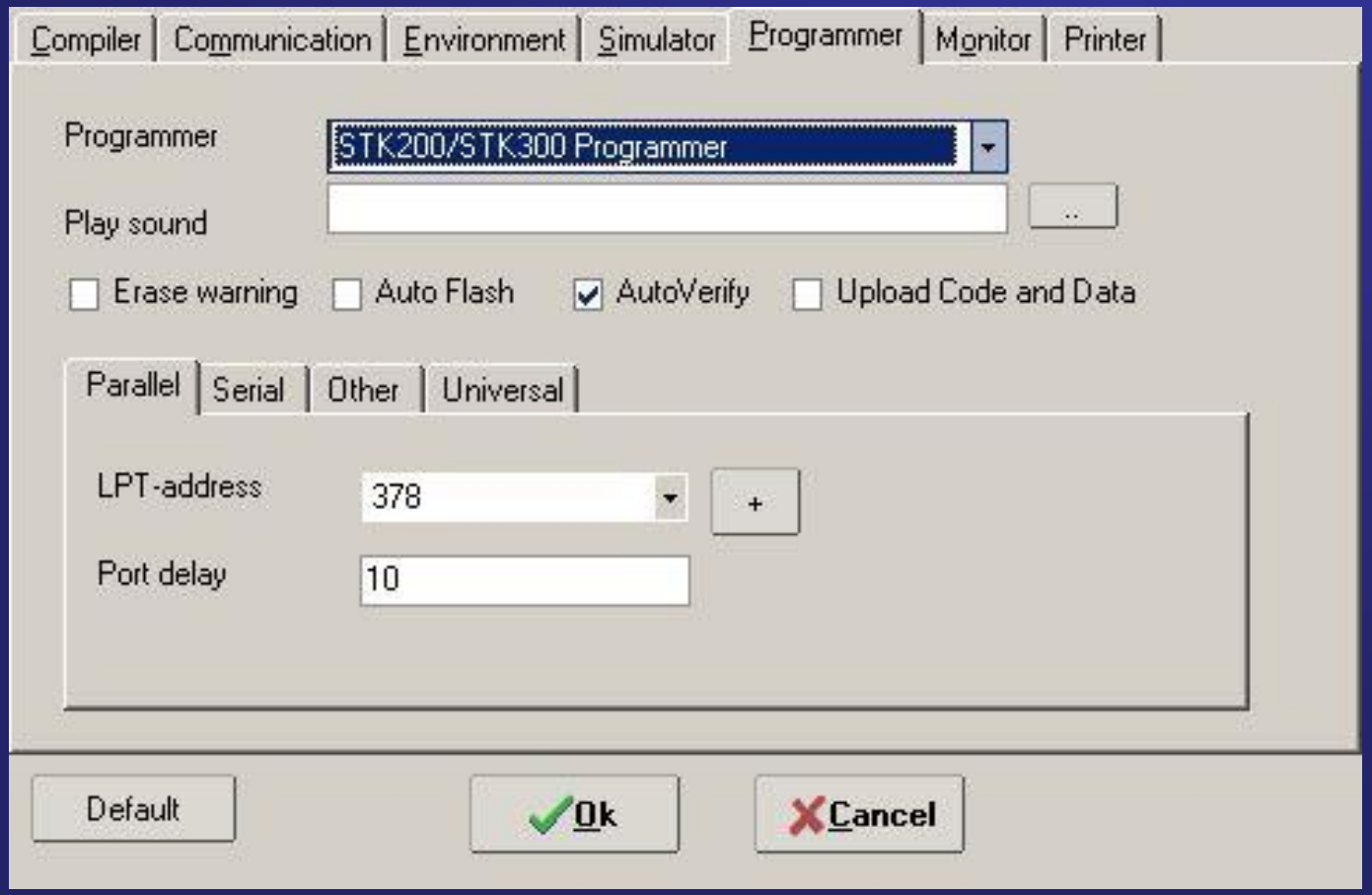
Konfiguracja programu Bascom AVR



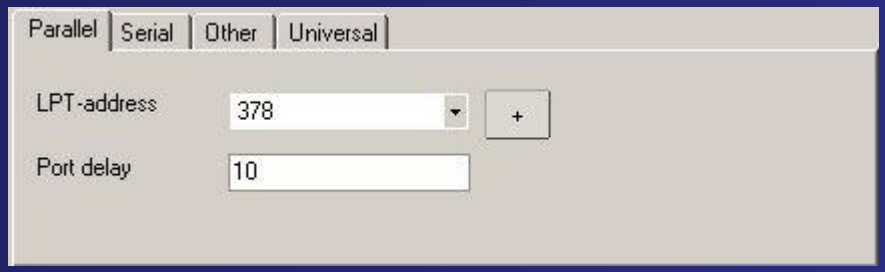
Konfiguracja programu Bascom AVR

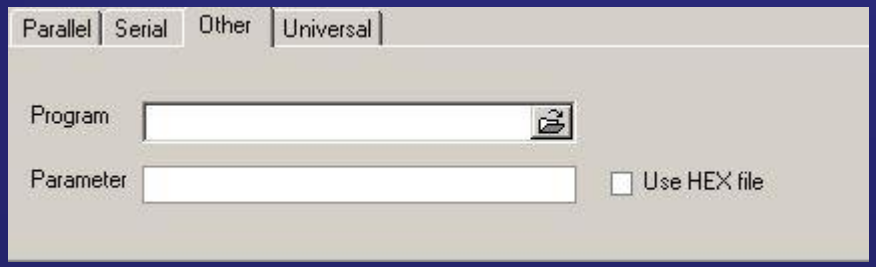


Konfiguracja programu Bascom AVR

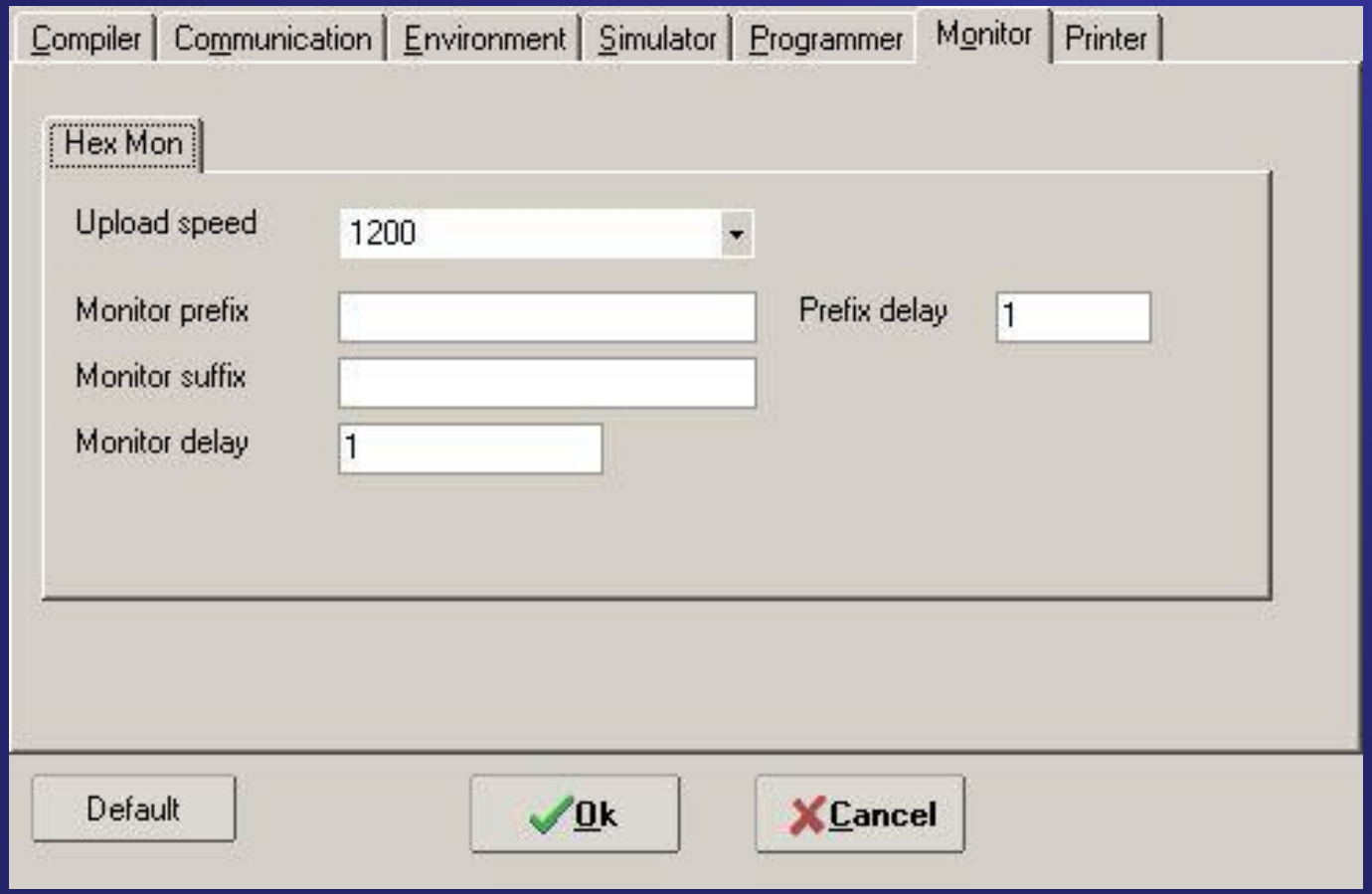


Konfiguracja programu Bascom AVR

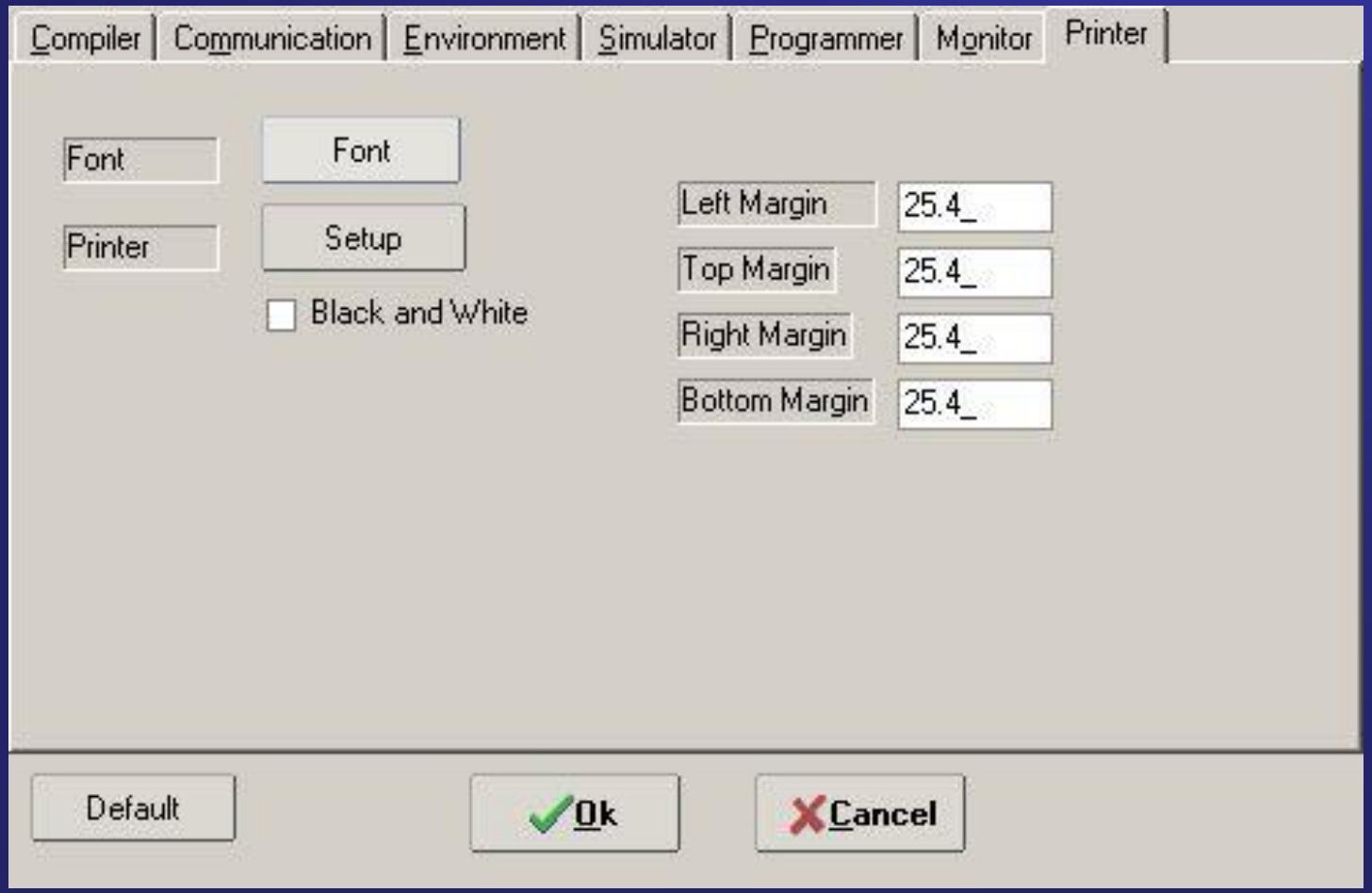




Konfiguracja programu Bascom AVR



Konfiguracja programu Bascom AVR



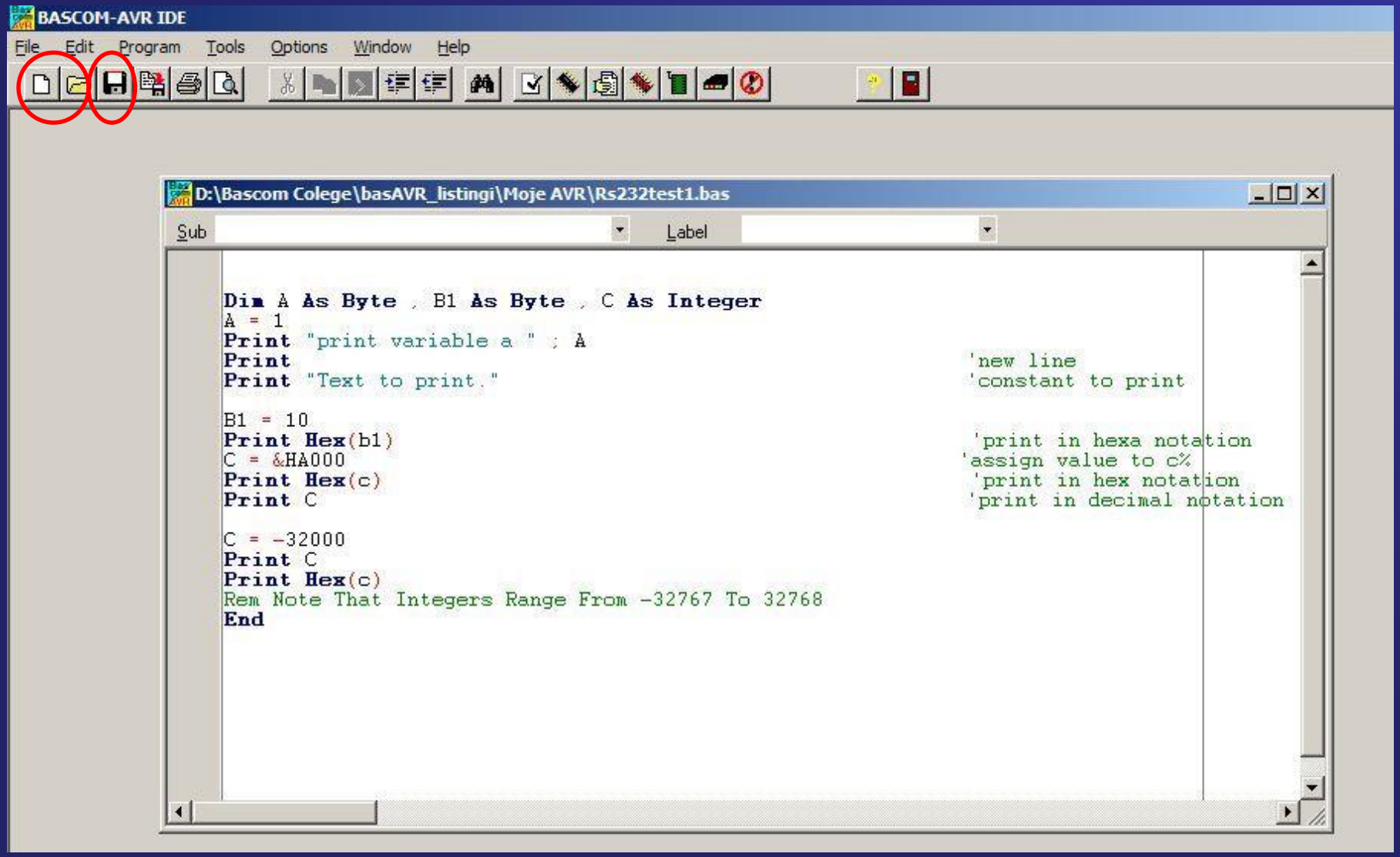
Pierwszy program Bascom AVR

Proces tworzenia programu w języku Bascom można podzielić na kilka etapów:

1. **Utworzenie nowego pliku programu lub otwarcie istniejącego**
2. **Napisanie programu lub wprowadzenie zmian do już istniejącego**
3. **Zapisanie programu**
4. **Przeprowadzenie kompilacji**
5. **Poprawienie błędów składniowych i ponowne przeprowadzenie kompilacji**
6. **Przeprowadzenie symulacji programowej lub sprzętowej**
7. **Poprawienie błędów logicznych w działaniu programu i powtórzenie kompilacji oraz symulacji**
8. **Zaprogramowanie mikrokontrolera i sprawdzenie jego działania w urządzeniu docelowym**

Pierwszy program

1. Utworzenie nowego pliku programu lub otwarcie istniejącego
2. Zapisanie programu na dysku !!!!

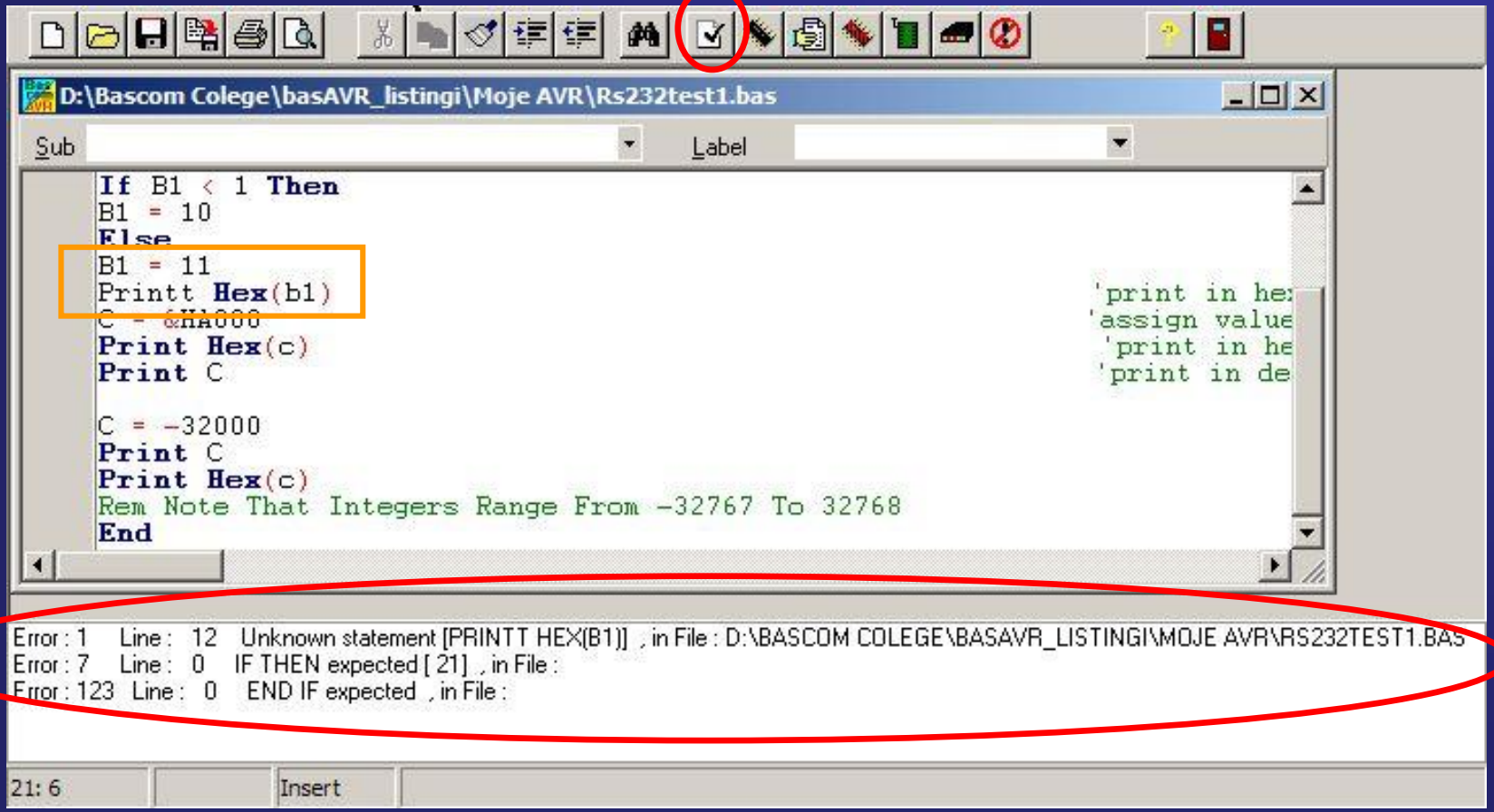


Sprawdzenie składni napisanego programu

Błędy

Przed kompilacją należy sprawdzić czy program nie zawiera błędów składniowych

Syntax check **klawisz Ctrl+F7**



Przy braku tylko jednej instrukcji End If bascom sygnalizuje dwa błędy o kodach: 7 (spodziewana instrukcja If….End) oraz 123 (brak instrukcji End If)

Poniższa tabela zawiera listę błędów mogących się pojawić podczas sprawdzania składni lub kompilacji.

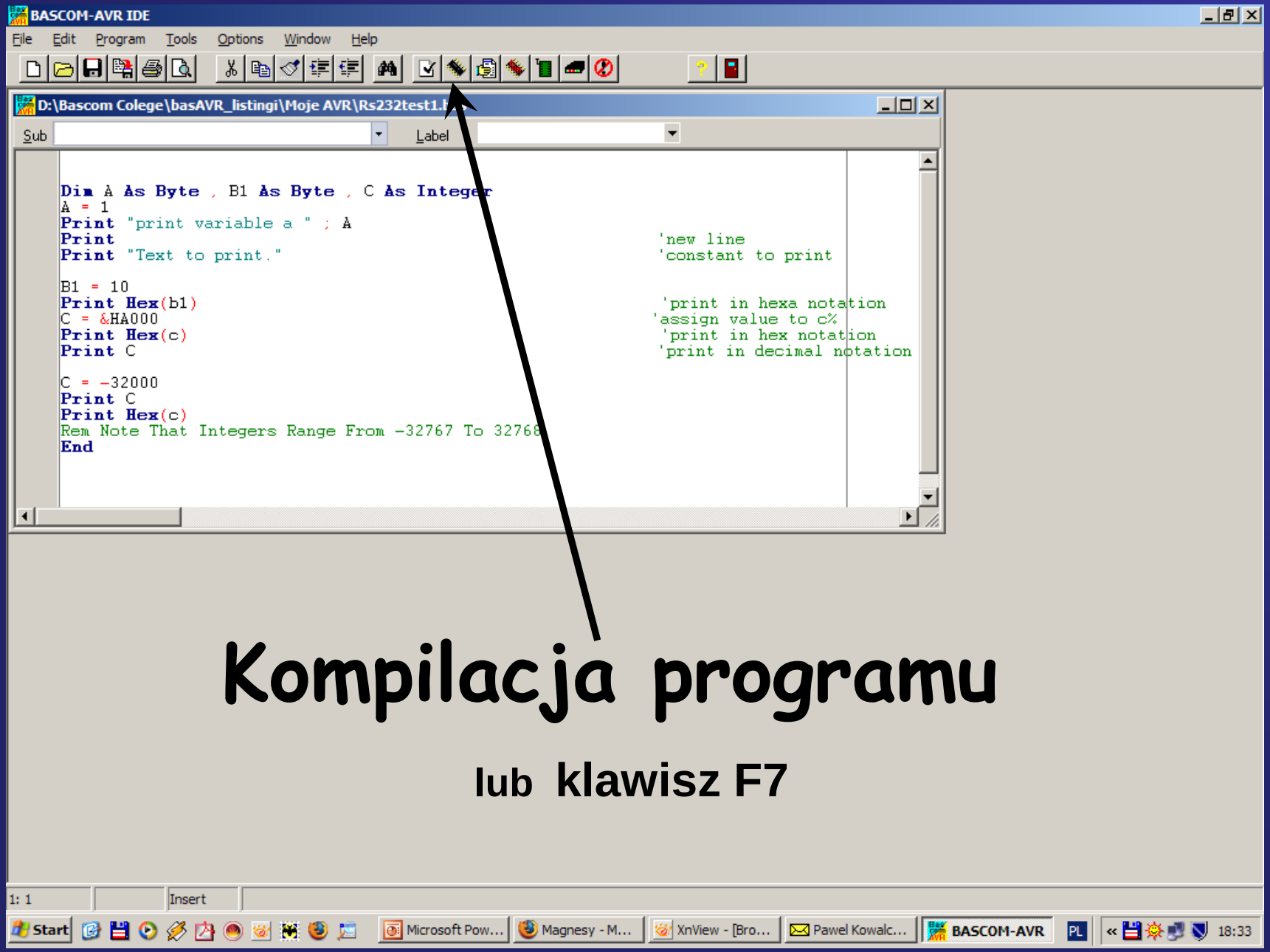
Kod błędu	Opis
1	Nieznana instrukcja
2	Nieznana struktura instrukcji EXIT
3	Spodziewano się WHILE
4	Brak miejsca w pamięci IRAM na zmienną typu Bit
5	Brak miejsca na zmienne typu Bit
6	Spodziewana . (kropka) w nazwie pliku.
7	Spodziewana instrukcja IF..THEN
8	Pliku źródłowego nie odnaleziono
9	Maksymalnie można użyć 128 instrukcji ALIAS
10	Nieznany typ wyświetlacza
11	Spodziewano się INPUT, OUTPUT, 0 lub 1
12	Nieznany parametr instrukcji CONFIG
13	Ta stała już jest zdefiniowana
14	Bajty mogą być tylko w IRAM
15	Błędny typ danych
16	Nieznana definicja
17	Spodziewano się 9 parametrów
18	Zmienne bitowe umieszczone mogą być tylko w pamięci SRAM lub IRAM
19	Spodziewano się określenia długości zmiennej typu String
20	Nieznany typ danych
21	Brak wolnej pamięci IRAM
22	Brak wolnej pamięci SRAM
23	Brak wolnej pamięci XRAM
24	Brak wolnej pamięci EEPROM
25	Ta zmienna już jest zdefiniowana
26	Spodziewano się AS
27	Spodziewano się parametru
28	Spodziewano się IF..THEN
29	Spodziewano się SELECT..CASE
30	Zmienne bitowe są zmiennymi globalnymi, nie można ich usuwać
31	Błędny typ danych
32	Niezdefiniowana zmienna
33	Zmienne globalne nie mogą być usuwane
34	Błędna ilość parametrów
35	Spodziewano się 3 parametrów
35	Spodziewano się 3 parametrów
36	Spodziewano się THEN
37	Błędny operator relacji
38	Nie można wykonać tej operacji dla zmiennych bitowych
39	Spodziewano się FOR
40	Ta zmienna nie może być parametrem instrukcji RESET
41	Ta zmienna nie może być parametrem instrukcji SET
42	Spodziewano się liczby jako parametru
43	Pliku nie odnaleziono
44	Spodziewano się 2 zmiennych
45	Spodziewano się DO
46	Błędne przypisanie
47	Spodziewano się UNTIL
50	Liczba nie mieści się w zmiennej Integer
51	Liczba nie mieści się w zmiennej Word
52	Liczba nie mieści się w zmiennej Long
60	Ta etykieta już istnieje
61	Etykiety nie znaleziono
62	Najpierw SUB lub FUNCTION
63	Parametrem funkcji ABS() może być liczba typu Integer lub Long
64	Spodziewany , (przecinek)
65	Urządzenie nie zostało otwarte
66	Urządzenie już jest otwarte
68	Spodziewano się numeru kanału
70	Ta szybkość transmisji nie może być użyta
71	Typ przekazanych parametrów nie jest zgodny z zadeklarowanym
72	Getclass error. Jest to błąd wewnętrzny.
73	Używanie PRINT w połączeniu z tą funkcją jeszcze nie działa
74	Spodziewano się 3 parametrów
80	Kod nie mieści się w pamięci tego układu
81	Użyj funkcji HEX() zamiast PRINTHEX
82	Użyj funkcji HEX() zamiast LCDHEX
85	Nieznane źródło przerwania
86	Błędny parametr w instrukcji CONFIG TIMER
87	Nazwa podana jako parametr instrukcji ALIAS już jest używana
88	Spodziewano się 0 lub 1
89	Liczba musi zawierać się w przedziale 1-4

...

242	Wystąpił brak zgodności zmiennych
243	Numer bitu wykracza poza liczbę dopuszczalną dla tej zmiennej
244	Nie możesz używać wskaźnika Y
245	Zmienne tablicowe nie mogą być w pamięci IRAM
246	Brak miejsca na definicje w pliku .DEF
247	Spodziewano się kropki
248	Powinien byc użyty argument BYVAL w tej deklaracji
249	Procedura obsługi przerwania jest już zdefiniowana
250	Spodziewano się GOSUB
251	Ta etykieta musi być nazwana SECTIC
252	Spodziewano się zmiennej Integer lub Word
253	Ta zmienna nie może być w pamięci ERAM
254	Spodziewana zmienna
255	Spodziewano się Z lub Z+
256	Spodziewano się zmiennej typu Single
257	Spodziewano się ""
258	Spodziewano się SRAM
259	Zmienne typu Byte nie mogą przyjmować wartości ujemnych
260	Ciąg znaków nie zmieści się w tej zmiennej typu String
261	Spodziewano się tablicy
262	Spodziewano się ON lub OFF
263	Indeks tablicy poza zakresem
264	Zamiast tego użyj ECHO OFF i ECHO ON
265	Spodziewano się offsetu w rozkazie LDD lub STD. Np. Z+1
266	Spodziewano się TIMER0, TIMER1 lub TIMER2
267	Spodziewano się stałej liczbowej
268	Paramter musi zawierać się w granicach 0 - 3
269	Spodziewano się END SELECT
270	Ten adres już jest zajęty
322	Ten typ danych nie jest obsługiwany przez tą instrukcję
232	Etykieta posiada zbyt dużo znaków
234	Ten układ nie jest obsługiwany przez bibliotekę I2C w trybie Slave
325	Stopień podziału preskalera musi wynosić: 1, 8, 32, 128, 256 lub 1024
326	Spodziewano się #ENDIF
327	Maksymalna wielkoś to 255
328	Nie działa z programowym układem UART
999	Wersje DEMO lub BETA generują kod tylko do 2 KB

Uwaga! Często zdarza się że kompilator raportuje błąd “File not found”, który jest zwykle spowodowany przez błędne określenie parametrów instrukcji - zwłaszcza CONFIG.

Kompilacja



Kompilacja programu

lub klawisz F7

Kompilacja

Podczas kompilacji, w zależności od ustawień w oknie konfiguracyjnym kompilatora, utworzone zostaną następujące pliki:

xxx.BIN - Plik binarny, który można wpisać do mikroprocesora.

xxx.DBG - plik wykorzystywany podczas symulacji.

xxx.OBJ - Plik wykorzystywany przez symulator AVR Studio ale także przez wewnętrzny symulator Bascoma.

xxx.HEX - plik hexadecimal wykorzystywany przez niektóre programy obsługujące programatory, plik ten może być także wpisany do pamięci programu mikrokontrolera przez botloader.

xxx.ERR - plik z komunikatem o błędach tworzonych po znalezieniu błędów składniowych.

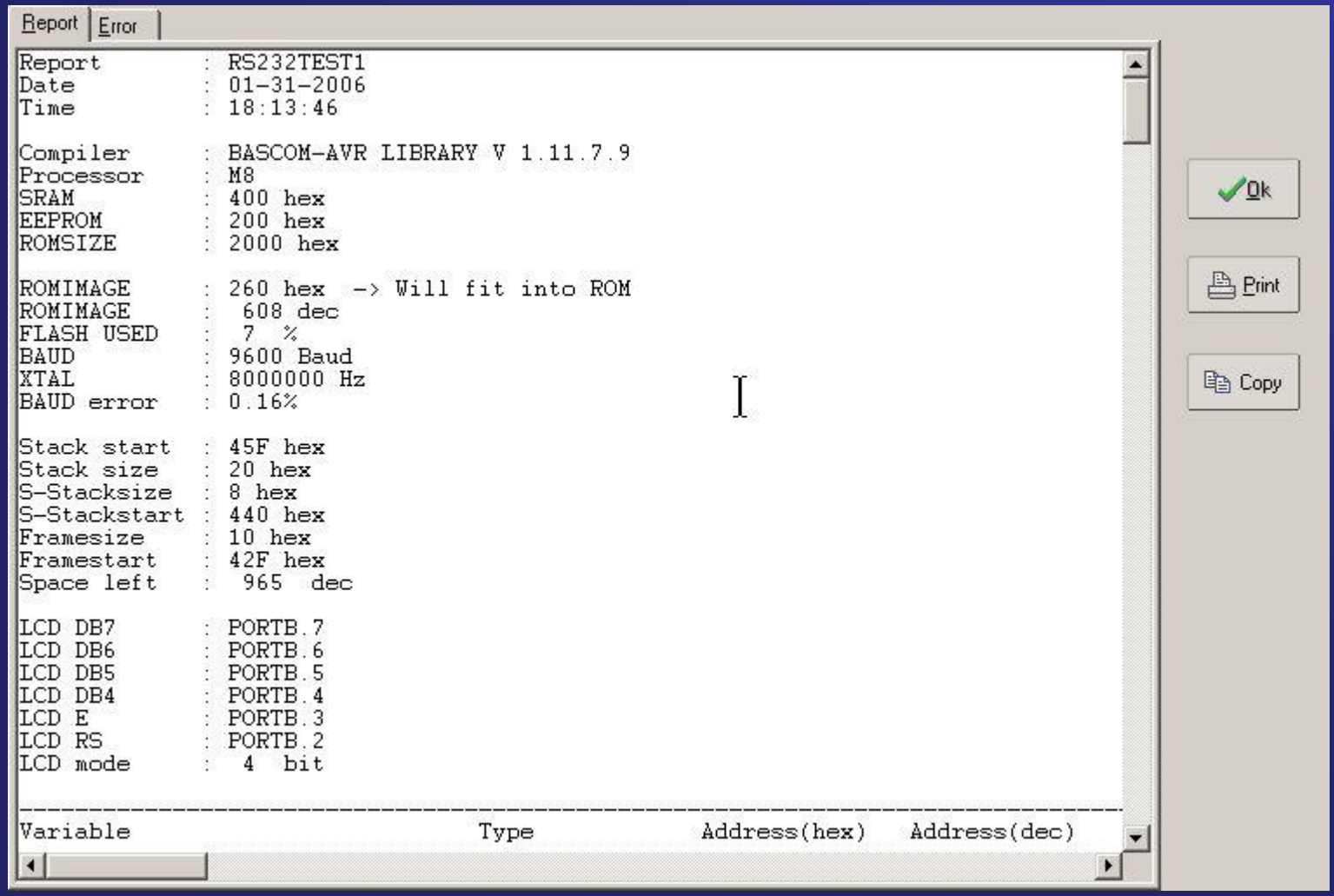
xxx.RPT - plik raportu z kompilacji.

xxx.EEP - plik obrazu pamięci EEPROM.

Uwaga: pliki te powstają w tym samym katalogu gdzie został zapisany program

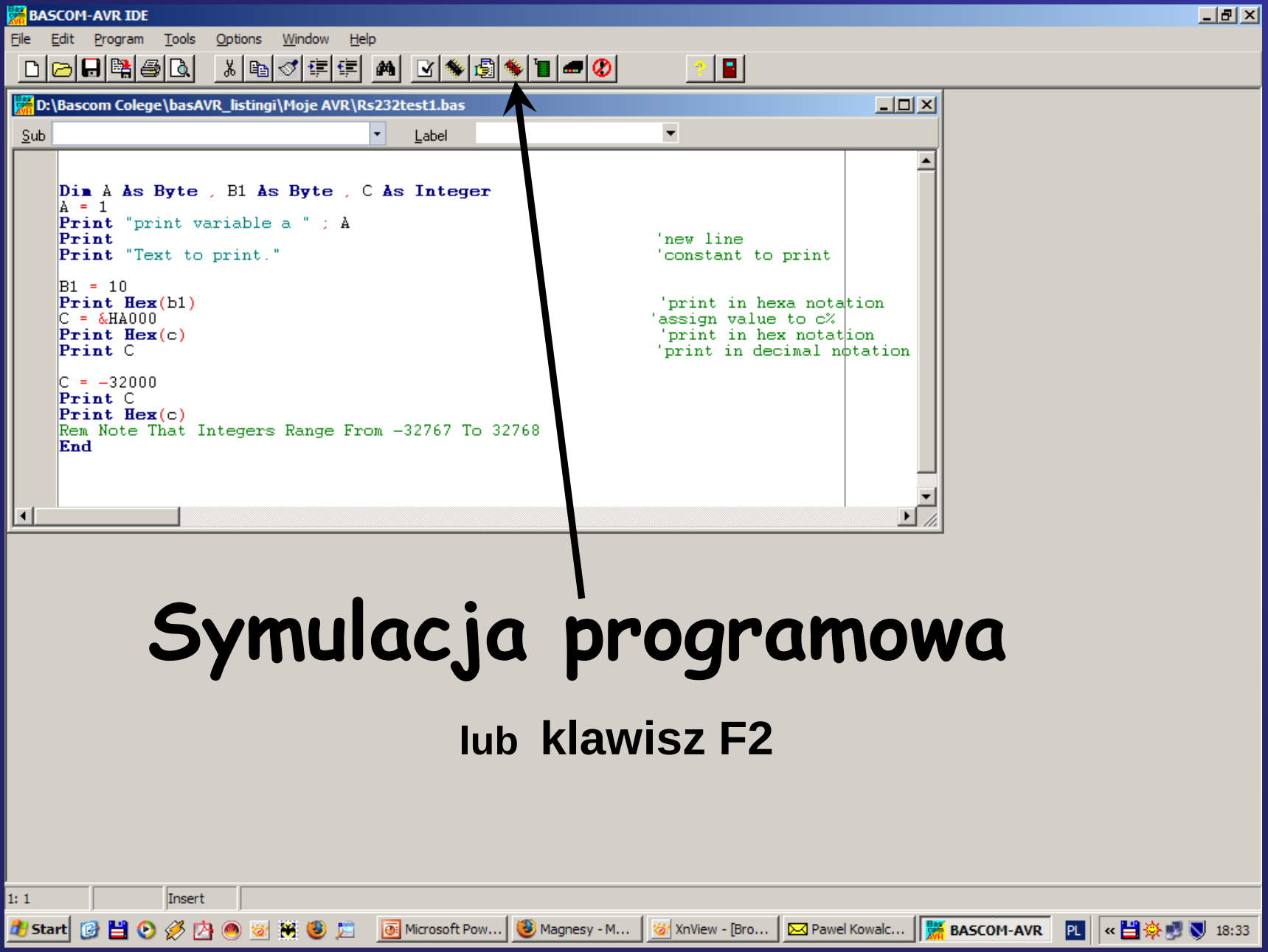
Informacja o kompilacji

Wybierając z menu *Program* polecenie *Show result* lub klawiszem Ctrl + W możemy obejrzeć informacje dotyczące przebiegu kompilacji:



Symulacja programowa

Symulacja programowa

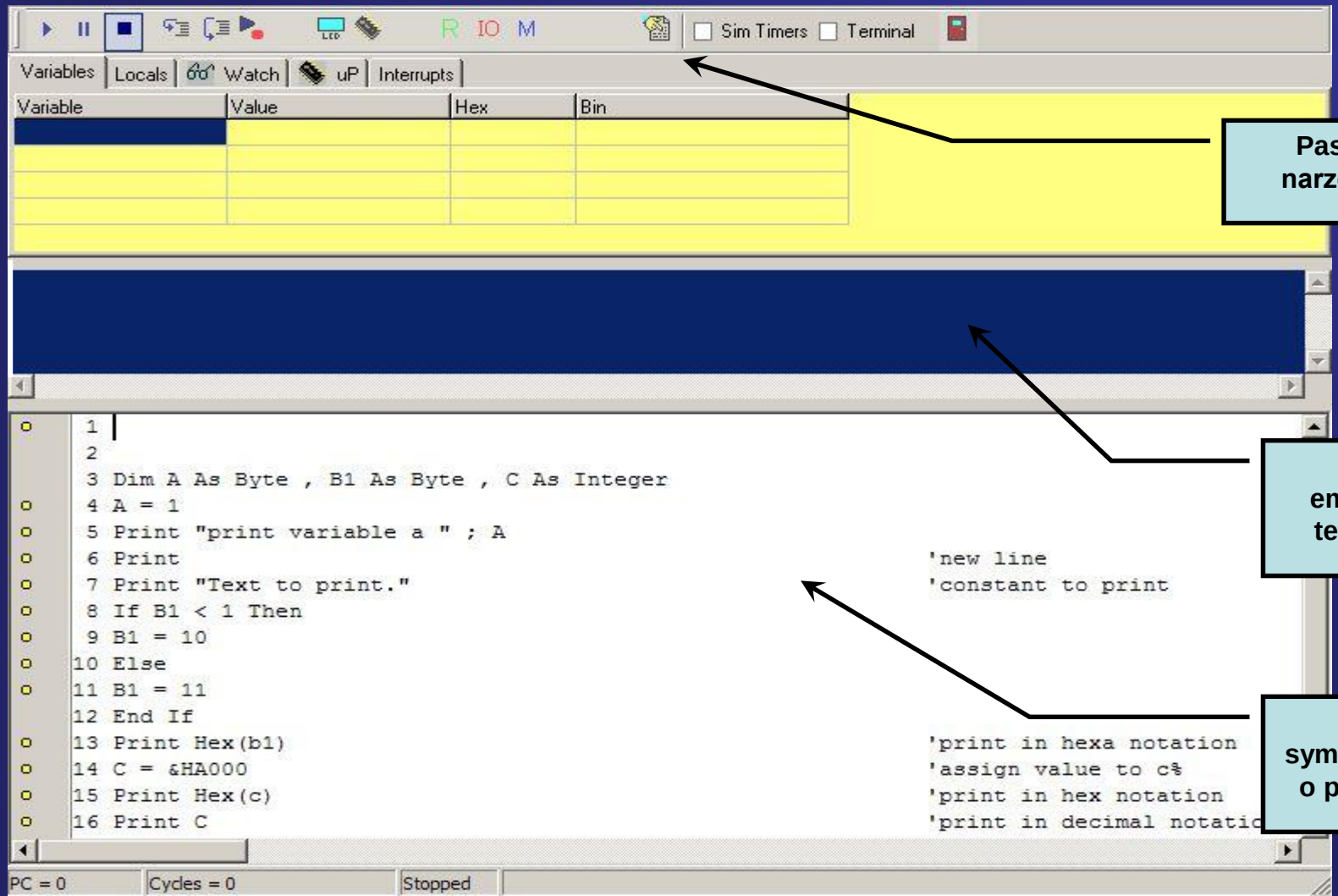


Symulacja programowa

lub klawisz F2

Symulacja programowa

Wybierając z menu *Program* polecenie *Simulate* lub klawiszem F2 możemy przeprowadzić symulacje działania napisanego programu bez konieczności programowania mikrokontrolera:



Symulacja programowa

Opisz ikon paska narzędzi

Start symulacji programu

Pauza symulacji programu

Symulacja trwająca do wybranej kursorem linii programu

Włącza okno programowego emulatora

Okno z wartościami komórek pamięci SRAM EEPROM

Wyjście z programu

Stop symulacji programu

Praca krokowa symulacji

Praca krokowa symulacji z ominięciem linii podprogramów

Włą/wył sprzętową symulację

Okno z wartościami rejestrów R0..R311

Okno z wartościami rejestrów I/O mikrokontrolera

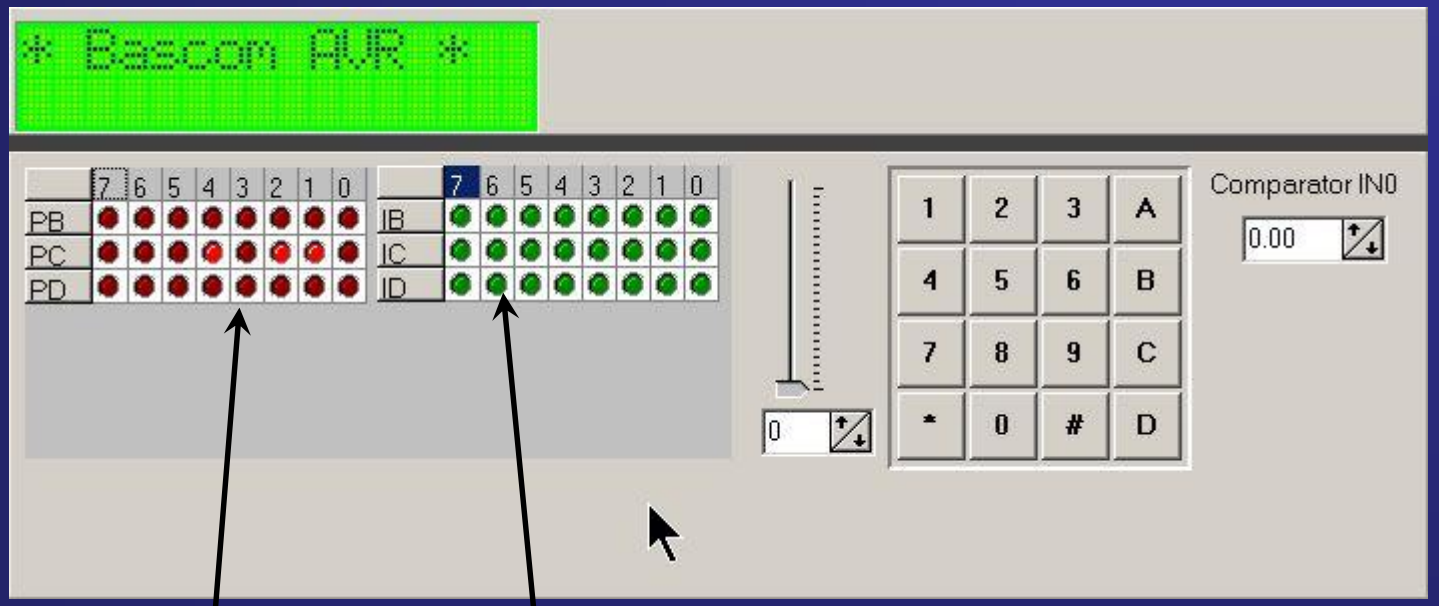
Włą/wył odświeżanie wartości zmiennych podczas symulacji

Włącza sprawdzanie czy symulowane są Timery

Włącza emulator terminala podczas symulacji

Symulacja programowa

Okno symulacji programowej z wirtualnymi peryferiami mikrokontrolera

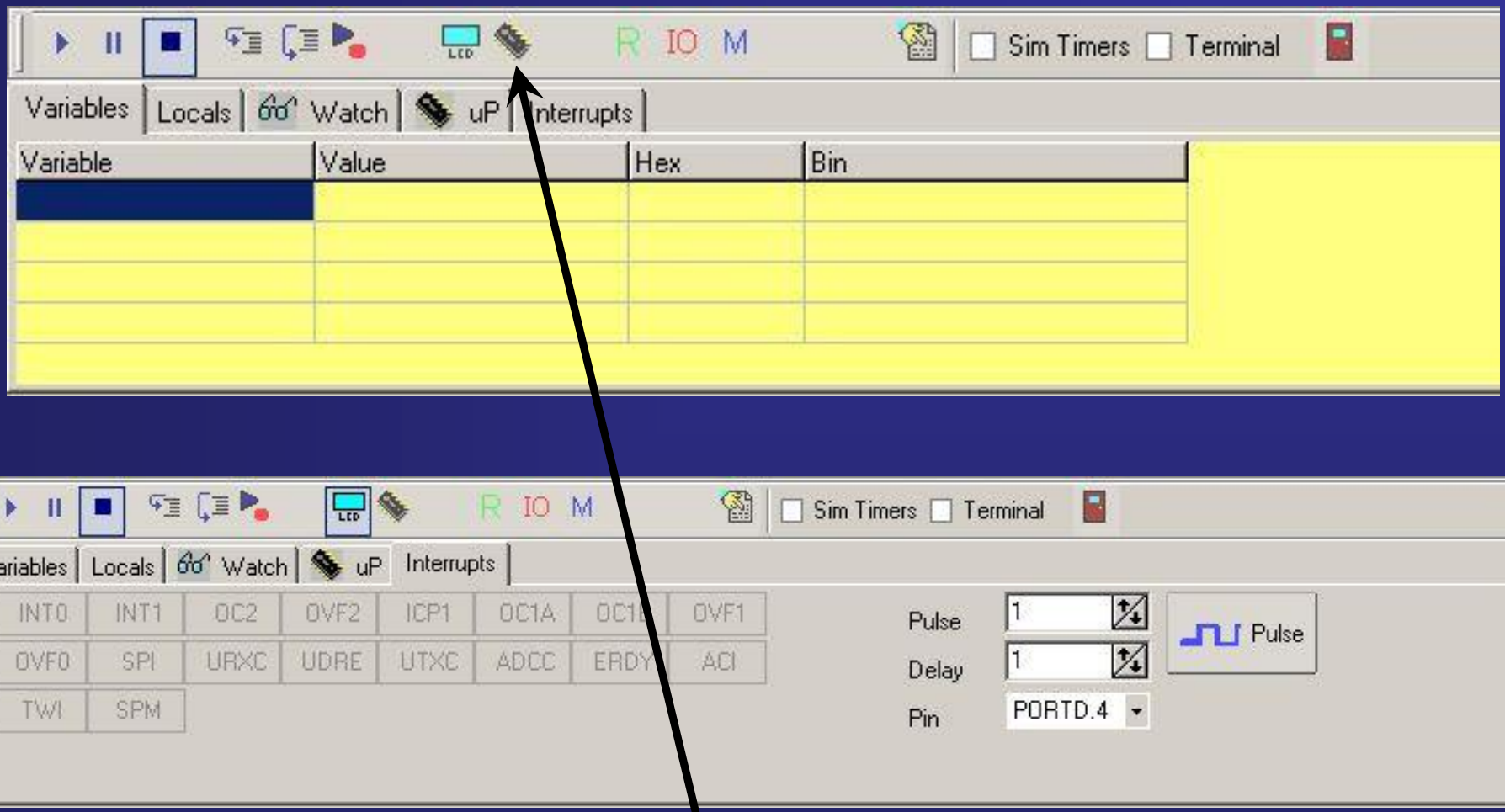


Czerwone diody dotyczą stanów linii portów wyjściowych

Zielone diody dotyczą stanów linii portów wejściowych

Symulacja sprzętowa

Okno symulacji programowej z wirtualnymi peryferiami mikrokontrolera



Symulacja sprzętowa

Symulacja sprzętowa

Do pamięci mikrokontrolera powinien być wpisany program będący rodzajem monitora, który komunikuje się z symulatorem Bascoma poprzez interfejs RS232

C:\Program Files\MCS Electronics\BASCOM-AVR\SAMPLES\basmon.bas

Sub | Label

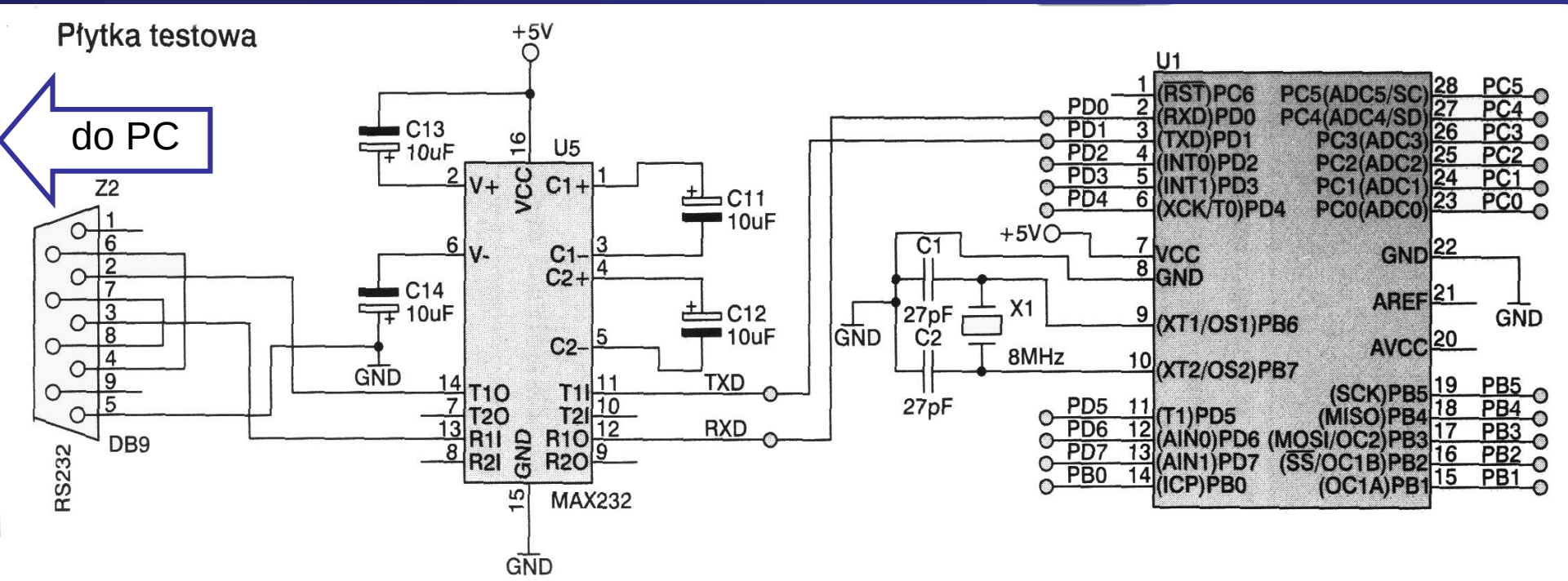
```
'----------------------------- MCS Electronics------------------------------
'                                 BASMON.BAS
'regfile must match the chip you use in your target system
$regfile = "m8def.dat"
'crystal must match too
$crystal = 8000000
'baudrate must match the baudrate of the terminal emulator
$baud = 19200

'[variables]
Dim Krk As Byte                                             'command
Dim Address As Word                                         'address
Dim Adrl As Byte , Adrh As Byte                              'for new OUT command
Dim Vl As Byte                                              'value
'[main program]
Print "BASMON Version 1.01"

Do
  Krk = Waitkey()
  If Krk = "T" Then                                         'check if it is attached
     Print Chr(13);                                         'it is working
  Elseif Krk = "W" Then
     Address = Waitkey()                                    'wait for address
     Vl = Waitkey()                                         'wait for value
     Out Address , Vl                                      'write value
     Print Chr(13);                                         'confirm
  Elseif Krk = "R" Then
     Address = Waitkey()                                    'wait for address
     Vl = Inp(address)                                     'get value
     Print Chr(vl);                                         'write back value
  Elseif Krk = "O" Then
     Adrl = Waitkey()                                       'wait for LSB of address
     Adrh = Waitkey()                                       'wait for MSB
     Vl = Waitkey()                                         'wait for data
     Address = Adrh * 256
     Address = Address + Adrl
     Out Address , Vl
     Print Chr(13);
  Elseif Krk = "?" Then                                     'just echo for test
     Print "?";
  End If
Loop                                                        'for ever
```

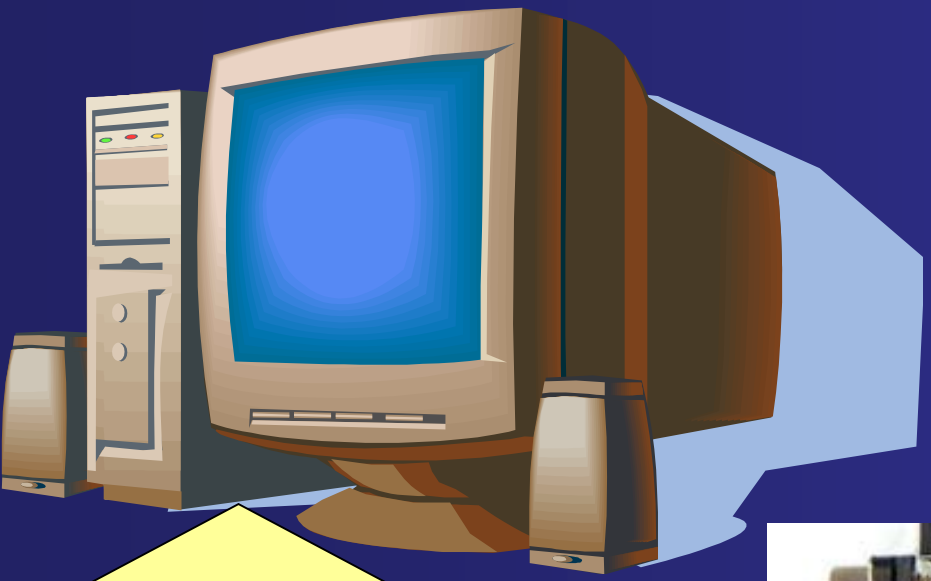
Symulacja sprzętowa

Schemat dołączenia mikrokontrolera przez konwerter napięć dla standardu RS232 do komputera PC

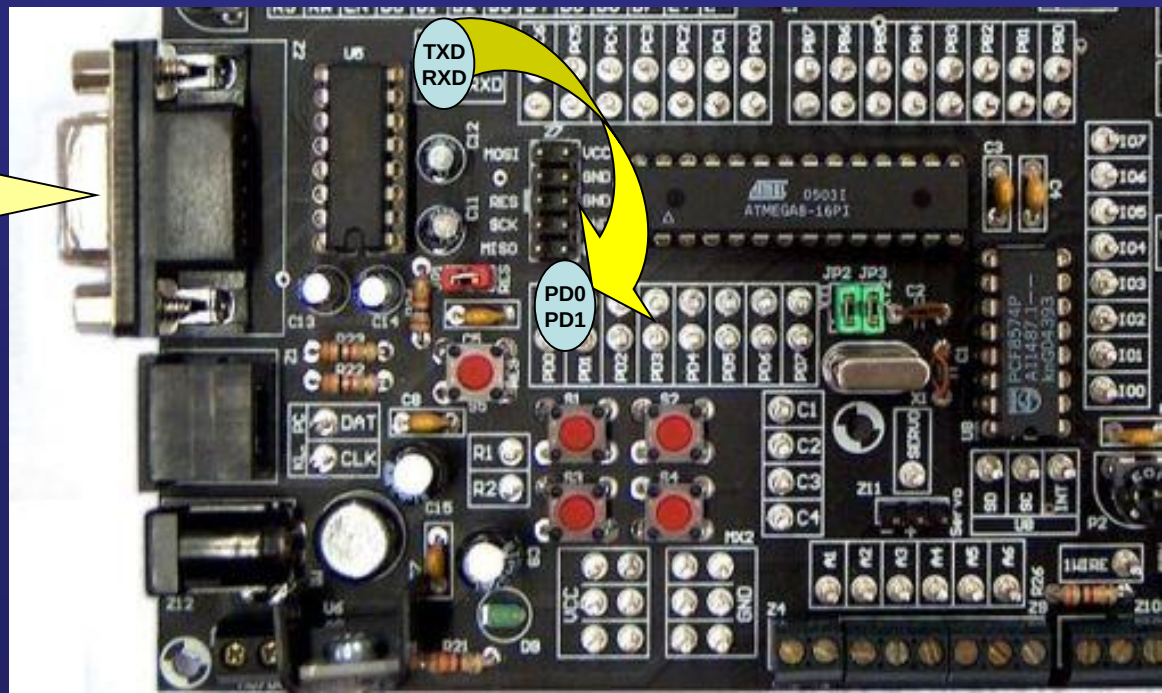


Symulacja sprzętowa

mikrokontrolera



RS232

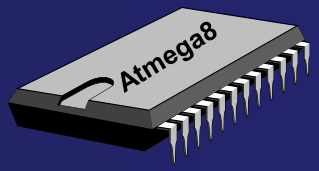


Programowanie

mikrokontrolera

Programowanie

mikrokontrolera



Wpisanie programu
do pamięci Flash ROM

```
$regfile = "m8def.dat"

$crystal = 8000000

$baud = 19200

'[variables]
Dim Rzk As Byte
Dim Adr As Word
Dim Adr1 As Byte , Adrh As Byte
Dim Y1 As Byte

'[main program]
Print "BASCOM Version 1.01"

Do
  Rzk = Inkey()
  If Rzk = "T" Then
     Print Chr(13);
  Elseif Rzk = "O" Then
     Adr = Waitkey()
     Y1 = Waitkey()
     Out Adr , Y1
     Print Chr(13);
  Elseif Rzk = "I" Then
     Adr = Waitkey()
     Y1 = Inp(adr)
     Print Chr(y1);
  Elseif Rzk = "P" Then
     Adr1 = Waitkey()
     Adrh = Waitkey()
     Y1 = Waitkey()
     Adr = Adrh * 256
     Adr = Adr + Adr1
     Out Adr , Y1
     Print Chr(13);
  Elseif Rzk = "?" Then
     Print "!";
  End If
Loop
```

Programator równoległy Uprog

Programowanie

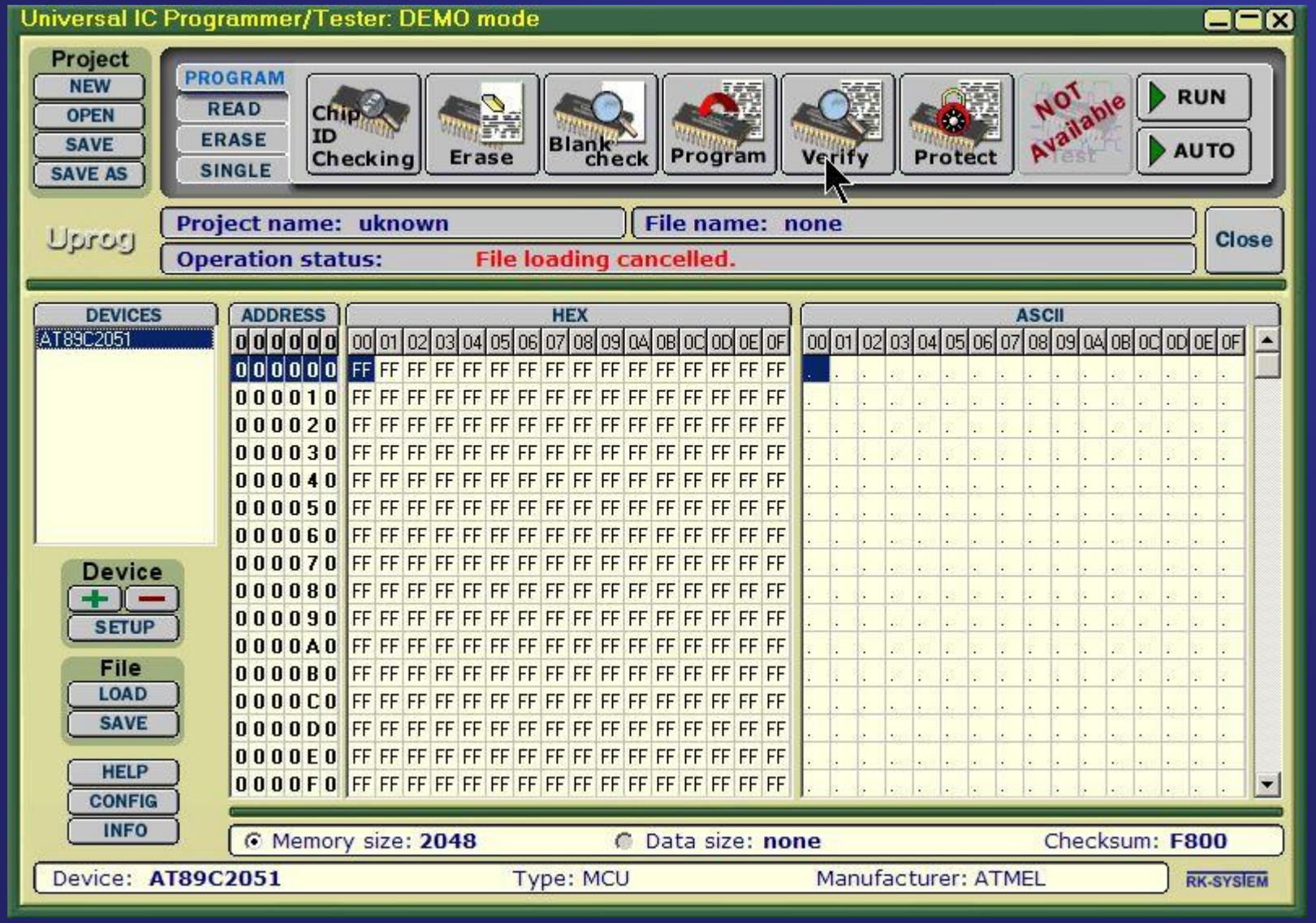
mikrokontrolera

Uwaga:

Ładunki elektrostatyczne!!

Programowanie

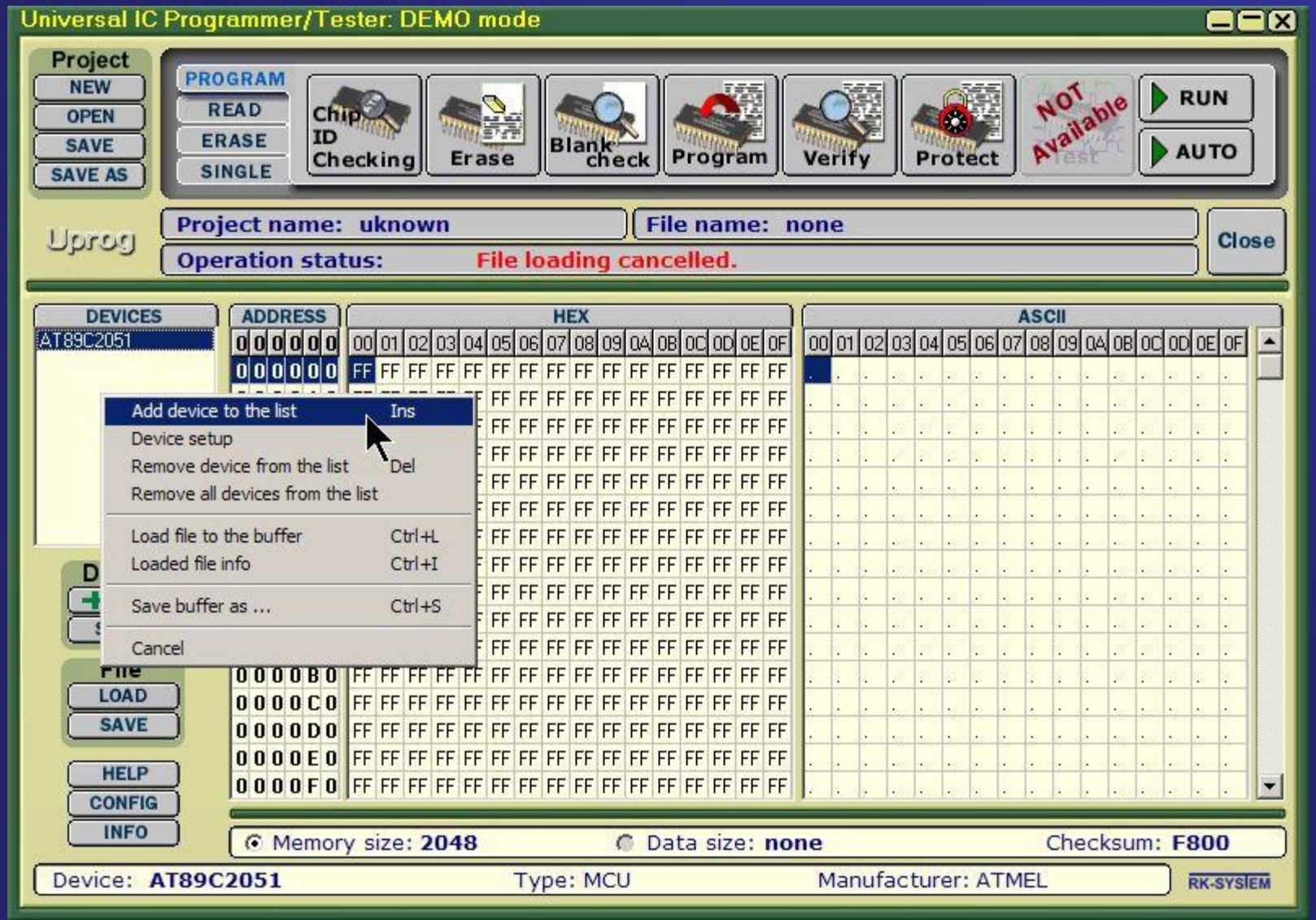
Mikrokontrolera



Programowanie

Mikrokontrolera

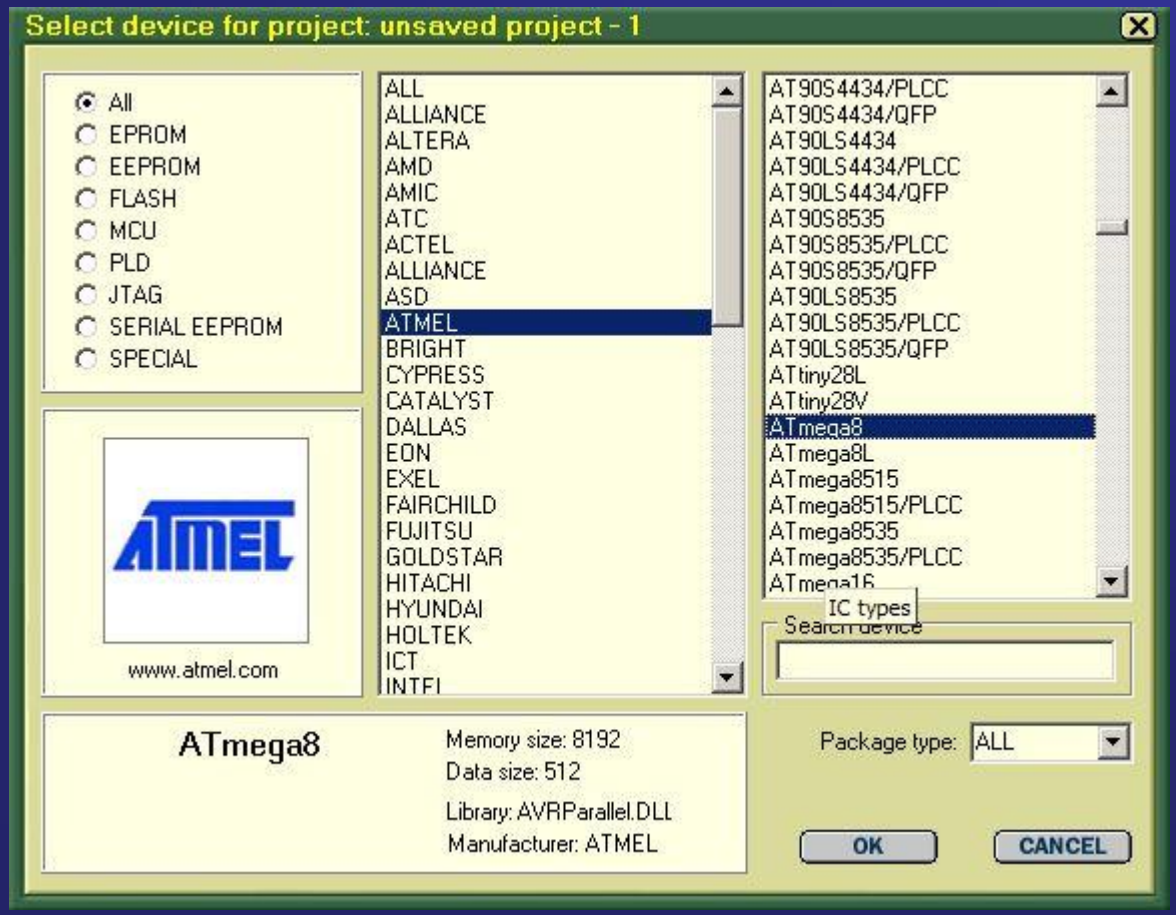
Wybranie konkretnego typu mikrokontrolera



Programowanie

Mikrokontrolera

Wybranie mikrokontrolera ATmega 8 firmy ATMEL



Programowanie

Odczytanie zawartości pamięci mikroprocesora oraz konfiguracji bitów *Fuse*

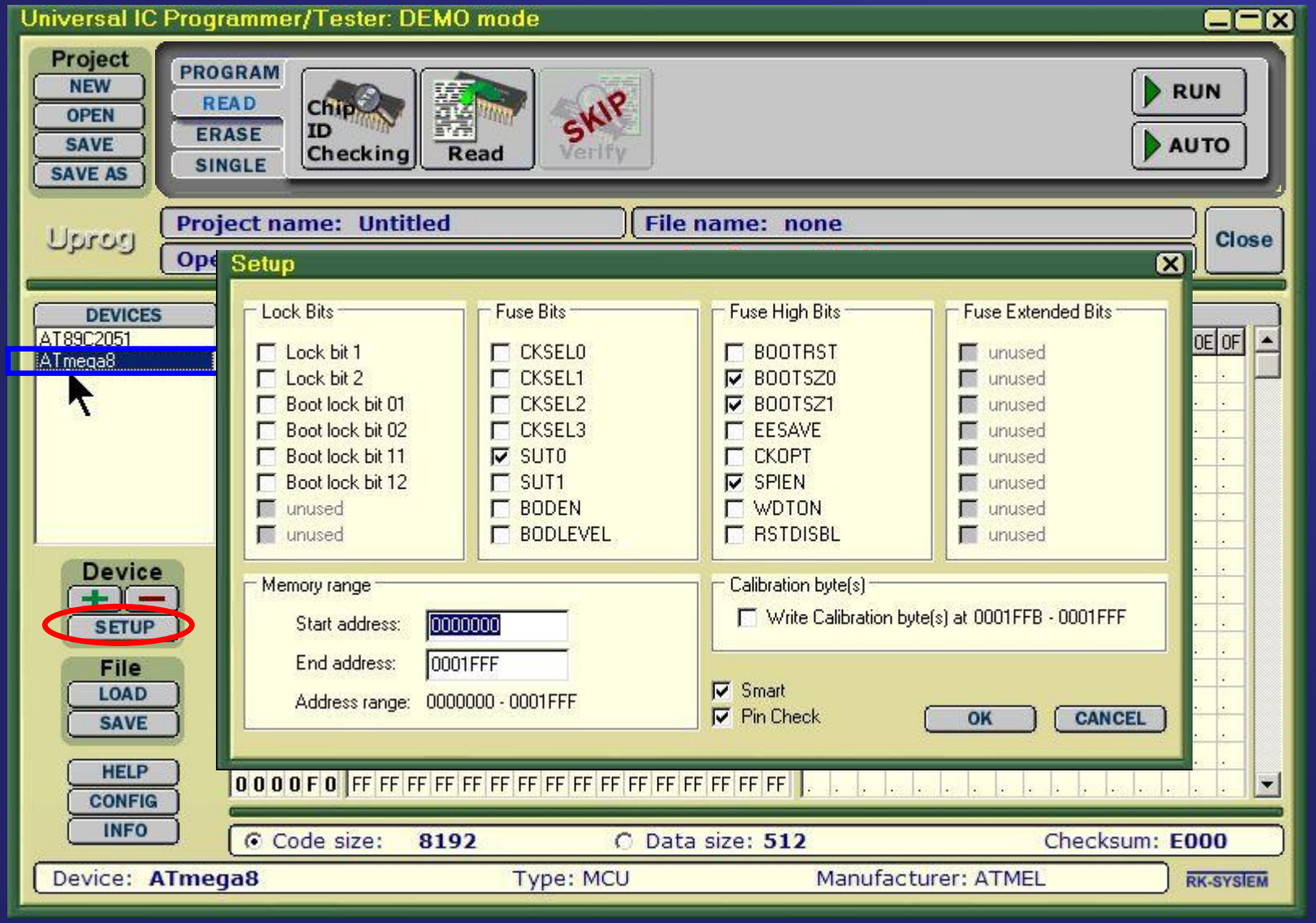
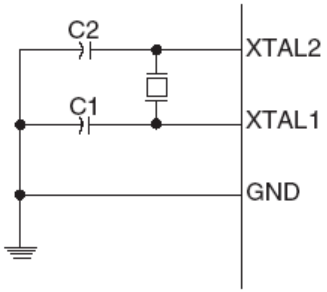


Figure 11. Crystal Oscillator Connections



My będziemy używać oscylatora 8MHz

Table 4. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range(MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101(1)	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

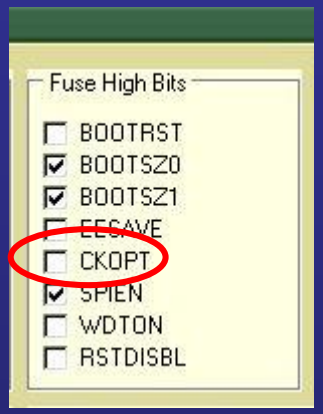
Note: 1. This option should not be used with crystals, only with ceramic resonators.

The CKSEL0 Fuse together with the SUT1..0 Fuses select the start-up times as shown in Table 5.

The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in Table 4.

Nr bitu	Nazwa **High Fuse Bitu**	Wartość **Fuse Bitu**
0	BOOTRST	**1**
1	BOOTSZ0	**0**
2	BOOTSZ1	**0**
3	EESAVE	**1**
4	CKOPT	**1**
5	SPINE	**0**
6	WDTON	**1**
7	RSTDISBL	**1**

Nr bitu	Nazwa **Fuse Bitu**	Wartość **Fuse Bitu**
0	CKSEL 0	**1**
1	CKSEL 1	**1**
2	CKSEL 2	**1**
3	CKSEL 3	**1**
4	SUT 0	**0**
5	SUT 1	**1**
6	BODEN	**1**
7	BODLEVEL	**1**



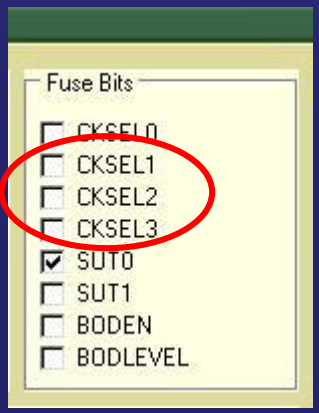


Table 5. Start-up Times for the Crystal Oscillator Clock Selection

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
0	00	258 CK(1)	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK(1)	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK(2)	–	Ceramic resonator, BOD enabled
0	11	1K CK(2)	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK(2)	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	–	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

Notes: 1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

Nr bitu	Nazwa **Fuse Bitu**	Wartość **Fuse Bitu**
0	CKSEL 0	**1**
1	CKSEL 1	**1**
2	CKSEL 2	**1**
3	CKSEL 3	**1**
4	SUT 0	**0**
5	SUT 1	**1**
6	BODEN	**1**
7	BODLEVEL	**1**

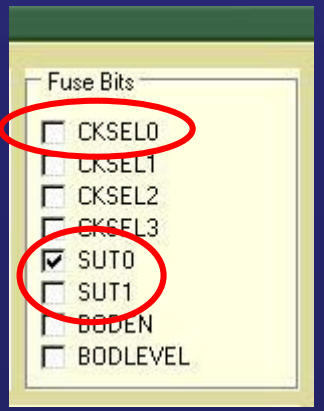


Table 9. Internal Calibrated RC Oscillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001(1)	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

Nr bitu	Nazwa **Fuse Bitu**	Wartość **Fuse Bitu**
0	CKSEL 0	**0**
1	CKSEL 1	**0**
2	CKSEL 2	**1**
3	CKSEL 3	**0**
4	SUTO 0	**0**
5	SUTO 1	**1**
6	BODEN	**1**
7	BODLEVEL	**1**

Oscillator RC inside

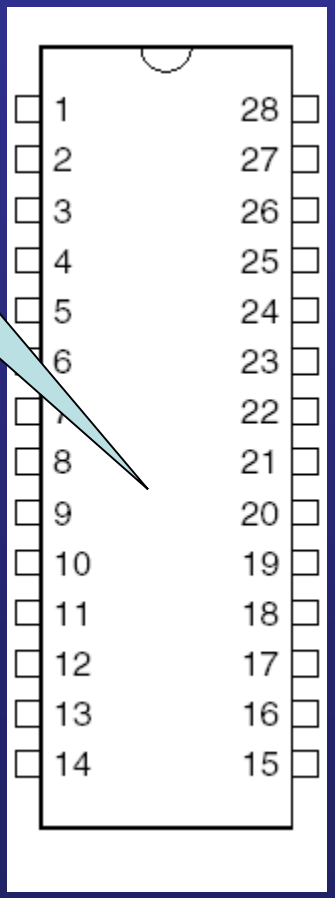


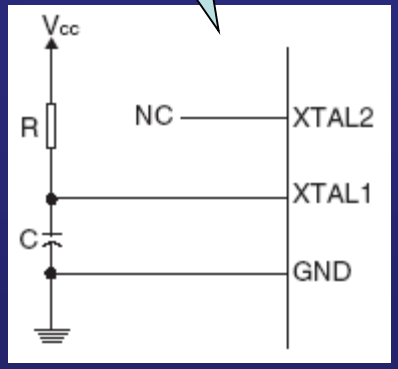
Table 7. External RC Oscillator Operating Modes

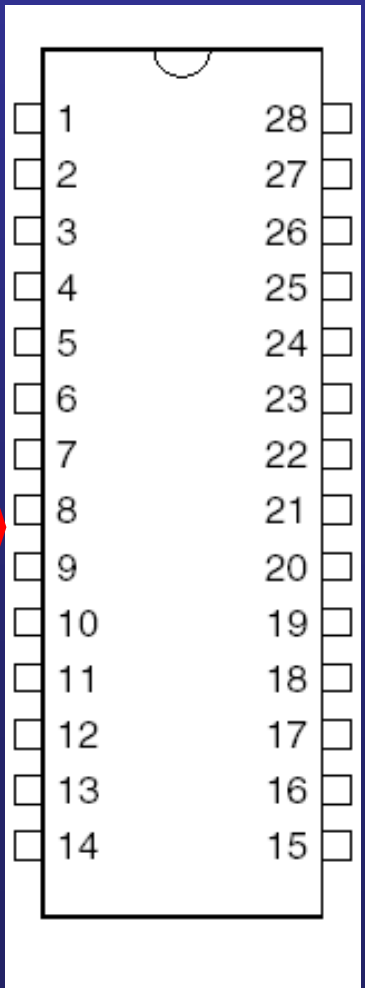
CKSEL3..0	Frequency Range (MHz)
0101	0.1 - 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 8.

Nr bitu	Nazwa **Fuse Bitu**	Wartość **Fuse Bitu**
0	CKSEL 0	**0**
1	CKSEL 1	**0**
2	CKSEL 2	**0**
3	CKSEL 3	**1**
4	SUTO 0	**0**
5	SUTO 1	**1**
6	BODEN	**1**
7	BODLEVEL	**1**

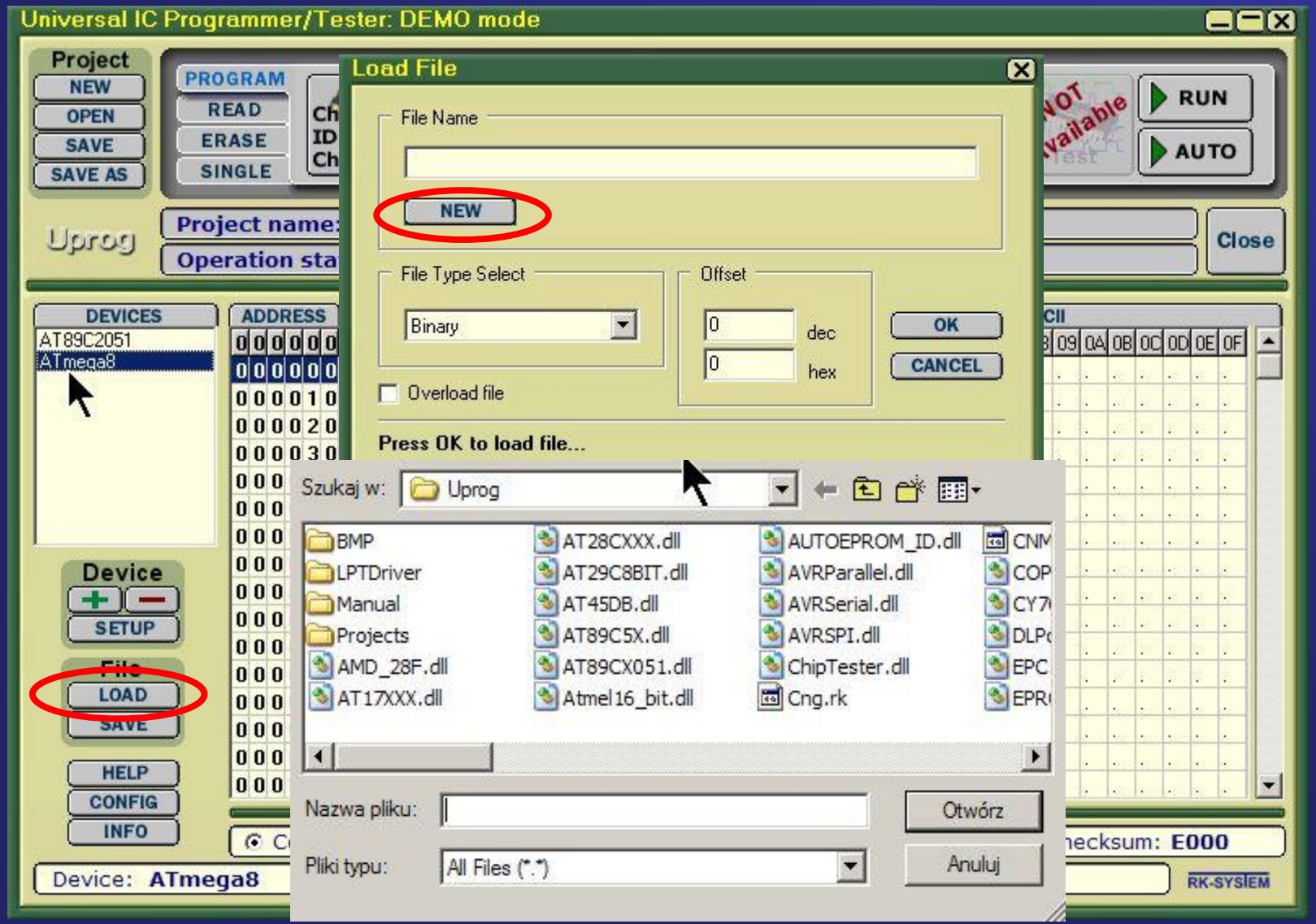
Oscillator RC outside





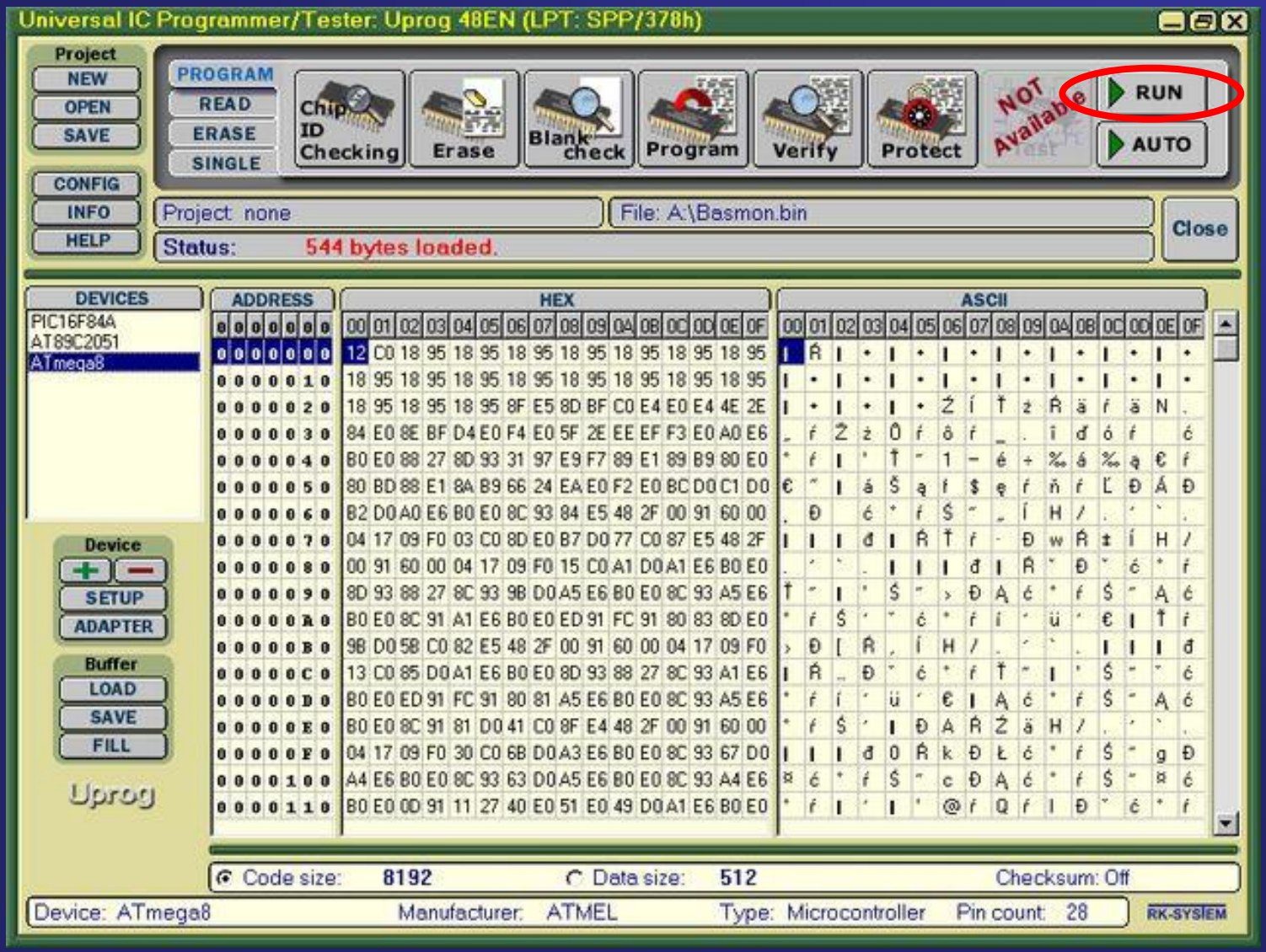
Programowanie

Wczytywanie napisanego programu w postaci binarnej lub hexadecymalnej



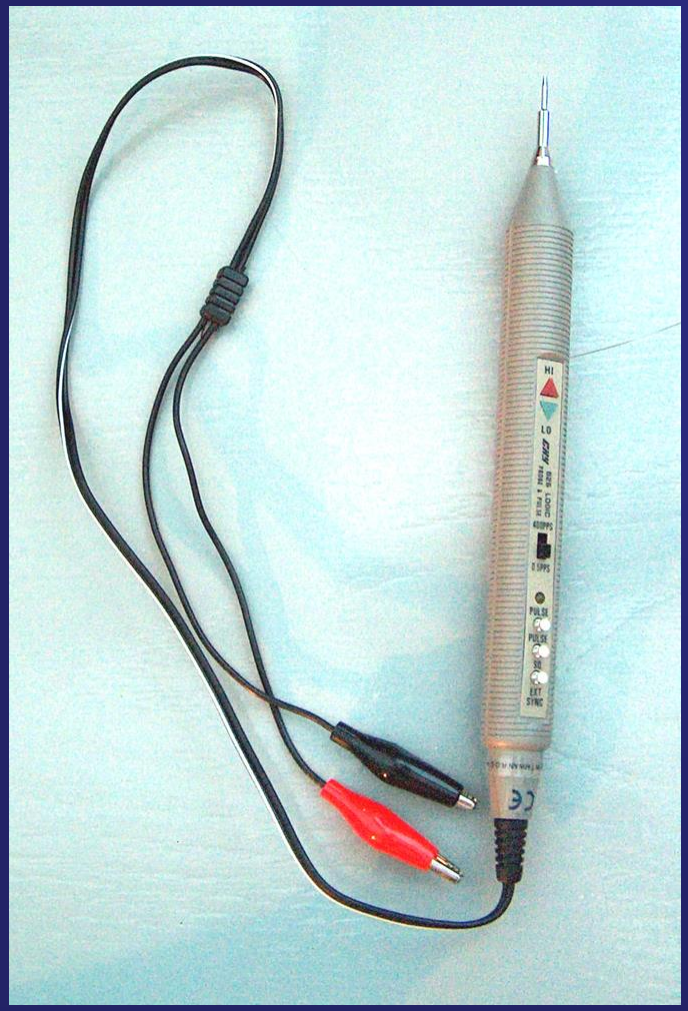
Programowanie

Wczytywanie napisanego programu w postaci binarnej lub hexadecymalnej



Przyrządy używane na pracowni

Przyrządy używane na pracowni



Sonda Logiczna

„0” – zero logiczne
„1” – jedynka logiczna

Przyrządy używane na pracowni

Miernik cyfrowy

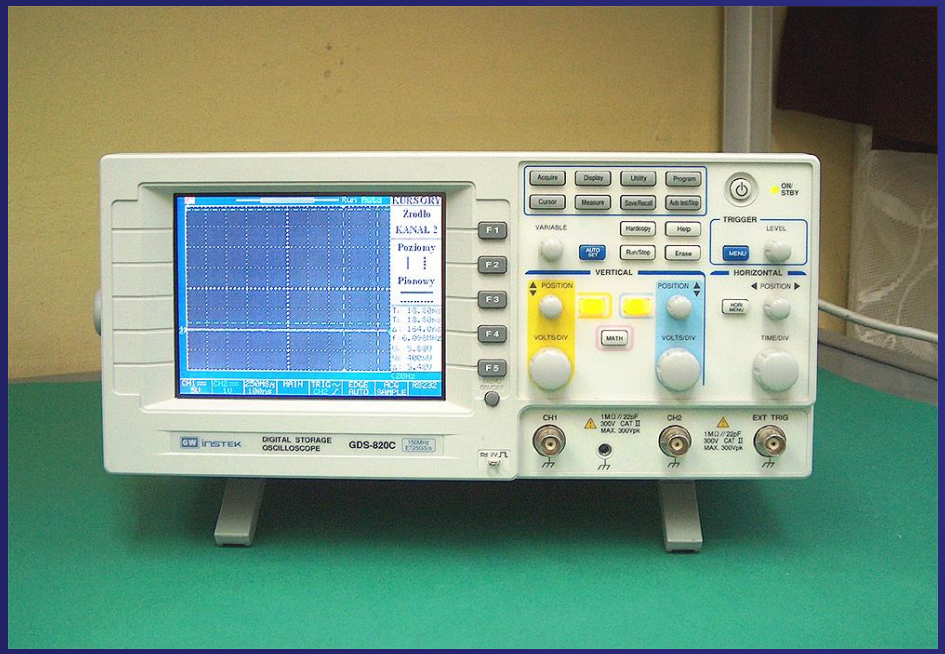
„0” – zero logiczne (0-0.8V)
„1” – jedynka logiczna (2-5V)

Przyrządy używane na pracowni

Zasilacz stabilizowany

„0V – 30V” – napięcie
„0A – 1A” – ograniczenie prądowe

Przyrządy używane na pracowni

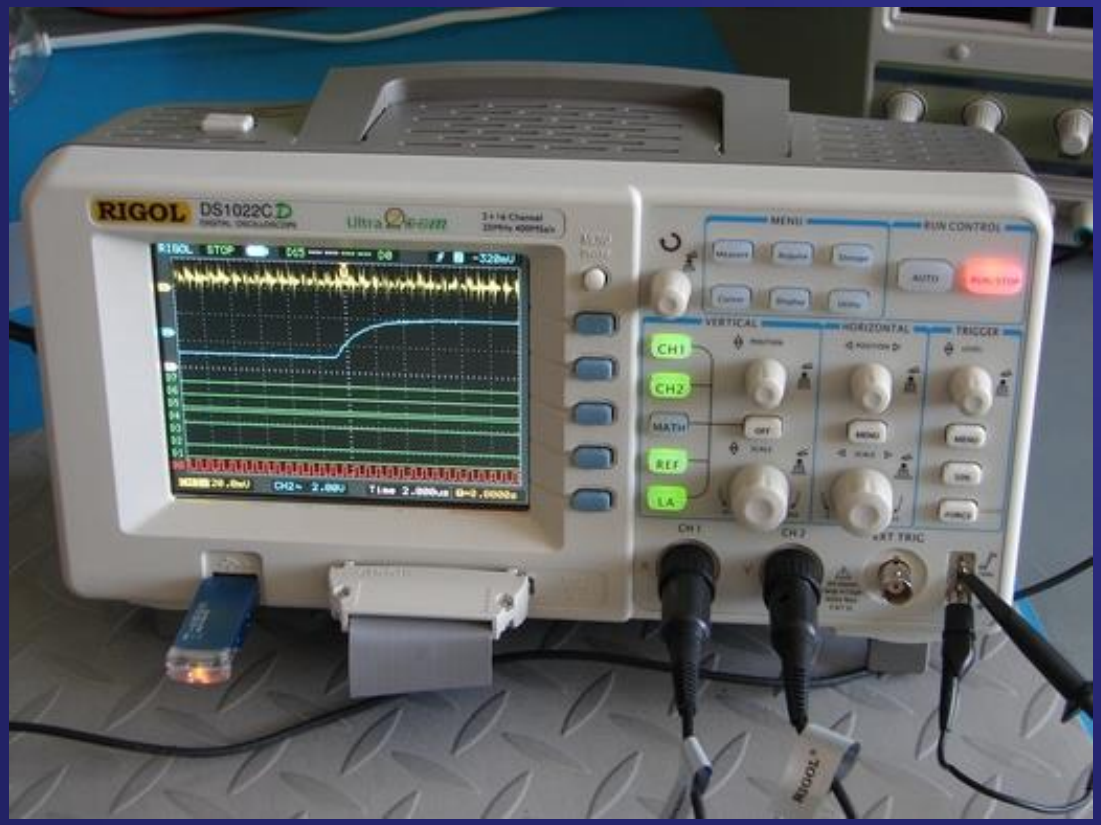


Oscyloskop cyfrowy

Możemy rejestrować przebiegi napięć zmienne w czasie oraz zapisywać je w pamięci wewnętrznej oscyloskopu

Przyrządy używane na pracowni

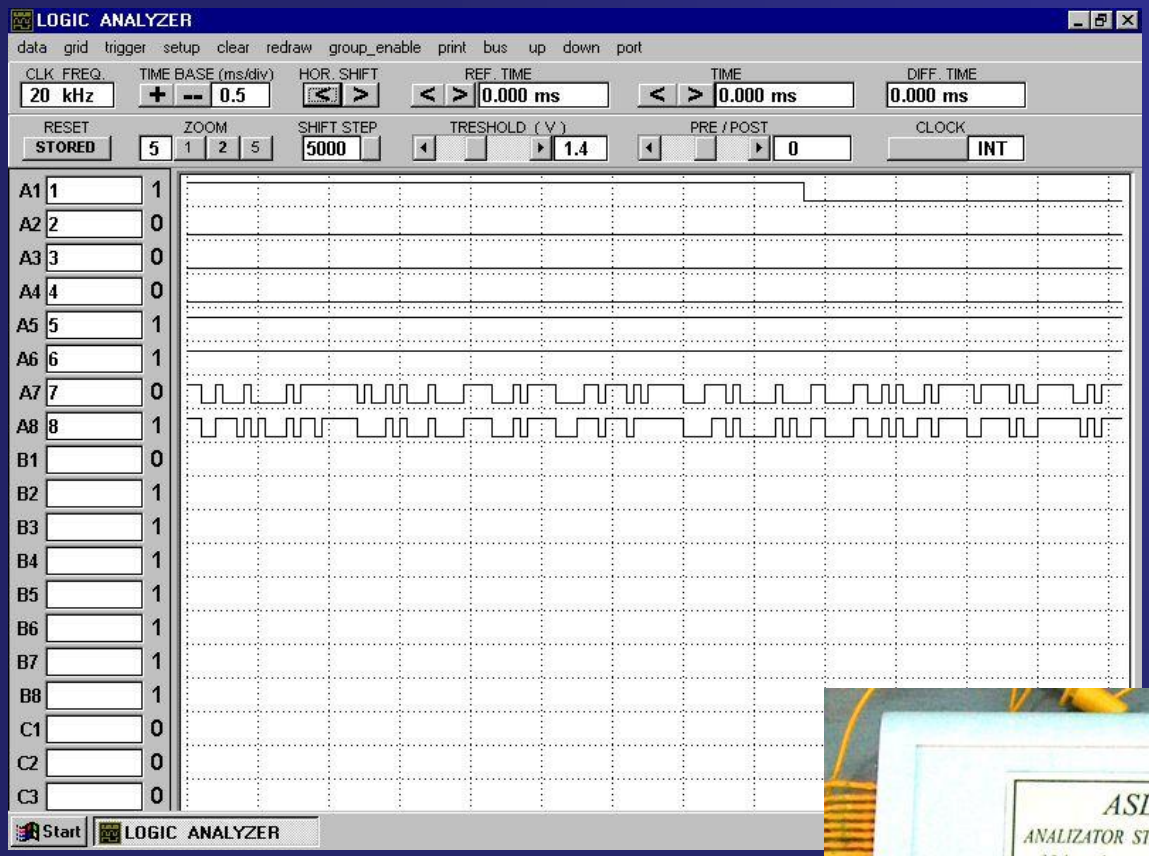
Analizator Stanów Logicznych



Możemy rejestrowane w czasie stany logiczne („0”, „1”) oraz zapamiętywać je w pamięci wewnętrznej analizatora

Przyrządy używane na pracowni

Analizator Stanów Logicznych

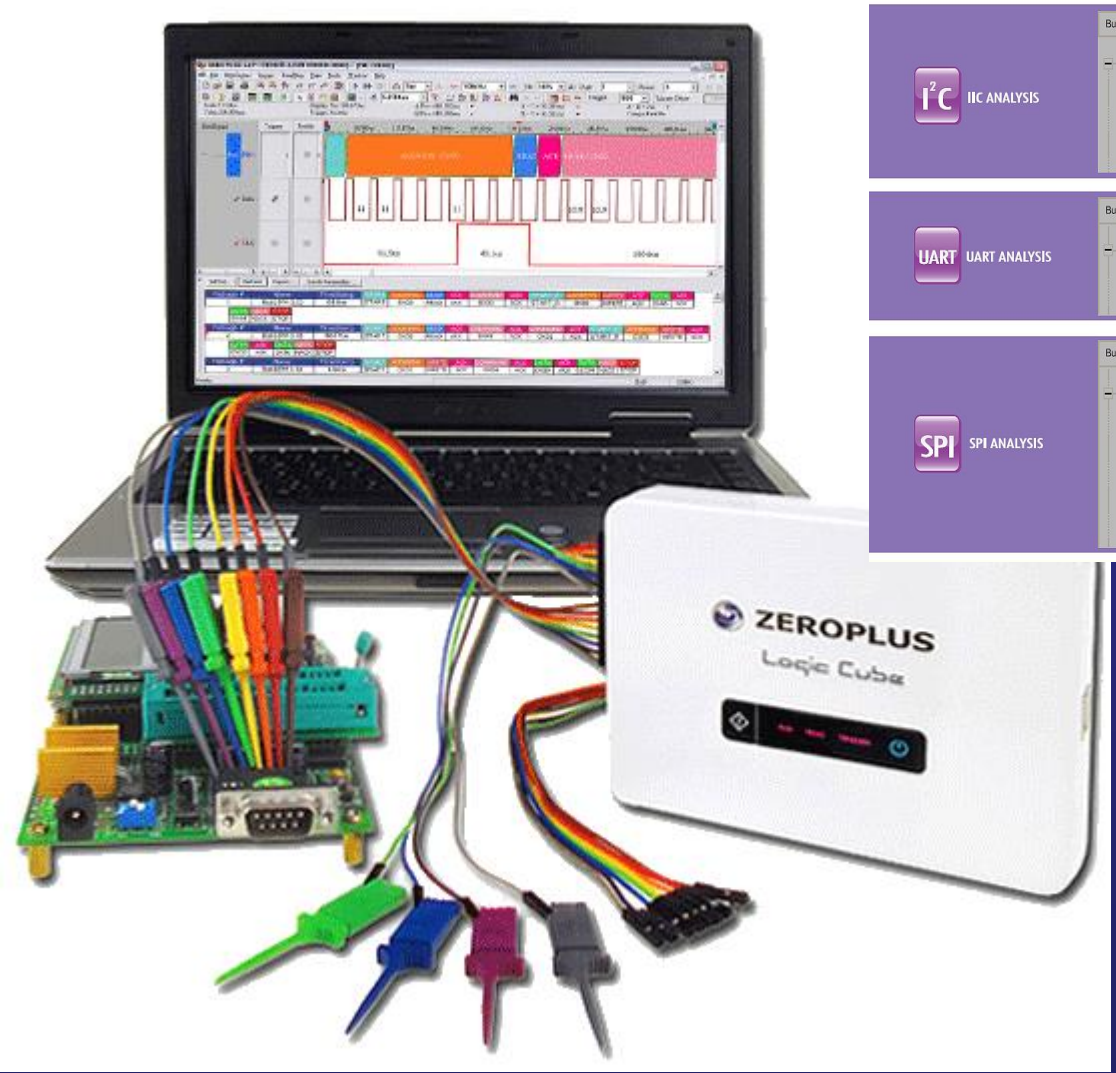


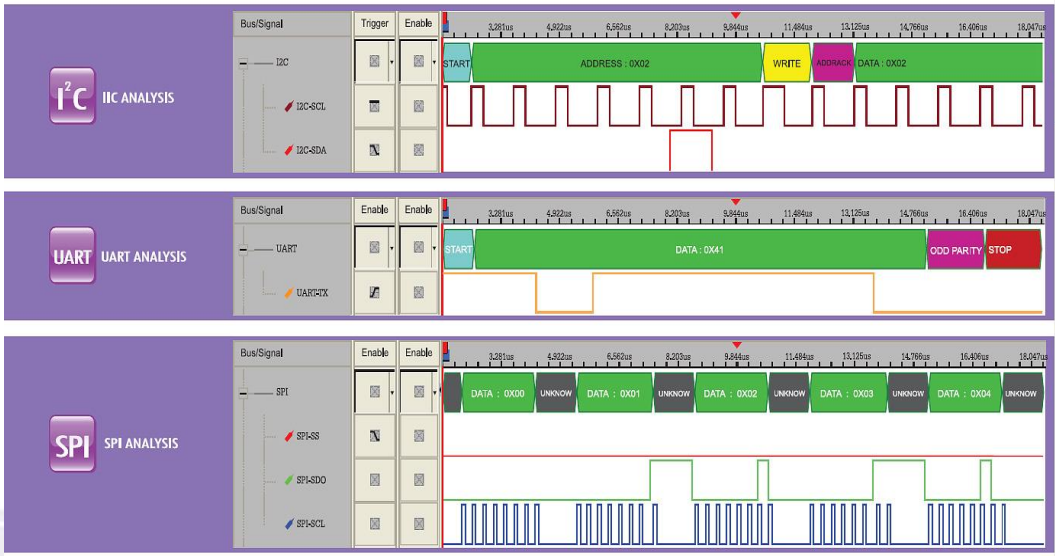
Rejestrowanie w czasie stanów logicznych

Możliwość zapamiętania danych w PC

Przyrządy używane na pracowni

Analizator Stanów Logicznych



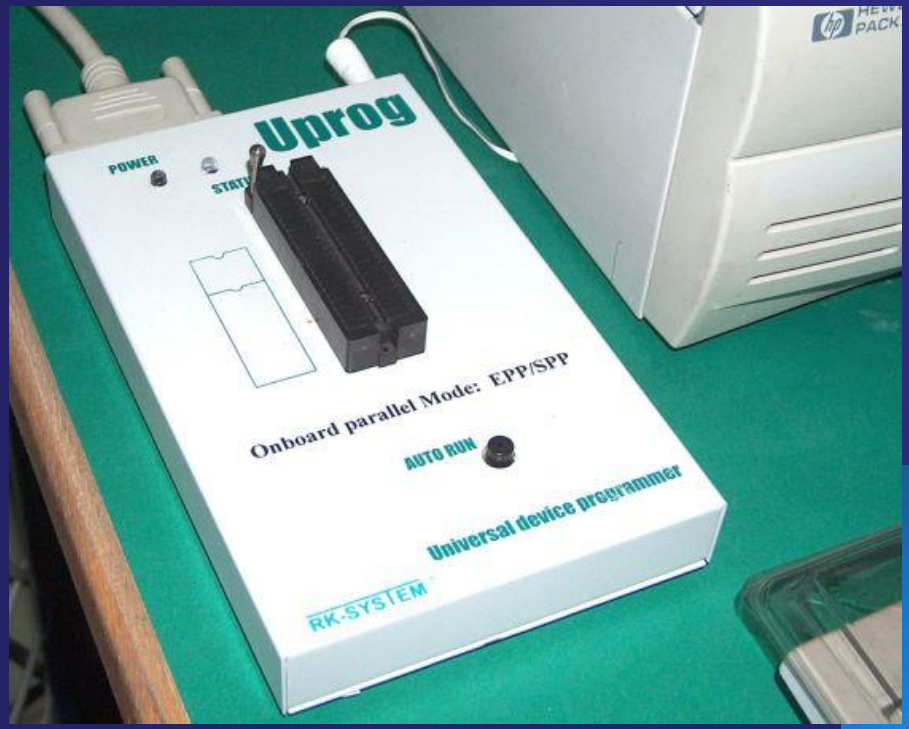


UMOŻLIWIA

Rejestrowanie w czasie stanów logicznych i dekodowanie sygnałów rożnych protokołów i magistral

Przyrządy używane na pracowni

Programatory mikroprocesorów



Programator Szeregowy ISP

Programatory równoległe

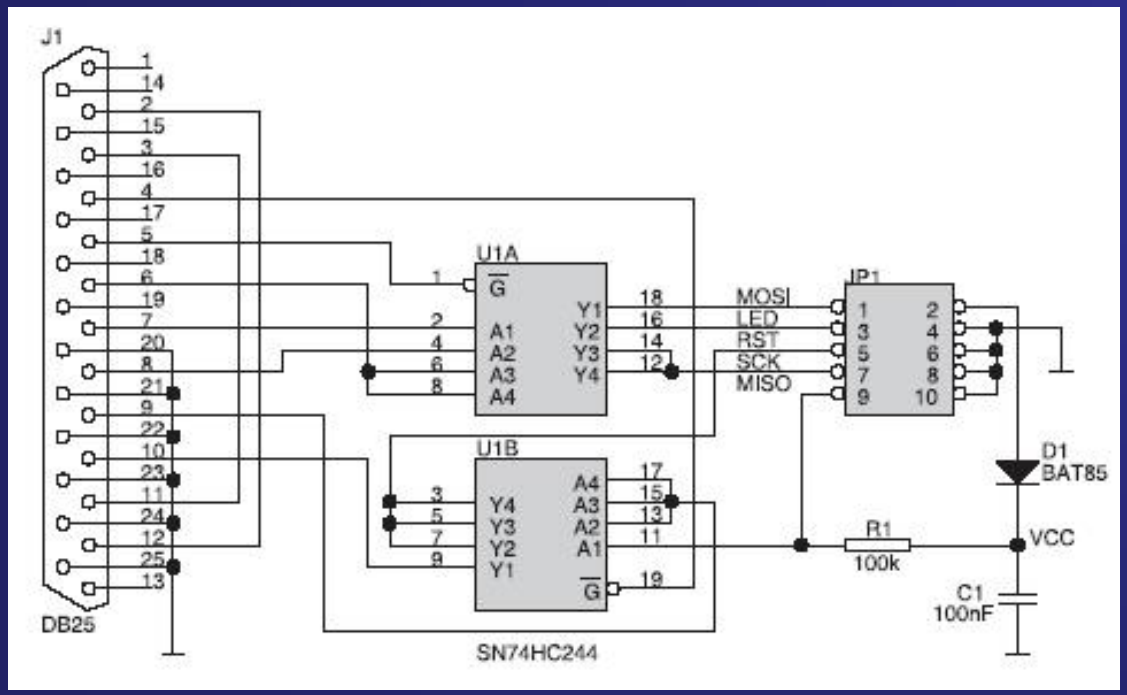

Lab. Mikroprocesory
Mikrokontroler ATmega8

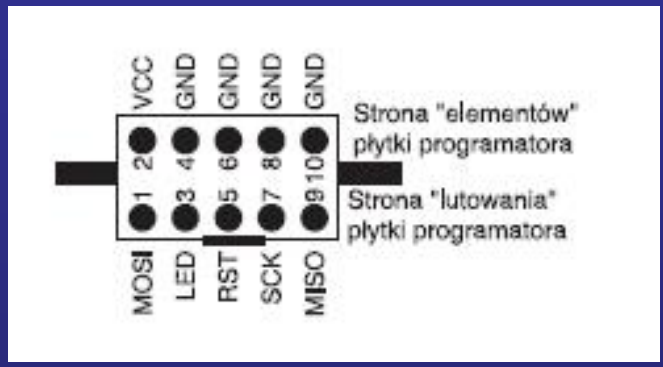
Programowanie szeregowe poprzez interfejs ISP

Serial Peripheral Interface

Programowanie

Programowanie szeregowe poprzez interfejs ISP Serial Peripheral Interface)





Programowanie

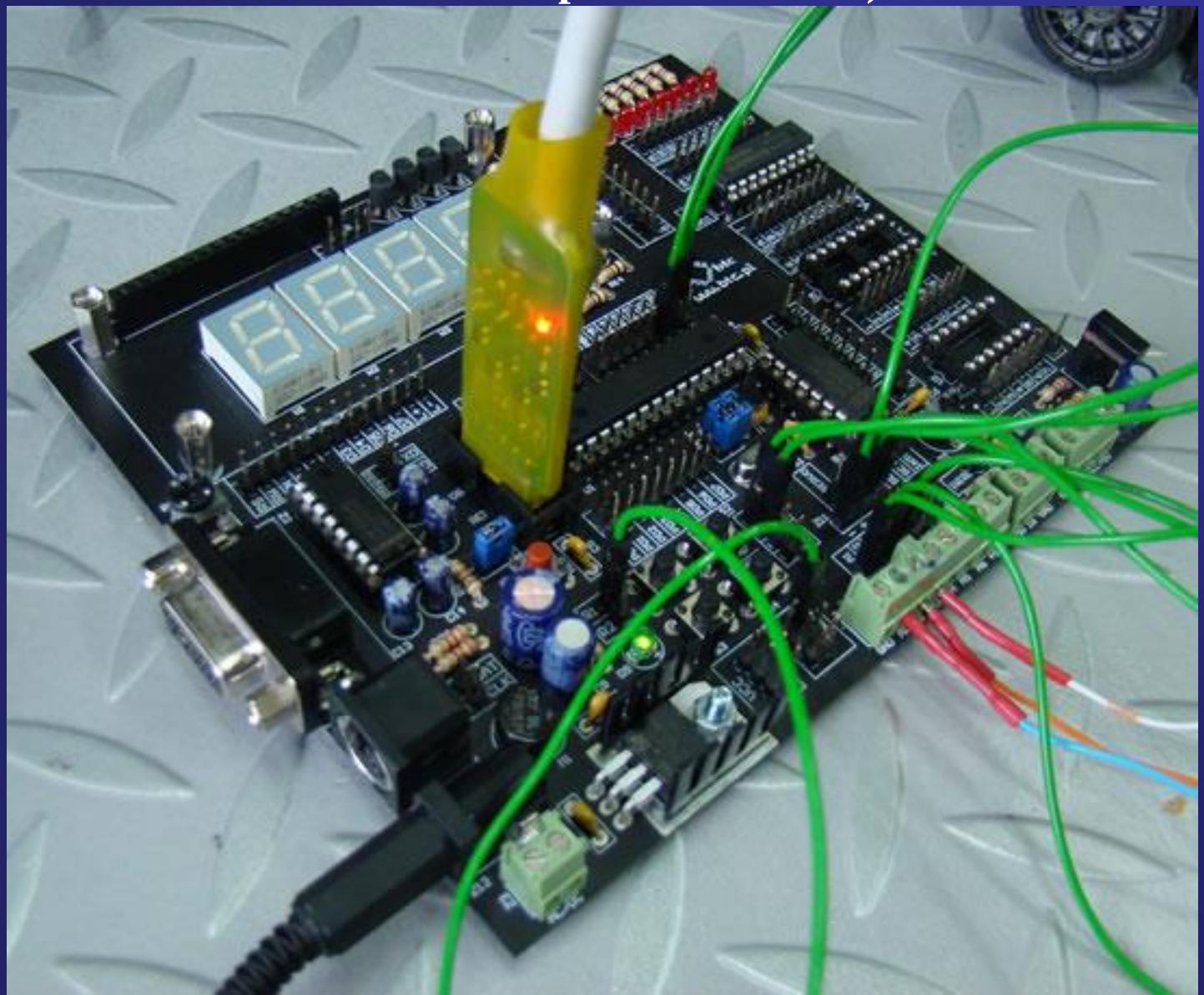
Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)

Programowanie

Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)

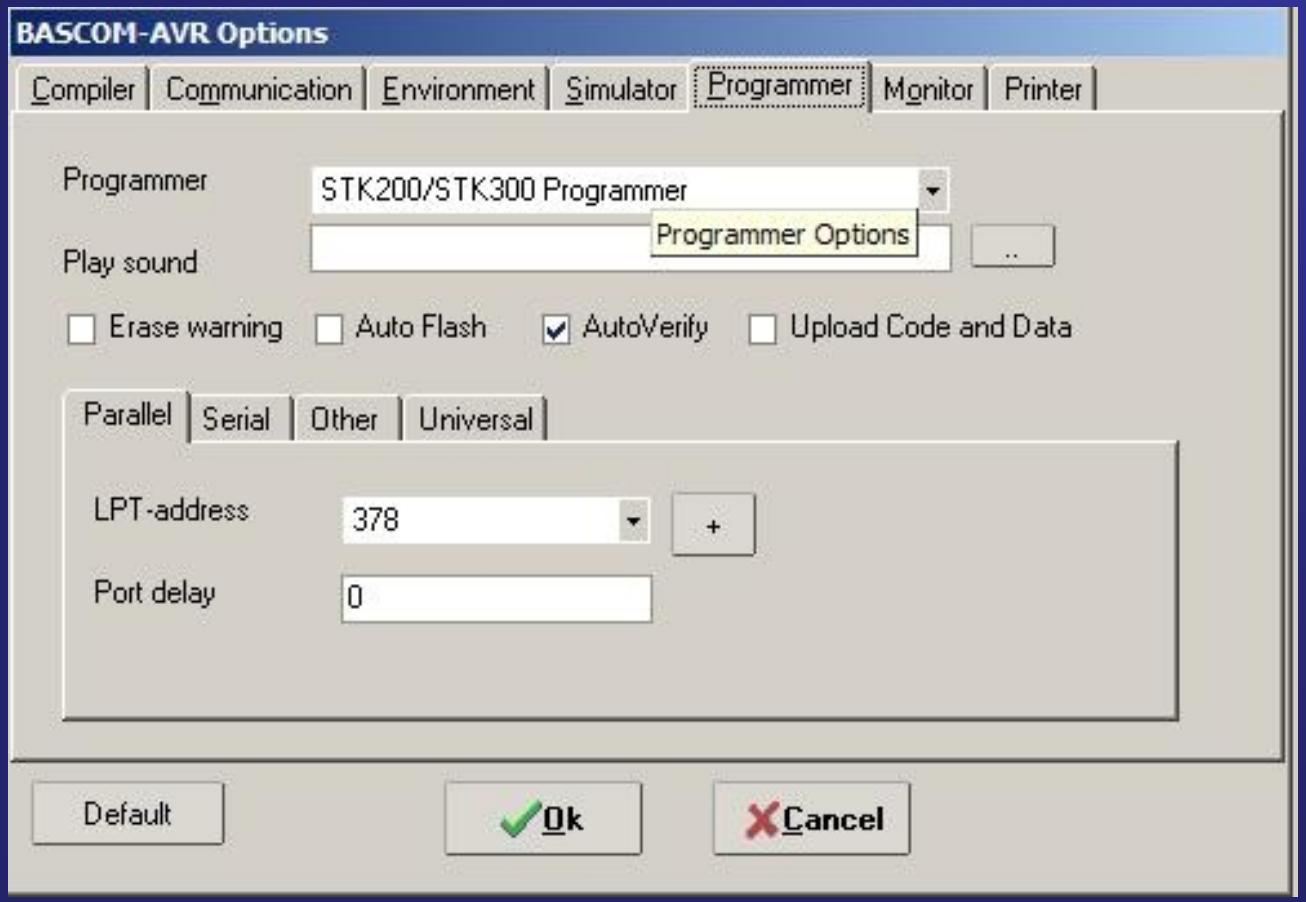
Programowanie

Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)



Programowanie

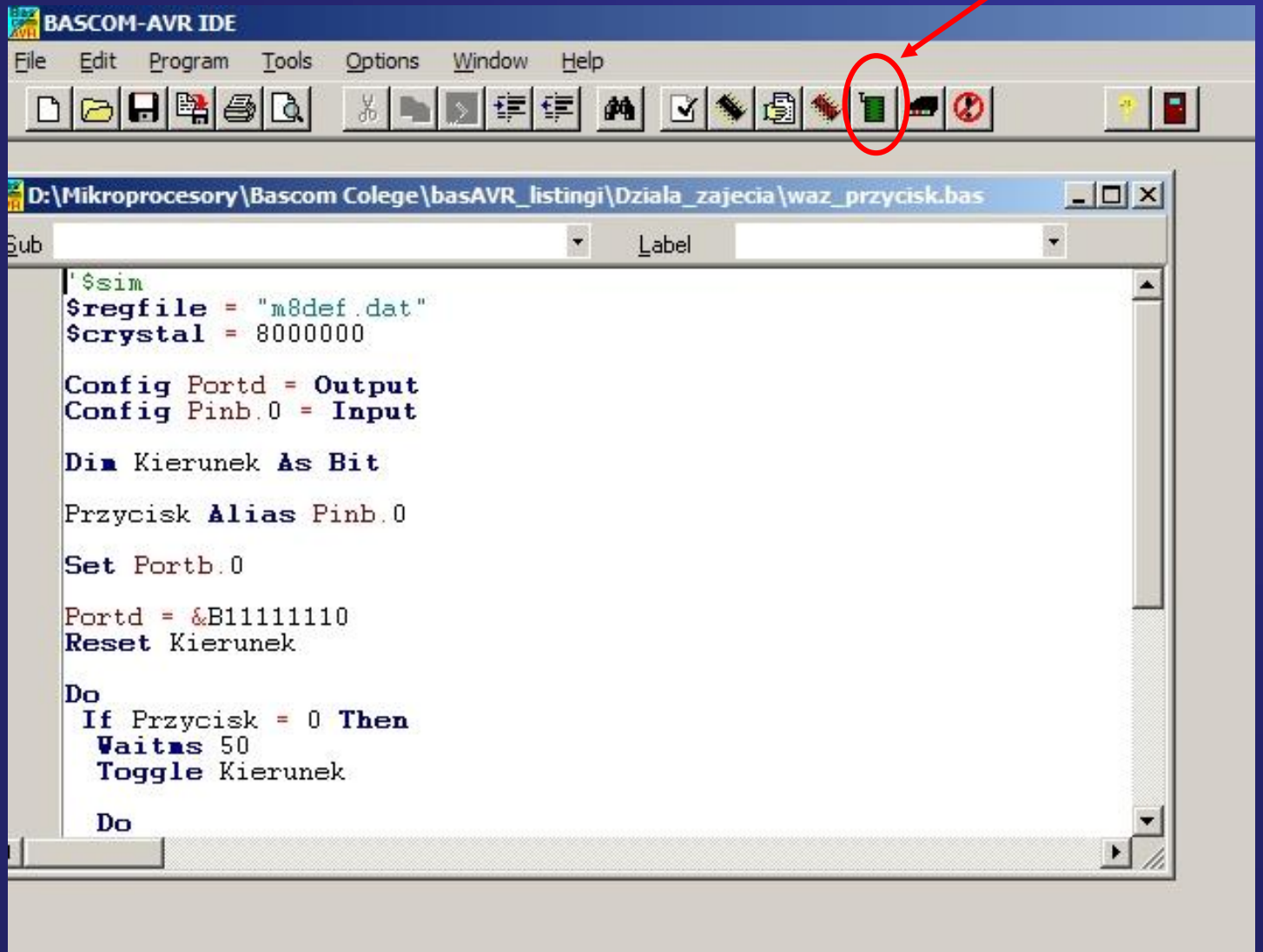
Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)



Programowanie

Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)

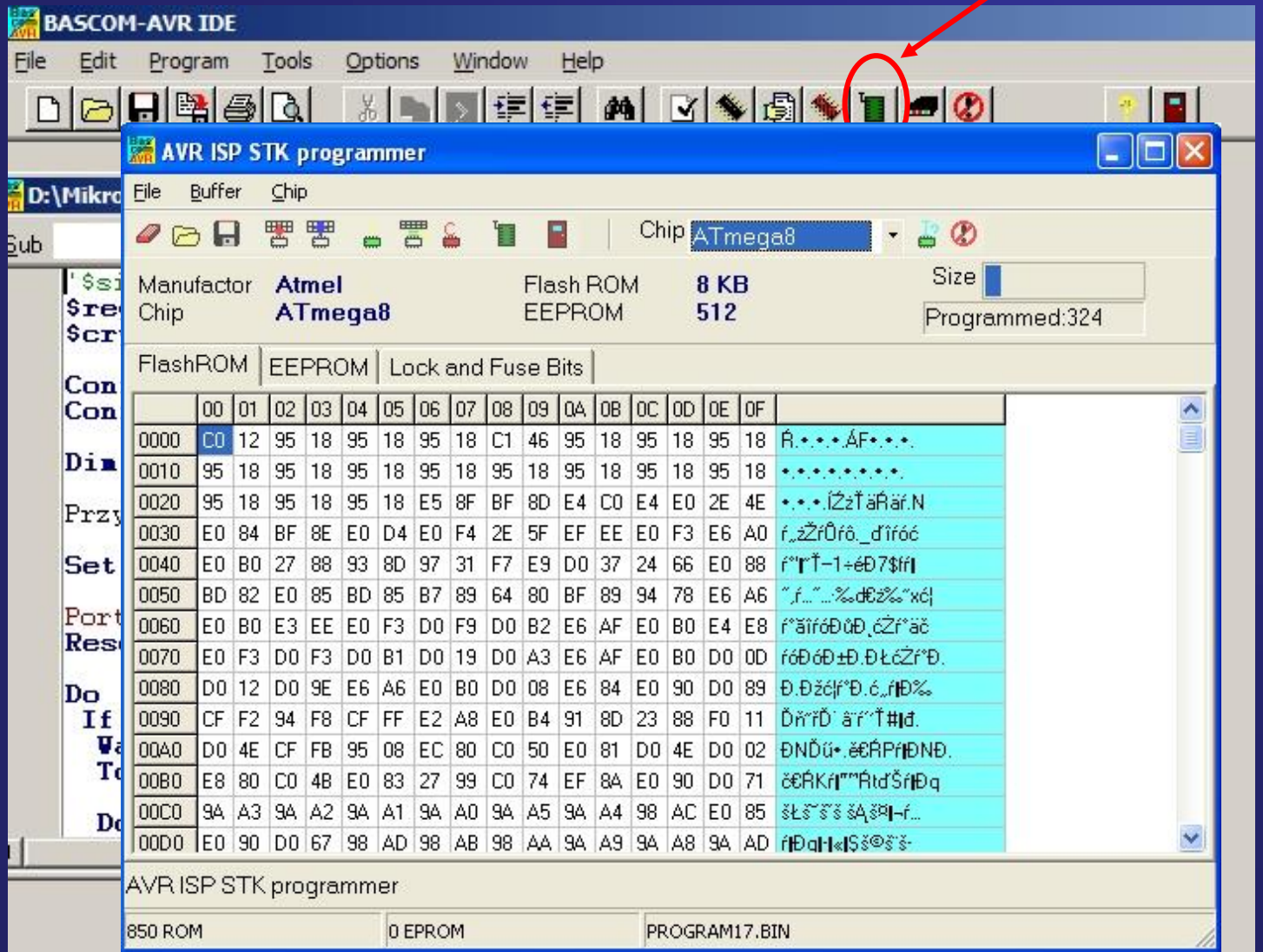
Ikona programatora

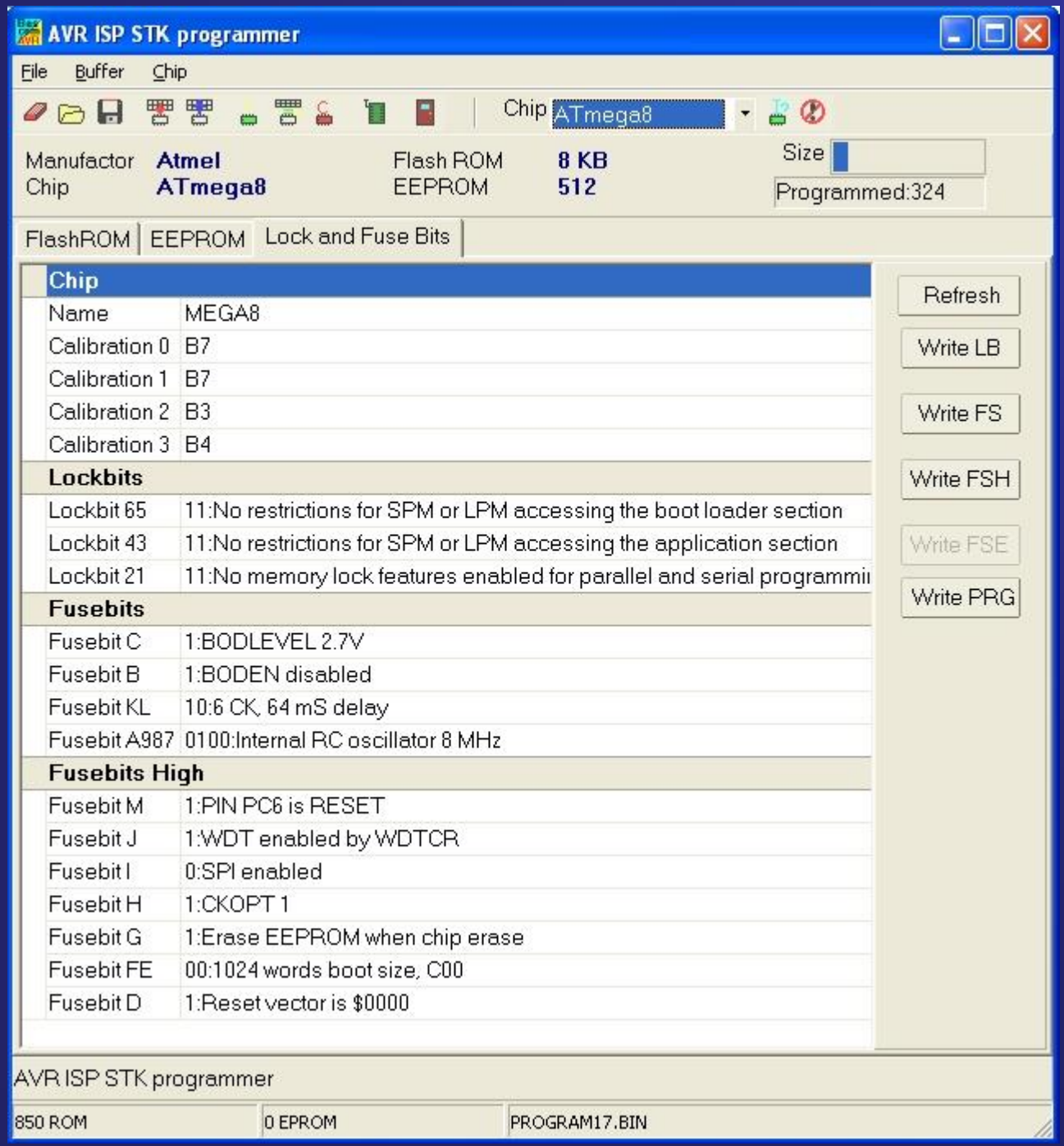


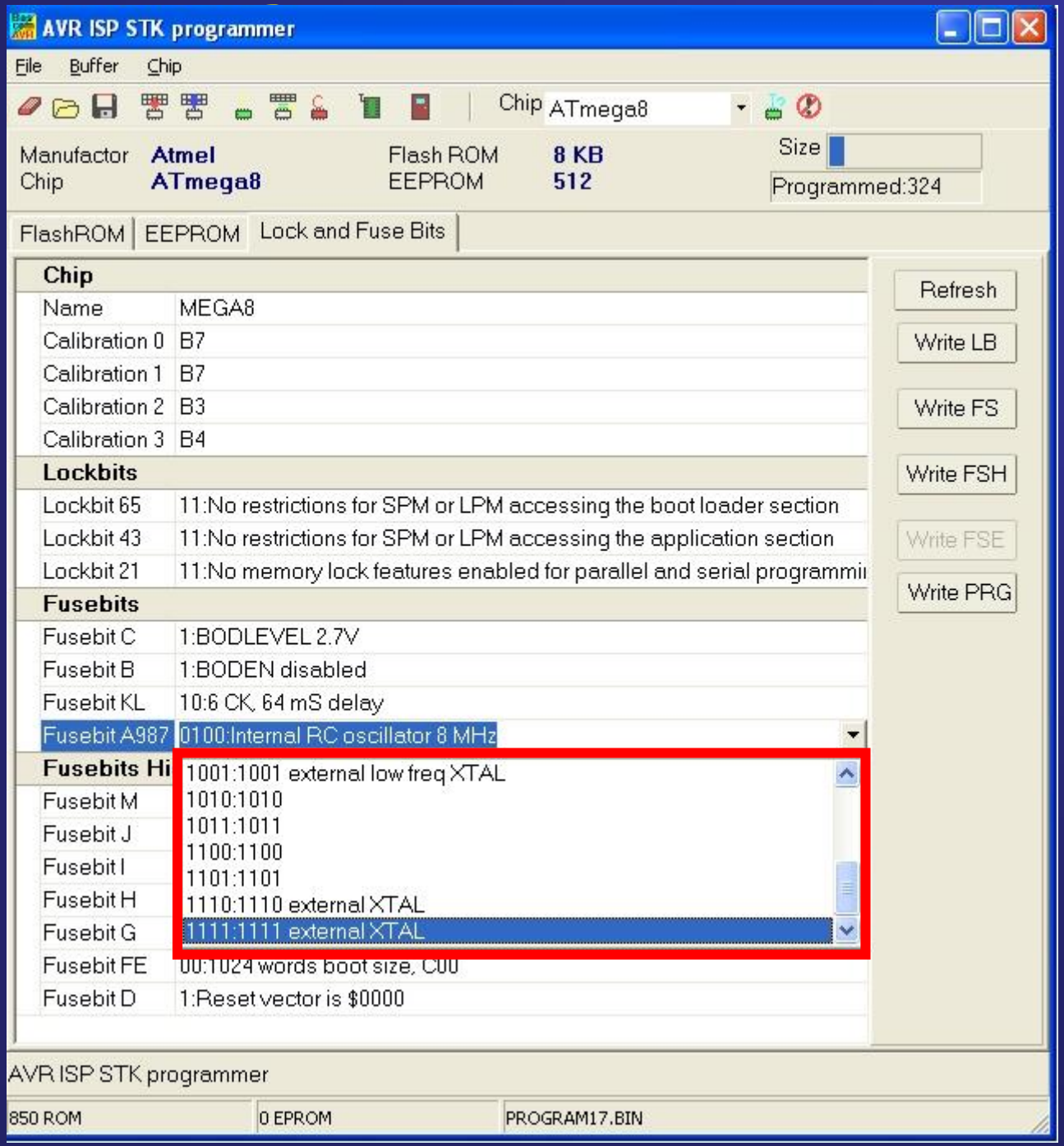
Programowanie

Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)

Ikona programatora




AVR ISP STK programmer
File
Buffer
Chip
Chip ATmega8
Manufactor Atmel
Chip ATmega8
Flash ROM 8 KB
EEPROM 512
Size
Programmed:324
FlashROM
EEPROM
Lock and Fuse Bits
Chip
Name MEGA8
Calibration 0 B7
Calibration 1 B7
Calibration 2 B3
Calibration 3 B4
Lockbits
Lockbit 65 11:No restrictions for SPM or LPM accessing the boot loader section
Lockbit 43 11:No restrictions for SPM or LPM accessing the application section
Lockbit 21 11:No memory lock features enabled for parallel and serial programmi
Fusebits
Fusebit C 1:BODLEVEL 2.7V
Fusebit B 1:BODEN disabled
Fusebit KL 10:6 CK, 64 mS delay
Fusebit A987 0100:Internal RC oscillator 8 MHz
Fusebits High
Fusebit M 1:PIN PC6 is RESET
Fusebit J 1:WDT enabled by WDTCR
Fusebit I 0:SPI enabled
Fusebit H 1:CKOPT 1
Fusebit G 1:Erase EEPROM when chip erase
Fusebit FE 00:1024 words boot size, C00
Fusebit D 1:Reset vector is $0000
Refresh
Write LB
Write FS
Write FSH
Write FSE
Write PRG
AVR ISP STK programmer
850 ROM
0 EPROM
PROGRAM17.BIN


AVR ISP STK programmer
File
Buffer
Chip
Chip ATmega8
Manufactor Atmel
Chip ATmega8
Flash ROM 8 KB
EEPROM 512
Size
Programmed:324
FlashROM
EEPROM
Lock and Fuse Bits
Chip
Name MEGA8
Calibration 0 B7
Calibration 1 B7
Calibration 2 B3
Calibration 3 B4
Lockbits
Lockbit 65 11:No restrictions for SPM or LPM accessing the boot loader section
Lockbit 43 11:No restrictions for SPM or LPM accessing the application section
Lockbit 21 11:No memory lock features enabled for parallel and serial programmi
Fusebits
Fusebit C 1:BODLEVEL 2.7V
Fusebit B 1:BODEN disabled
Fusebit KL 10:6 CK, 64 mS delay
Fusebit A987 0100:Internal RC oscillator 8 MHz
Fusebits Hi
1001:1001 external low freq XTAL
1010:1010
1011:1011
1100:1100
1101:1101
1110:1110 external XTAL
1111:1111 external XTAL
Fusebit M
Fusebit J
Fusebit I
Fusebit H
Fusebit G
Fusebit FE 00:1024 words boot size, C00
Fusebit D 1:Reset vector is $0000
Refresh
Write LB
Write FS
Write FSH
Write FSE
Write PRG
AVR ISP STK programmer
850 ROM
0 EPROM
PROGRAM17.BIN

Programowanie

Programowanie szeregowe poprzez interfejs ISP
Serial Peripheral Interface)