

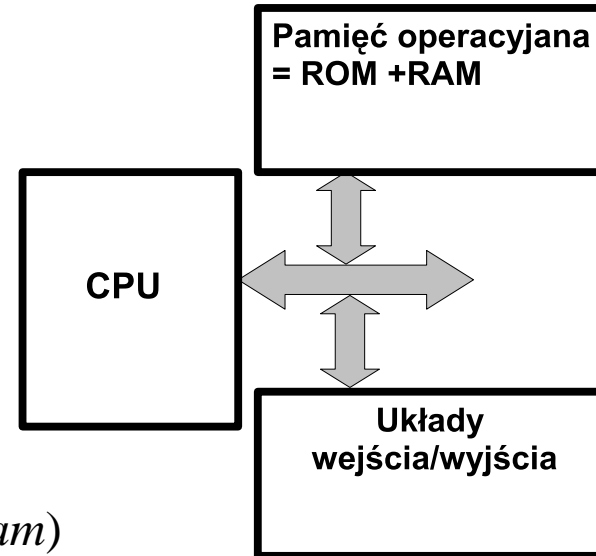
Elementy składowe systemu komputerowego

Podstawowe elementy składowe:

- procesor z ALU
- pamięć komputera (zawierająca *dane i program*)
- urządzenia wejścia/wyjścia

Cechy komputera von Neumanna:

- skończona lista rozkazów,
- możliwość wprowadzenia programu i jego przechowywanie w pamięci (tak jak dane),
- sekwencyjne odczytywanie instrukcji z pamięci i ich wykonywanie.



Architektura harwardzka

- *Pamięć danych programu* jest oddzielona od *pamięci rozkazów* (inaczej niż w architekturze von Neumanna).
- Prostsza, w stosunku do architektury von Neumanna, budowa ma większą szybkość działania - wykorzystuje się w procesorach sygnałowych oraz przy dostępie procesora do pamięci cache.
- *Architektura harwardzka* jest obecnie powszechnie stosowana w *mikrokomputerach jednoukładowych* (program w pamięci ROM (ang. Read Only Memory), dane w RAM (Random Access Memory)).

Mikrokontrolery

Mikrokontroler - komputer zrealizowany w postaci pojedynczego układu scalonego, zawierającego jednostkę centralną (CPU), pamięć RAM, na ogół, pamięć programu oraz rozbudowane układy wejścia-wyjścia.

Określenie mikrokontroler pochodzi od głównego obszaru zastosowań, jakim jest sterowanie urządzeniami elektronicznymi.

Typowy mikrokontroler zawiera:

- Jednostkę obliczeniową (ALU) - przeważnie 8-bitową,
- Pamięć danych (RAM),
- Pamięć programu,
- Uniwersalne porty wejścia - część tych portów może pełnić alternatywne funkcje, wybierane programowo,
- Kontrolery transmisji szeregowej lub równoległej (UART, SPI, I2C, USB, CAN, itp.),
- Przetworniki analogowo-cyfrowe lub cyfrowo-analogowe,
- timery,
- Układ kontroli poprawnej pracy (watchdog)
- wewnętrzne czujniki wielkości nieelektrycznych (np. temperatury)

Systemy wbudowane - **definicje**

- Ogólna definicja *systemów wbudowanych* określa je jako urządzenia używane do kontroli, monitoringu lub wspomagania pracy urządzeń i maszyn. Pojęcie "wbudowane" odnosi się do faktu, że stanowią one integralną częścią systemu w którym pracują. W wielu przypadkach obecność *systemów wbudowanych* może nie być oczywista dla niewtajemniczonych obserwatorów.
- System wbudowany (ang. Embedded system) - system komputerowy specjalnego przeznaczenia, który staje się integralną częścią obsługiwanego przez niego sprzętu.

Systemy wbudowane - cd.

- System wbudowany spełnia określone wymagania, zdefiniowane do zadań które ma wykonywać,
- Typowy system wbudowany oparty jest na mikroprocesorze (lub mikrokontrolerze), zaprogramowanym do wykonywania konkretnych zadań,
- Niektóre systemy wbudowane zawierają system operacyjny.

Charakterystyka systemów wbudowanych

- *Program* - zadania jakie ma wykonywać program znane są a priori,
- *Statyczne* planowanie i alokacja zasobów systemu,
- *Systemy czasu rzeczywistego* - kompromis pomiędzy użytym sprzętem i oprogramowaniem, obsługa sytuacji wyjątkowych,
- *Interakcja* pomiędzy systemem wbudowanym a zewnętrznym środowiskiem,
- *Hierarchia zachowań* - sekwencja zachowań, konkurencyjne scenariusze zachowań.

System operacyjny - podstawowe zadania

System operacyjny (ang. skrót OS Operating System) – oprogramowanie zarządzające sprzętem komputerowym, tworzące środowisko do **uruchamiania i kontroli zadań** użytkownika.

Główne zadania systemu operacyjnego:

- Zarządzanie zasobami systemu w tym: *czasem procesora, pamięcią operacyjną,*
- Dostarcza mechanizmów do synchronizacji zadań i komunikacji pomiędzy zadaniami,
- Zapewnienia równoległe wykonywanym zadaniom jednolity, wolny od interferencji dostępu do sprzętu,
- Zarządzanie bezpieczeństwem (np. dostępem do zasobów),
- Inne np. ustalanie połączeń sieciowych, zarządzanie plikami.

Systemy wbudowane z systemami operacyjnymi

Główne *systemy operacyjne* stosowane w *systemach wbudowanych*

- *QNX* - system czasu rzeczywistego,
- *Embedded Linux*,
- *Windows CE*,
- *VxWorks*.

Procesory wspierane przez systemy operacyjne

- *QNX*: all generic x86 based processors(386+),
- *Linux*: procesory ogólnego przeznaczenia - ARM, StrongARM, MIPS, Hitachi SH, PowerPC, x86,
- *Windows CE*: x86, MIPS, Hitachi SH3 and SH4, PowerPC and StrongArm processors
- *VxWorks*: PowerPc, 68K, CPU32, ColdFire, MCore, 80x86 and Pentium, i960, ARM and StrongARM, MIPS, SH, SPARC, NECV8xx, M32 R/D, RAD6000, ST 20, TriCore

Minimalne wymagania pamięci

Dla *systemów operacyjnych* stosowanych w *systemach wbudowanych* wymaga się następujących zasobów pamięci:

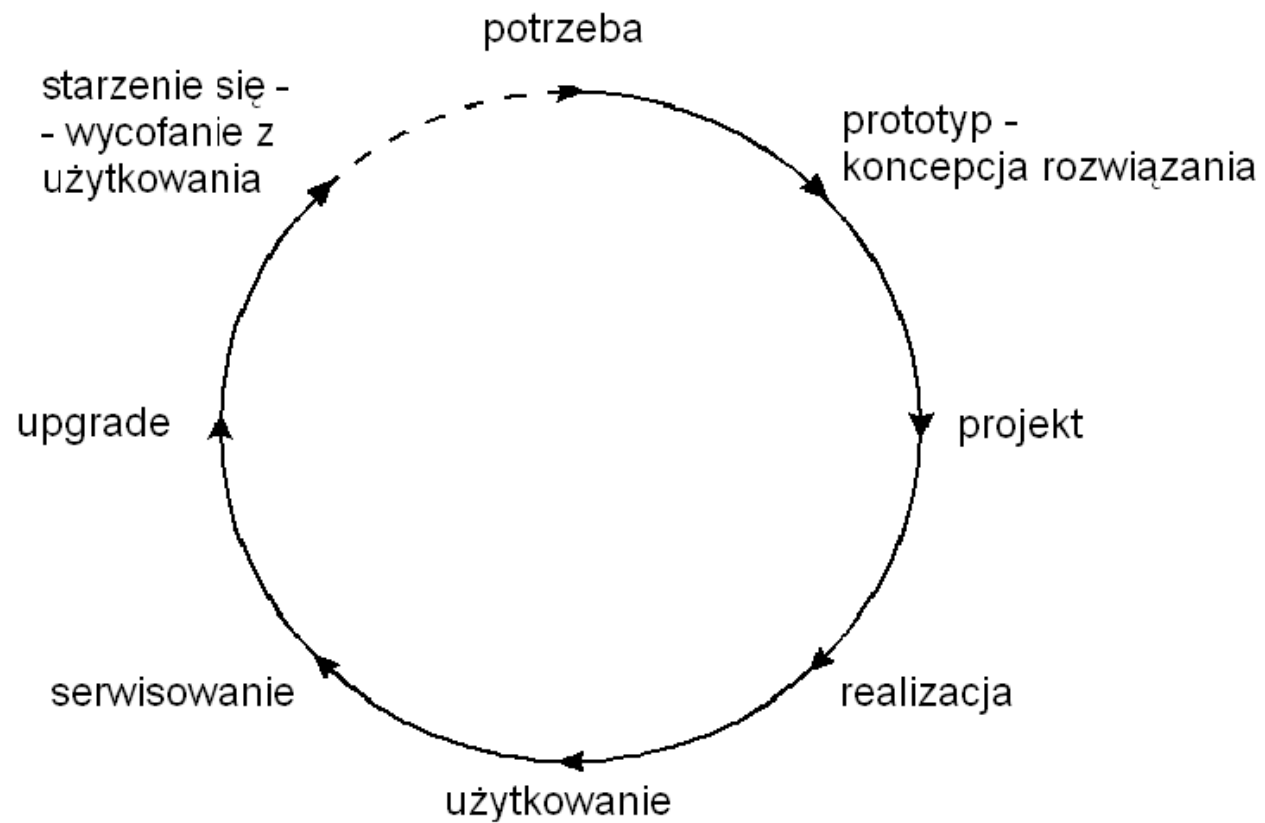
- *QNX*: ma najmniejsze wymagania od ok. $10KB$,
- *Windows CE*: wymaga minimum $350KB$,
- *Linux*: wymaga $125 - 256KB$ dla sensownej konfiguracji jądra,
- *VxWorks*: kilka kilobajtów najprostszego systemu wbudowanego.

Dokładniejsza charakterystyka systemów operacyjnych (w kontekście systemów wbudowanych) na następnych wykładach.

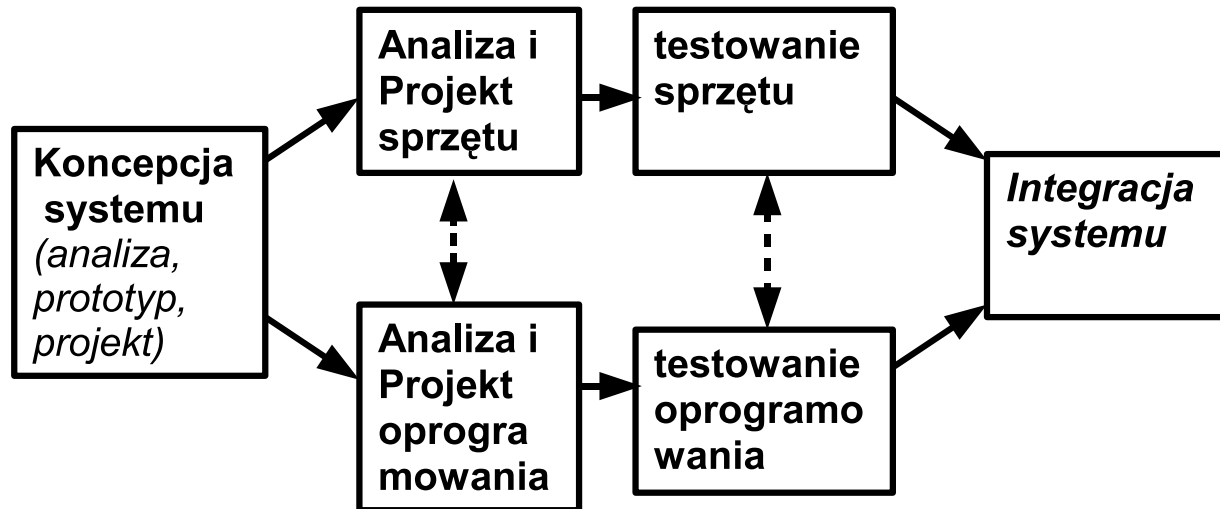
Systemy wbudowane a Komputery PC

- Prosty interfejs użytkownika (wyświetlacz, port, dioda, przyciski),
- Za pomocą portu diagnozowany może być system, w którym pracuje system wbudowany, a nie sam system wbudowany,
- Zadania programu mogą nie być bezpośrednio widoczne (program napisany pod specyficzne zastosowania).

Projektowanie systemów wbudowanych - cykl życia



Współbieżne projektowanie sprzętu i oprogramowania

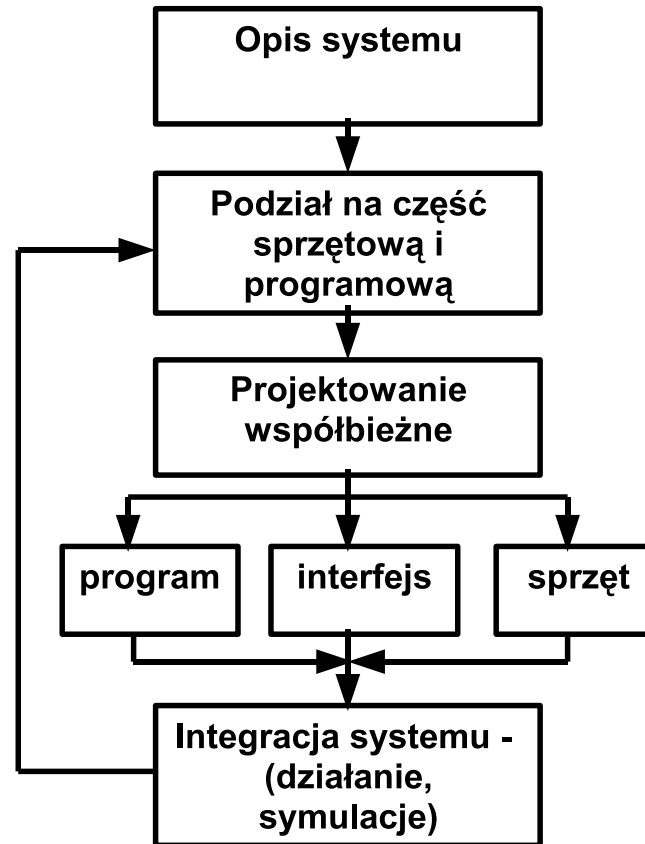


- Kooperacyjny projektowanie oprogramowania i sprzętu,
- Unifikacja osobno projektowanego oprogramowania i sprzętu,
- Wymienność pod względem funkcjonalności sprzętu i oprogramowania,

Współbieżne projektowanie sprzętu i oprogramowania - cd.

- Wspólne "wspieranie się" sprzętu i oprogramowania pod kątem stawianych celów systemowych,
- Współbieżne projektowania sprzętu i oprogramowania jest szczególnie istotne, gdy sprzęt jest jednym układem scalonym,
- W przypadku używania gotowych układów scalonych, dedykowanych do specjalnych zastosowań, cały ciężar projektowy spoczywa na części programowej,
- System powinien mieć zdolność do adaptacji do zmieniającego się środowiska, lub do niekompletnej specyfikacji.

Proces projektowania *układów wbudowanych*



- Uproszczony schemat procesu projektowania *układu wbudowanego*.

Projektowanie *systemów wbudowanych* wykorzystaniem mikrokontrolerów jednokładowych

- Wybór odpowiedniego dla danego zadania mikrokontrolera ze względu na procesor, wymagane peryferia czy systemy komunikacji,
- Proces projektowania redukuje się wówczas do oprogramowania kontrolera. Pomocnym mogą być wówczas:
 - kompilatory języków wyższego rzędu,
 - dostępne biblioteki,
 - symulatory,
 - cross-kompilacja.

Przegląd obecnych mikrokontrolerów

Do najbardziej popularnych mikrokontrolerów należą:

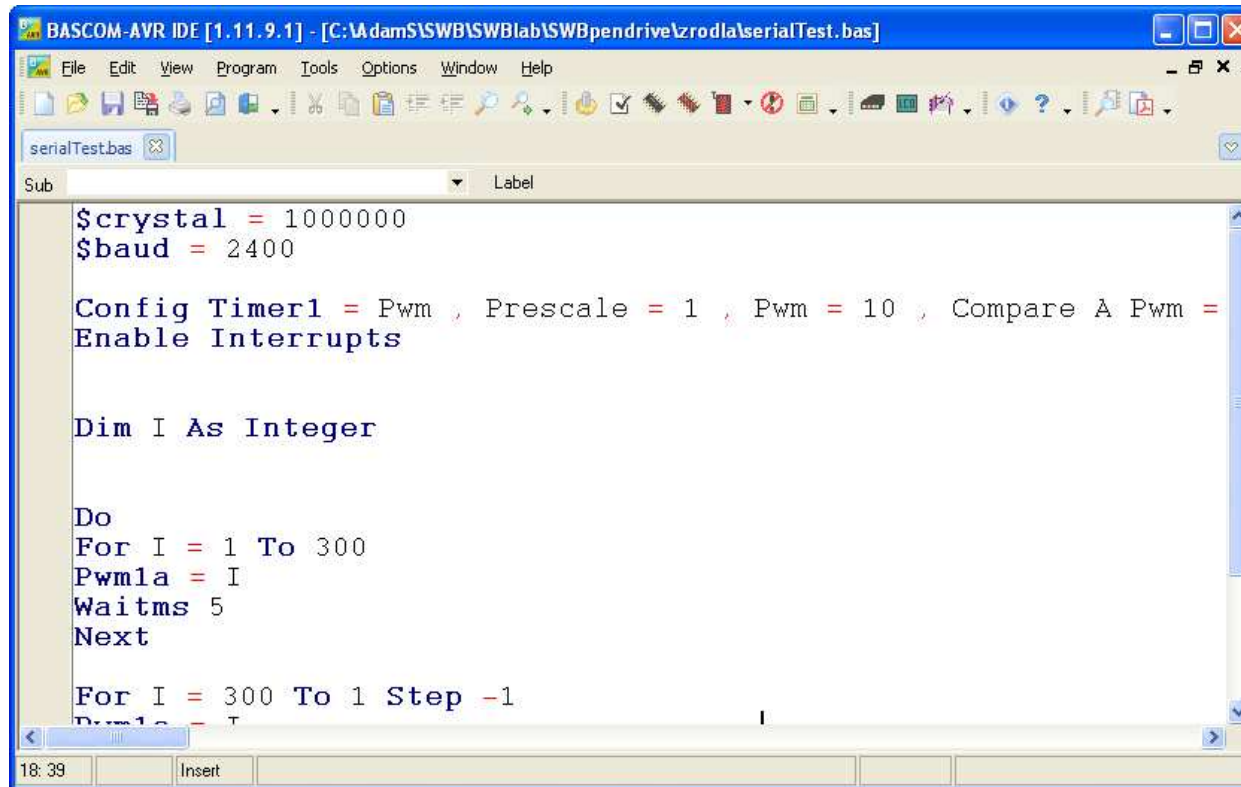
1. Niekwestionowany standard dla rynku masowego narzuciła firma *Intel*, która wprowadziła na rynek mikrokontroler *8051*,
2. Bardzo popularne są również mikrokontrolery *AVR* firmy *Atmel* - w oparciu o nie będą prowadzona zajęcia laboratoryjne,
3. *PIC* firmy *Microchip Technology*,
4. inne.

Układy AVR

W dalszej części skupimy się głównie na *układach AVR* w tym na:

1. Środowisko programistyczno-uruchomieniowe *AVR Studio*,
2. Środowisko programistyczno-uruchomieniowe wraz z symulatorem *BASCOM-AVR*.

Bascom-AVR



```
Sub
$crystal = 1000000
$baud = 2400

Config Timer1 = Pwm , Prescale = 1 , Pwm = 10 , Compare A Pwm =
Enable Interrupts

Dim I As Integer

Do
For I = 1 To 300
Pwm1a = I
Waitms 5
Next

For I = 300 To 1 Step -1
Pwm1a = I
```

- Wygląd głównego okna programu Bascom-AVR

Sprzęt obsługiwany przez Bascom-AVR

- Sprzęt zintegrowany w układzie scalonym
 - Timery (TIMER0 i TIMER1) i liczniki,
 - Rejestry wewnętrzne,
 - Port A i B,
 - Watchdog,
- obsługa zewnętrznych urządzeń
 - LCD
 - UART - możliwość emulator terminala
 - I2C
 - 1 WIRE protocol
 - SPI protocol w tym *In System Programming (ISP)*.

Język programowania używany w Bascom-AVR

- BASIC
- Assembler

Basic - znaki i operatory

Character	Name
ENTER	Terminates input of a line
	Blank (or space)
'	Single quotation mark (apostrophe)
*	Asterisks (multiplication symbol)
+	Plus sign
,	Comma
-	Minus sign
.	Period (decimal point)
/	Slash (division symbol) will be handled as \
:	Colon
"	Double quotation mark
;	Semicolon
<	Less than
=	Equal sign (assignment symbol or relational operator)
>	Greater than
\	Backslash (integer/word division symbol)
^	Exponent

Operator	Relation Tested	Expression
=	Equality	$X = Y$
<>	Inequality	$X <> Y$
<	Less than	$X < Y$
>	Greater than	$X > Y$
<=	Less than or equal to	$X <= Y$
>=	Greater than or equal to	$X >= Y$

Operator	Meaning
NOT	Logical complement
AND	Conjunction
OR	Disjunction
XOR	Exclusive or

Basic - typy zmiennych

- Bit ($\frac{1}{8}$ byte),
- Byte (1 byte) - liczba 8-bitowa bez znaku ($0 \div 255$),
- Integer (dwa bajty) - liczby całkowitoliczbowe ze znakiem ($-32,768 \div +32,767$),
- Word (dwa bajty) - liczba całkowitoliczbowa bez znaku ($0 \div 65535$),
- Long (cztery bajty) - liczby całkowitoliczbowe ze znakiem ($-2147483648 \div 2147483647$),
- Single (32 bity) liczba zmiennoprzecinkowa ze znakiem ($1.5x10^{-45} \div 3.4x10^{38}$),
- Double (64 bity) liczba zmiennoprzecinkowa ze znakiem ($5.0x10^{-324} \div 1.7x10^{308}$),
- String (do 254 bajtów) -przechowywane jako bajty i zakończone 0-bajtem tj. rozmiar stringa jest o jeden większy.

Układy AVR nie posiadają koprocatora. Operacje na liczbach zmiennoprzecinkowych (*Single, Double*) są emulowane.

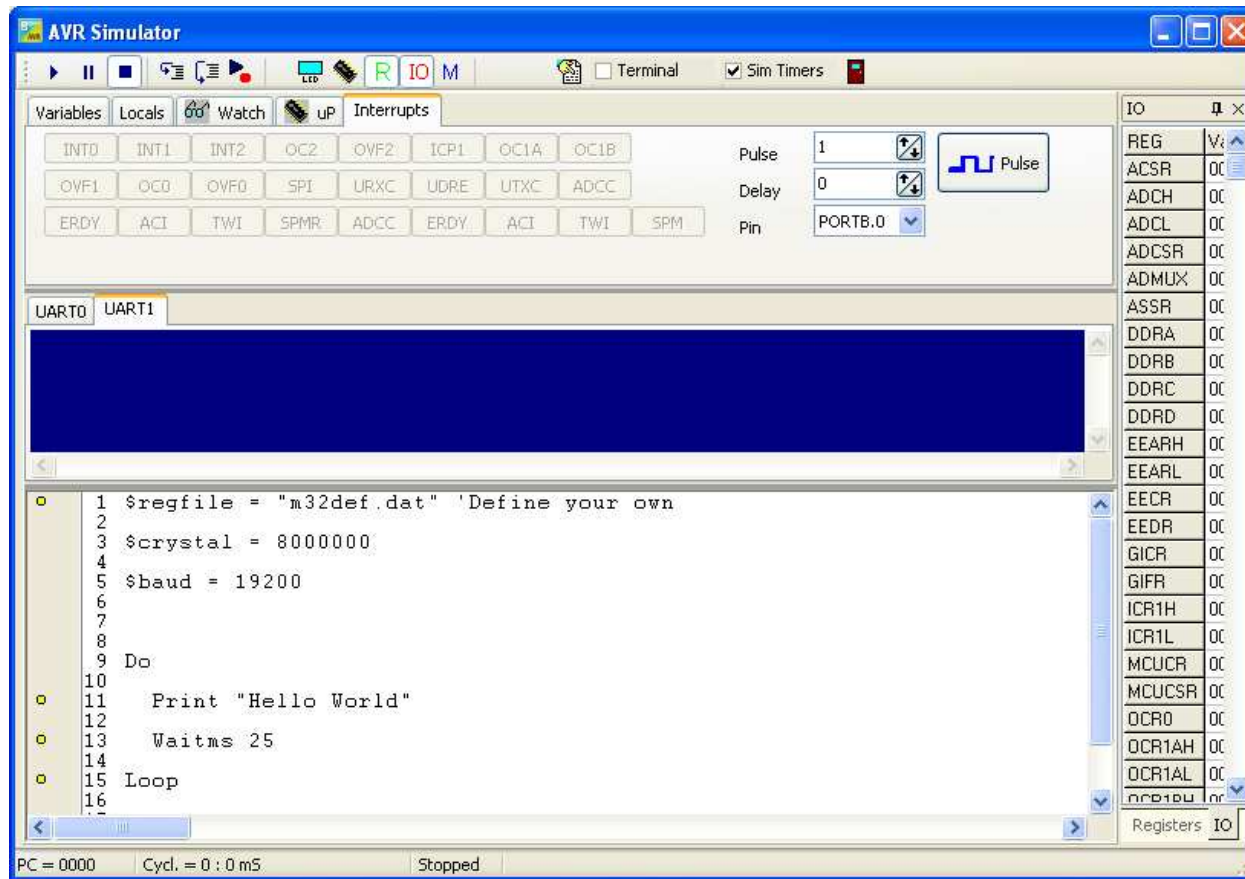
Manual języka dostępny jest w pomocy

- W szczególności opis funkcji (wraz z przykładami użycia) potrzebne na ćwiczeniach jak *waitkey()*, *input()* dostępny są w helpie !

Biblioteka wspomagająca assembler Baskom-AVR

- I2C, Extended I2C,
- MCSBYT, MCSBYTEINT - wspomagają konwersję typów ze stringa,
- TCPIP,
- Floating Point,
- LCD,
- CF Card,
- SPI,
- Data i czas,
- inne.

Symulator w Bascom-AVR



The screenshot displays the AVR Simulator interface. The main window is titled "AVR Simulator" and contains several panels:

- Top Panel:** Includes a toolbar with simulation controls (play, pause, stop, reset) and checkboxes for "Terminal" and "Sim Timers".
- Interrupts Panel:** A grid of buttons for various interrupts: INTD, INT1, INT2, OC2, OVF2, ICP1, OC1A, OC1B, OVF1, OC0, OVF0, SPI, URXC, UDRE, UTXC, ADCC, ERDY, ACI, TWI, SPMR, ADCC, ERDY, ACI, TWI, SPM.
- Pulse Generator:** A section for generating a pulse with a value of 1, a delay of 0, and a pin set to PORTB.0. A pulse waveform icon is visible.
- UART Panel:** A section for UART0 and UART1, currently showing a blue screen.
- Code Editor:** A text area containing the following code:

```
1 $regfile = "m32def.dat" 'Define your own
2
3 $crystal = 8000000
4
5 $baud = 19200
6
7
8
9 Do
10
11   Print "Hello World"
12
13   Waitms 25
14
15 Loop
16
```
- IO Panel:** A list of registers and I/O devices, including REG, ACSR, ADCH, ADCL, ADCSR, ADMUX, ASSR, DDRA, DDRB, DDRC, DDRD, EEARH, EEARL, EECR, EEDR, GICR, GIFR, ICR1H, ICR1L, MCUCR, MCUCSR, OCR0, OCR1AH, OCR1AL, and PORTB.
- Status Bar:** Shows "PC = 0000", "Cycl. = 0 : 0 mS", and "Stopped".

Zadania na ćwiczenia

Wszystkie zadania należy wykonać w środowisku BASCOM-AVR

1. Zrealizuj kalkulator wykonujący operacje dodawania "+" i odejmowania "-" na dwóch operandach w notacji infiksowej. Do wczytywania użyj funkcji *input()*. Każdą daną (w tym liczby i znaki) należy zatwierdzić enterem.
2. Zrealizuj kalkulator wykonujący operacje dodawania "+", odejmowania "-", mnożenia "*" i dzielenia "\" na dwóch operandach w notacji infiksowej. Do wczytywania użyj funkcji *waitkey()*. Operandy powinny być typu *byte*, wynik typu *integer*. W przypadku przekroczenia zakresu operandów lub nieprawidłowych danych należy wyświetlić stosowny komunikat.
3. Zrealizuj kalkulator wykonujący operacje dodawania "+", odejmowania "-", mnożenia "*" i dzielenia "\" na dwóch operandach

w notacji infiksowej. Wykorzystując funkcję *input()* wczytaj całe wyrażenie, następnie dokonaj analizy poprawności danych. Załóż, że dane są reprezentowane poprzez bajt a wynik może być innym typem danych.