

# Mikrokontrolery 8-bitowe



# Czym jest mikrokontroler?

- Mikrokontroler ( C ) - to komputer wykonany w jednym układzie scalonym, używany do sterowania urządzeniami elektronicznymi. Układ, oprócz jednostki centralnej CPU, posiada zintegrowane, różnego rodzaju pamięci oraz układy wejścia wyjścia. (Wikipedia)

# Co w nich jest?

- jednostka obliczeniowa (ALU) - 8-bitowa, ale także 16 i 32-bitowa
- pamięć danych (RAM, EEPROM) i programu (EPROM, EEPROM, Flash)
- liczniki
- kontrolery przerwa

- kontrolery transmisji szeregowej i/lub równoległej (UART, SPI, I2C, USB, CAN, 1-Wire itp.)
- przetworniki analogowo-cyfrowe i/lub cyfrowo-analogowe
- zegar czasu rzeczywistego RTC

# Zastosowania:

- Urządzenia AGD
- Przemysł
- Motoryzacja tzw. ECU (Engine Control Unit)
- Automatyzacja pomiarów
- ...

# Popularne rodziny (C) :

- PIC
  - Microchip: PIC16F84
- Intel 8051
  - Intel: MCS-51
  - Atmel: 89C2051
- AVR
  - Atmel: Tiny AVR (ATiny13), Mega AVR (Atmega8)

# PICmicro (*Peripheral Interface Controller*)

- architektura harwardzka
- typ RISC
- dane przetwarzane w 8-bitowych porcjach ale rozmiar instrukcji może się różnić od rozmiaru jednego słowa danych (np.: dla w/w PIC16F84 długość instrukcji wynosi 14-bitów, a podstawowy rozmiar danych 8-bitów)

- kompilatory:
  - Asembler:
    - <http://gputils.sourceforge.net/> - GNU PIC Utilities (Linux, Windows)
  - ANSI C
    - <http://www.microchip.com> - kompilatory stworzone przez producenta mikrokontrolerów PIC (Linux, Windows) oraz środowisko programistyczne ( IDE ) tylko dla Windows.



- programatory
  - schematy + oprogramowanie:
    - <http://www.gnupic.org/>
    - <http://www.jdm.homepage.dk/newpics.htm>
  - komercyjne do kupienia na stronie producenta
    - [www.microchip.com](http://www.microchip.com) najtańszy zestaw: PICkit 2 Development Programmer ~ 80 EUR

# Intel 8051

- rozszerzona architektura harwardzka ( tzw. mieszana)
  - dane i rozkazy przechowywane są osobno ale przesyłane jedną magistral
- typ CISC
- podstawowy rozmiar słowa 8- bitów
- architektura popularna do początku lat 90 - tych

- kompilatory:
  - Asembler
    - <http://plit.de/asem-51/> - ASEM-51 freeware na licencji GPL dostępny dla Linux i Windows.
  - ANSI C
    - <http://sdcc.sourceforge.net/> - SDCC - Small Device C Compiler kompilator działający dla Linux i Windows

- programatory
  - komercyjny
    - <http://www.cyfronika.com.pl/wg/t51prog.html>
  - schemat + oprogramowanie
    - <http://chaokhun.kmitl.ac.th/~kswichit/easy1/easy.htm> - schemat programatora wraz z oprogramowaniem dla Windows
    - <http://www.pjrc.com/tech/8051/> - materia y dla ró nych systemów operacyjnych

# Konkretny przykład (AVR): Atmega8/8L

- Taktowanie
  - Atmega8L : 0 – 8 MHz
  - Atmega8 : 0 – 16 MHz
  - dla obu domy lnie 1 MHz
- Napi cia zasilania
  - 2.7 – 5.5 V Atmega8L
  - 4.5 – 5.5 V Atmega8

- Architektura RISC
  - 130 rozkazów wi kszo wykonywana w jednym cyklu zegara. Wszystkie rozkazy maj d ugo 16 lub 32 bitów
  - 32 x 8 -bitowych rejestrów ogólnego przeznaczenia
- Osobna pami danych i programu
  - Pami programu ( FLASH )
    - 8 KiB
      - je eli 2 bajty przypadaj na rozkaz to program nie mo e mie wi cej ni 4096 rozkazów.
    - 10,000 cykli zapisu.
  - Pami Danych
    - 1024 Bajty SRAM

## – Pamięć EEPROM

- 512B

- 100,000 cykli zapis

- Pamięć czysto wykorzystywana do przechowywania stałych danych i ciągłów tekstowych wypisywanych na wyświetlaczu LCD

## • Przerwania

- 19 wektorów przerwa np:

- TIMER1\_OVF – przerwanie „zegarowe”

- USART\_RXC – koniec odbioru danych poprzez interfejs USART

- USART\_TXC – koniec wysyłania danych poprzez interfejs USART

- Porty We/Wy

- 23 pin-y wykorzystywane jako linie we/wy

- zgrupowane w trzy porty:

- port A, B – 8 pinów

- port C – 7 pinów

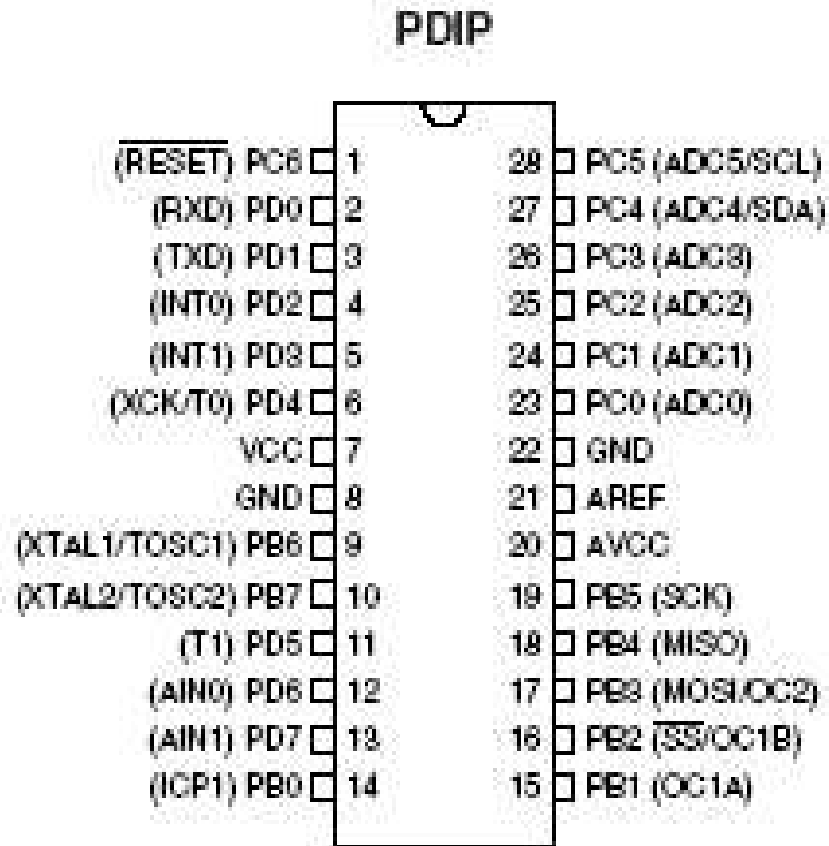
- niektóre z pinów mogą spełniać podwójną rolę

- „zwykłego” We/Wy

- We/Wy zaawansowanego interfejsu np. UART, SPI lub I2C jako sygnału zegara



- Konfiguracja pinów w Atmega8/8L ( PDIP )



- Dodatkowe funkcje

- Dwa liczniki 8- bitowe i jeden 16- bitowy ka dy generuj cy odpowiednie przerwanie

- Przyk adowo licznik Timer/Counter 0 mo e generowa przerwanie raz na 1, 8 , 64, 256 lub 1024 cykle zegara mo e tak e by taktowany zegarem zewn trznym pod czonym do pinu 6 ( T0 ).

- Watchdog timer – licznik okresowo resetuj cy C w celu zapobiegania zap tleniu si programu

- Boot-loader – który mo e za adowa program do pami ci flash z dowolnego ród a danych

## – interfejs USART

- ramki od 5 do 9 bitów z jednym lub dwoma bitami stopu
- sprawdzanie bitu parzystości
- wykrywanie błędów ramki
- dla domyślnego taktowania 1MHz maksymalna prędkość transmisji 115.2 kbps
- UWAGA:
  - W komputerach PC za interfejs USART mogą służyć porty COM jednak pracują one w standardzie RS-232 który korzysta z innych napięć niż TTL z jakiego korzysta Atmega. Do komunikacji przy pomocy interfejsu USART pomiędzy Atmega i PC-tem należy układ konwertujący sygnał z jednego standardu na drugi. Np: MAX 232 , DS275

- interfejsy SPI, TWI
- konwerter analogowo-cyfrowy
- komparator analogowy

# Programowanie AVR - ów:

- AVR libc ( <http://www.nongnu.org/avr-libc/> ) - zestaw darmowych narzędzi ( licencja GPL ) do pisania programów pod AVR -y w j zyku C.
- WinAVR ( <http://winavr.sourceforge.net/> ) - odpowiednik dla systemów MS Windows.

- Zawarto biblioteki AVR libc:
  - `<avr/io.h>` - definicje rejestrów specjalnego przeznaczenia, oraz przydatne makra typu:
    - `loop_until_bit_is_set(sfr,bit)`
    - `bit_is_clear(sfr,bit)`
  - `<avr/eeprom.h>` - funkcje do manipulowania pamięcią EEPROM
  - `<avr/interrupt.h>` - obsługa przerwa (przerwania z reguły nie są wywoływane)
  - odpowiedniki nagłówków ze „zwykłego” C:
    - `<math.h>`
    - `<stdint.h>` `<inttypes.h>`
    - `<stdio.h>`

- <util/delay.h> - funkcje „czekaj ce”
  - nie wy czaj przerwa !
- <util/crc.h> - funkcje obliczaj ce CRC 8 oraz CRC16

# Przykładowy kod zapalający diodę na 0.1 sekundy:

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/delay.h>

int main()
{
    /* Na początku konfiguruj wykorzystywaną „nórkę” układu. */
    /* Pin 1 czyli PD0 konfiguruj jako wyjściowy. */
    DDRD |= 1 << PD0;

    PORTD |= 1 << PD0; /* Zapalam diodę. */
    _delay_loop_2( 25000 ); /* Czekam 0.1 sekundy */
    PORTD &= ~( 1 << PD0 ); /* I ja gaszę */
}
```



# Jak za adowa program do AVR ?

- hardware:
  - popularne sposoby po czenia AVR z portem równoleg ym drukarki:

stk200

AVR	LPT
Reset (1)	Init (16)
MOSI (17)	D0 (2)
MISO (18)	Busy (11)
SCK (19)	Strobe (1)
GND	GND (18)

bsd

AVR	LPT
Reset (1)	PIN 7
MOSI (17)	PIN 9
MISO (18)	PIN10
SCK (19)	PIN 8
GND	GND

- software:
  - <http://www.nongnu.org/uisp/> - UISP - AVR In-System Programmer – programator działający zarówno w systemie Windows jak i Linux. Ładujący program do pamięci Flash mikrokontrolera. Program aby został załadowany musi być w formacie Intel HEX.

# Inne sposoby programowania AVR:

- [www.avr-asm-tutorial.net/gavrasm/](http://www.avr-asm-tutorial.net/gavrasm/) - assembler pod Linux i Windows, wraz ze szczegółowym samouczkiem.
- BASCOM AVR – komercyjny język programowania zbliżony w składni do BASIC. Posiada gotowe biblioteki do obsługi niezaimplementowanych sprzętowych protokołów. Działa tylko na systemach operacyjnych z rodziny Windows.

## ród a wiedzy:

- strony producentów mikrokontrolerów:

- <http://www.atmel.com>
- <http://www.microchip.com>
- <http://www.intel.com>

- inne strony:

- [http://www.societyofrobots.com/microcontroller\\_tutorial.shtml](http://www.societyofrobots.com/microcontroller_tutorial.shtml) - wprowadzenie do tematu mikrokontrolerów
- <http://www.belleresearch.com/beginnerpic.htm> - podręcznik dla rozpoczynających zabaw z mikrokontrolerem PIC
- <http://www.8052.com/tutorial.phtml> - bardzo szczegółowe samouczki dotyczące używania mikrokontrolerów Intel 8051
- <http://www.fuw.edu.pl/~pablo/s/index.html> - strona dr Pawła Klimczewskiego z materiałami na temat programowania Atmega8
- <http://www.elektroda.pl> - internetowe forum poświęcone elektronice
- <http://www.avrfreaks.com> - portal poświęcony w całości korzystaniu z AVR