



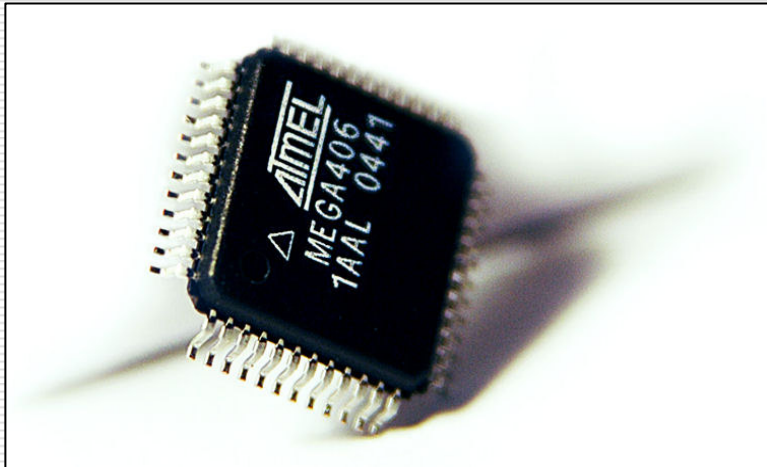
*Studenckie Koło Naukowe
Mikroprocesorowych Systemów Autonomicznych „MIPSA”
przy Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki
Politechniki Łódzkiej*

Inkubator AVR

Podstawy obsługi i programowania
mikrokontrolerów rodziny **AVR**.

CZĘŚĆ I. Wprowadzenie i hardware

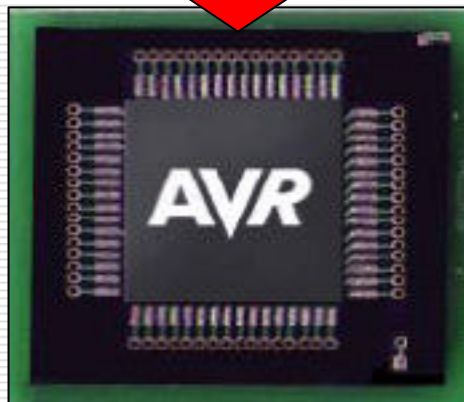
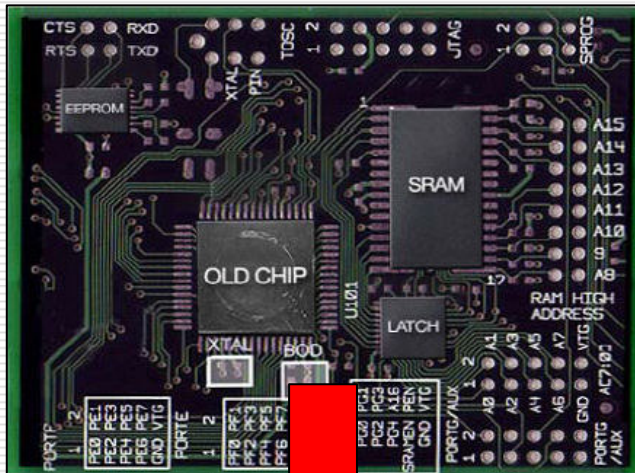
Co na temat AVR każdy wiedzieć powinien?



- ❑ Producent: ATMEL (www.atmel.com)

- ❑ Instrukcje wykonywane w jednym cyklu zegarowym (większość),
- ❑ Architektura RISC, bogaty zestaw instrukcji i 32 rejestry ogólnego przeznaczenia,
- ❑ Bardzo szybkie i wygodne programowanie w systemie (ISP),
- ❑ Praca w dużym zakresie napięć 1.8V – 5.5V, tryby niskiego poboru mocy.

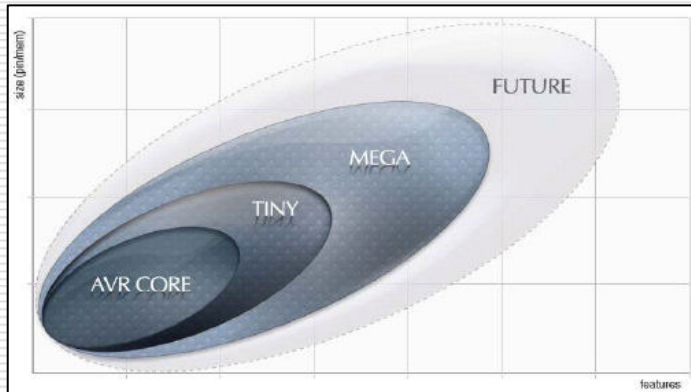
Dlaczego AVR?



- Kilka powodów:
 - wysoka wydajność,
 - niski pobór mocy,
 - maks. 20MIPS przy 20MHz,
 - działanie w zakresie: 1.8 – 5.5 V,
 - architektura nastawiona na języki wysokiego poziomu (język C),
 - 32 rejestry ogólnego przeznaczenia,
 - wbudowana mnożarka,

źródło: AVR Product Line Introduction

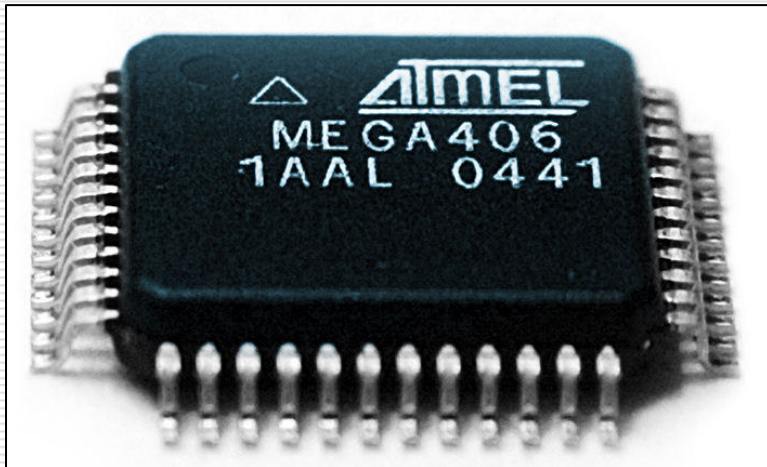
Dlaczego AVR?



- skalowalność (urządzenia od 1 do 256 kB, od 8 do 100 wyprowadzeń, pełna kompatybilność kodu, kompatybilność pin-to-pin z innymi urządzeniami),
- programowanie, debuggowanie i weryfikacja w systemie,
- wbudowana pamięć Flash.

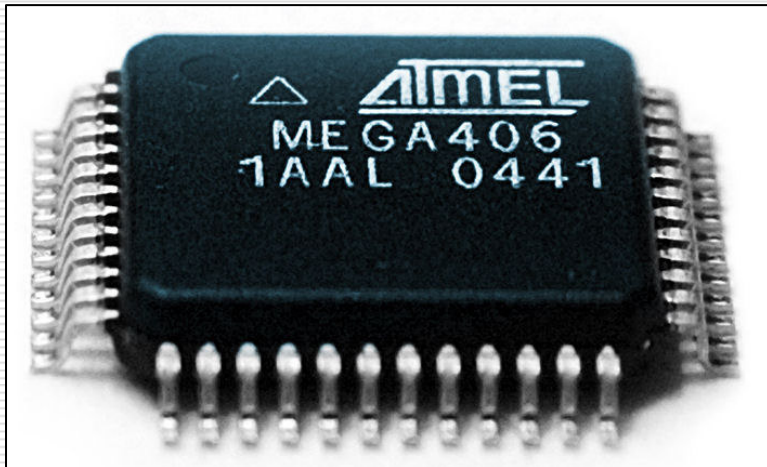
źródło: AVR Product Line Introduction

Rodziny mikrokontrolerów AVR



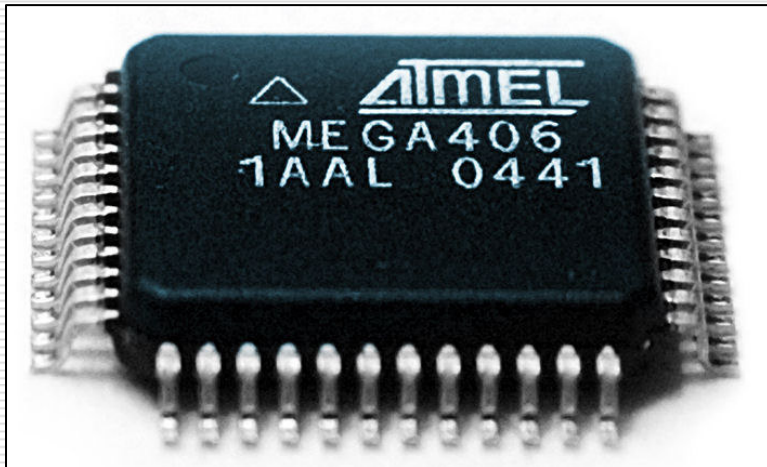
- tinyAVR™ – mikrokontrolery ogólnego przeznaczenia (max 4kB Flash, 126B SRAM i EEPROM),
- megaAVR™ – możliwość samoprzeprogramowania pamięci (max 256kB Flash, 4kB SRAM i EEPROM),
- LCD AVR – zintegrowane drivery LCD, pobór mocy ($32 \text{ kHz} < 20\mu\text{A}$),
- CAN AVR – zintegrowane kontrolery magistrali CAN.

Rodziny mikrokontrolerów AVR



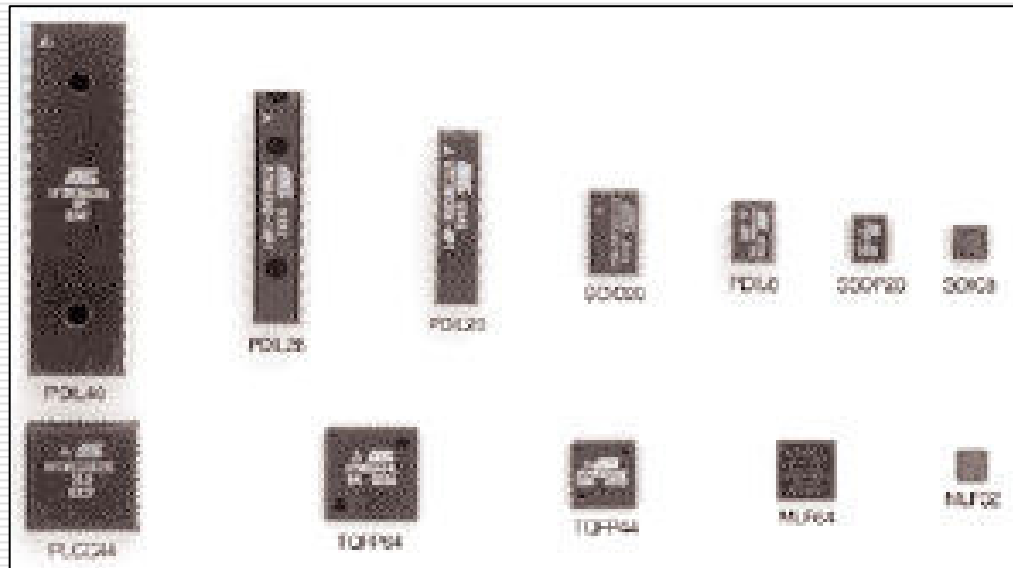
- ❑ Automotive AVR – mikrokontrolery dla automatyki,
- ❑ USB AVR – zintegrowane kontrolery USB,
- ❑ Lighting AVR (Motor Drivers) – zaawansowane tryby PWM,
- ❑ Smart Battery AVR – możliwość liczenia ładunków,
- ❑ FPGA AVR – rdzeń AVR obudowany w matryce FPGA (*Field Programmable Gate Array*), pozwala uniknąć układów otaczających tzw. *Glue Logic*, budować w FPGA własne struktury.

Rodziny mikrokontrolerów AVR



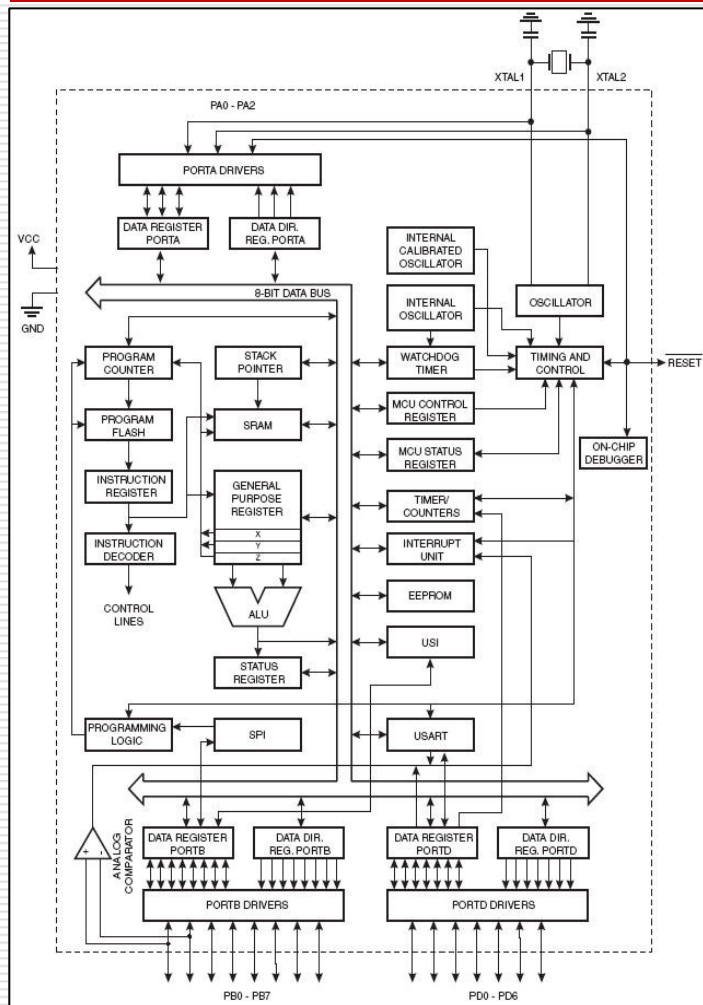
- PicoPower Technology – firma Atmel stopniowo wprowadza 16 urządzeń oznaczonych literą „P” (np.: ATmega169P):
 - Zasilanie: 1.8V
 - Zminimalizowane prądy upływu,
 - Niskonapięciowy oscylator 32kHz,
 - Możliwość wyłączenia wejść cyfrowych,
 - Bramkowanie zegara.

Dostępne wersje obudów



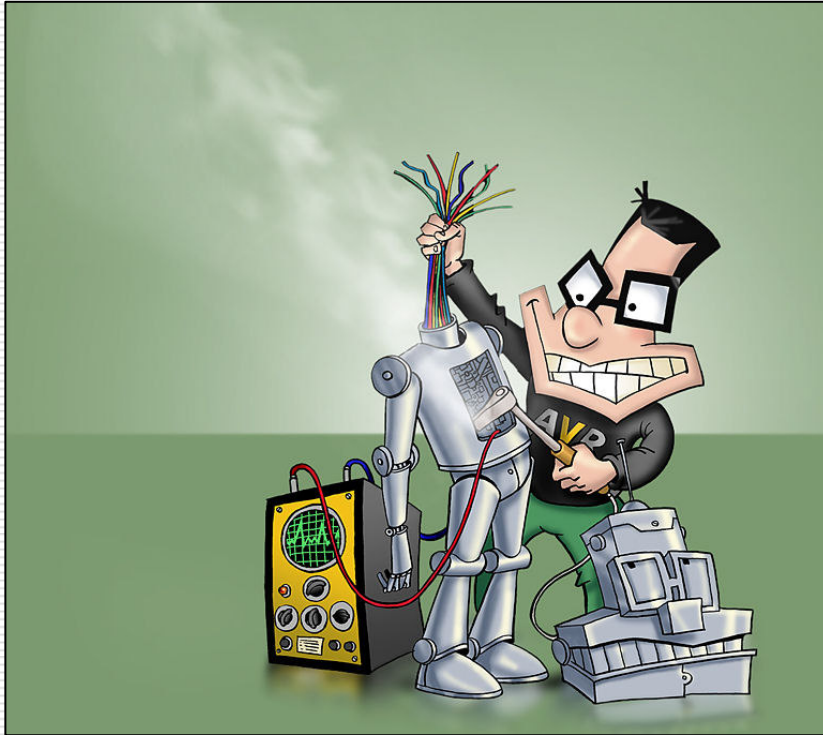
- ❑ Wiele możliwych wersji obudów – od PDIP, przez TQFP do takich jak PLCC, SOIC i MLF

A jak to wygląda?



- ❑ Rejestry ogólnego przeznaczenia,
- ❑ Rejestr stanu,
- ❑ ALU (Arithmetic Logic Unit),
- ❑ Pamięci Flash (programu), SRAM, EEPROM,
- ❑ Liczniki (+ Watchdog),
- ❑ USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter),
- ❑ USI (Universal Serial Interface),
- ❑ System przerwań,
- ❑ Komparator analogowy,
- ❑ Porty wejścia / wyjścia,
- ❑ I inne :-)

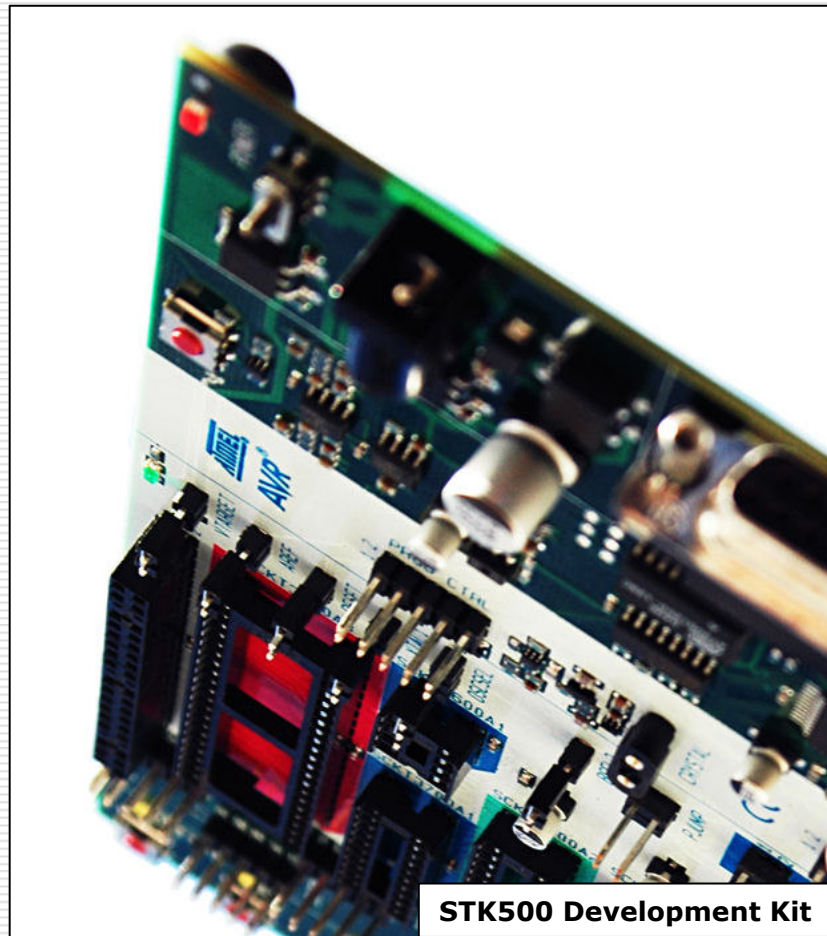
Jak zacząć (dobrą) zabawę?



- Czego potrzebujemy?
 - Mikrokontrolera,
 - Płytki „testowej”,
 - Programatora,
 - Kompilatora (AVR assembler, C++, Basic, Pascal, Ada...),
 - Oprogramowania bezpośrednio ładującego FLASH,
 - Ciekawych pomysłów...

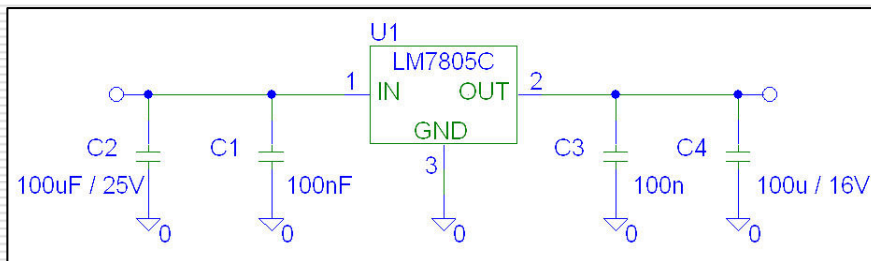
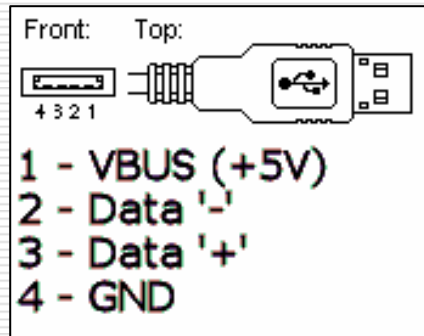
źródło: AVR Product Line Introduction

1. Płyta „testowa”



- Co musi się tam znaleźć koniecznie:
 - Podstawka pod uC,
 - Obwód zasilania,
 - Oscylator,
 - Układ resetu (+ np. przycisk resetu),
 - Gniazdo dla programatora,
- Co jest opcjonalne:
 - Dowolne „kontrolki”,
 - Kolejne przyciski (klawiatura),
 - Wyświetlacz LCD, 7-segmentowy,
 - Konwerter poziomów logicznych CMOS na RS232,
- Tzw. „bajery”:
 - Układ RTC, układy 1-Wire, przetworniki A/C, C/A i wiele innych...

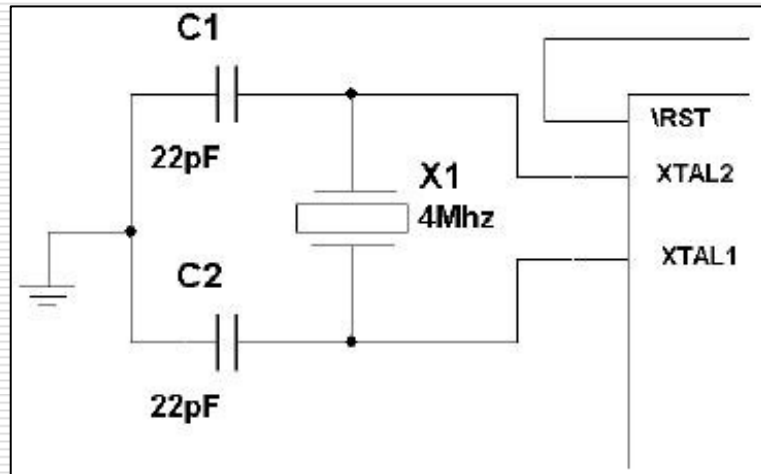
1a. Obwód zasilania



□ Są dwa podejścia do sprawy:

1. Zakładamy, że jesteśmy nieomylni i dysponujemy PC'tem (zasilanie np. z USB),
2. Znamy siebie na tyle, że założymy stabilizator i proste zabezpieczenie.
3. Rozwiązanie pośrednie (odpowiednie złącze).

1b. Oscylator (kwarc)



□ Istnieje wiele źródeł sygnału zegara:

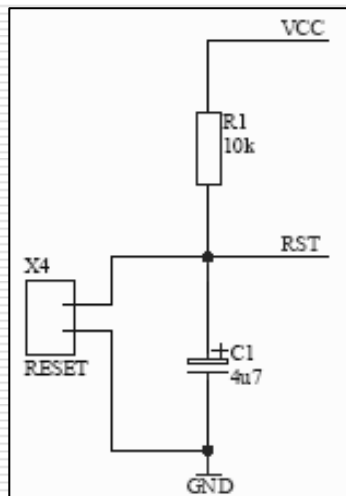
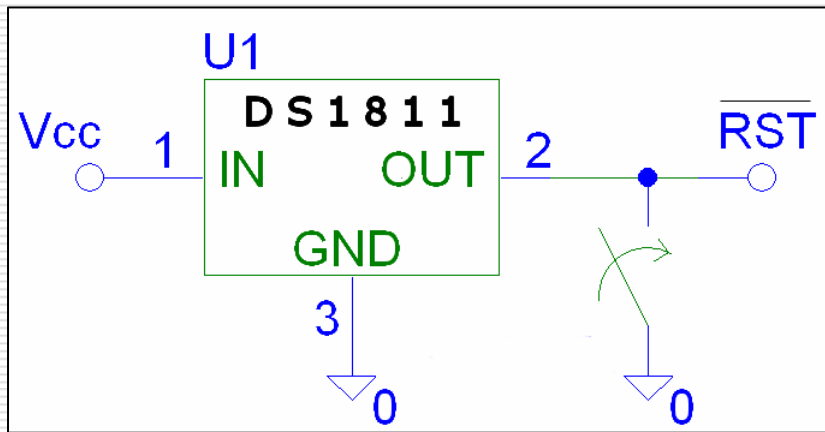
- Zewnętrzne układy zegara o poziomach TTL,
- Rezonatory kwarcowe,
- Rezonatory ceramiczne,
- Oscylator wewnętrzny.



Device Clocking Option	CKSEL3..0
External Clock	0000
Calibrated Internal RC Oscillator 4MHz	0010
Calibrated internal RC Oscillator 8MHz	0100
Watchdog Oscillator 128kHz	0110
External Crystal/Ceramic Resonator	1000 - 1111
Reserved	0001/0011/0101/0111

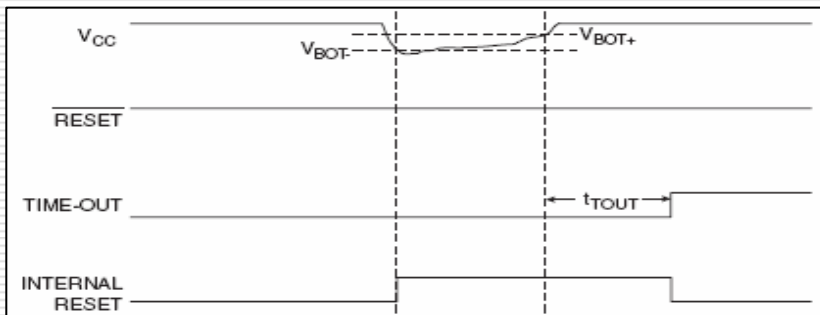
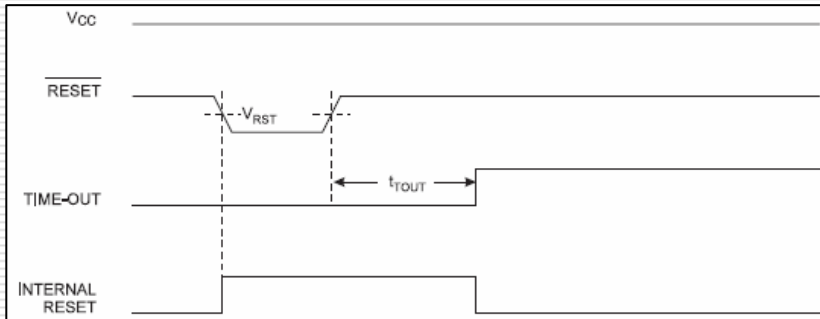
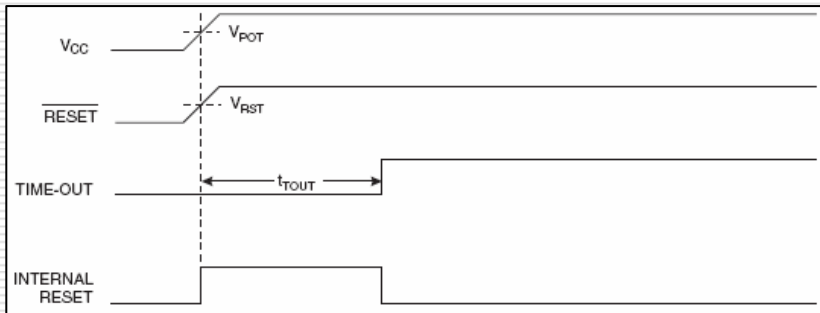
Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

1c. Układ generowania RESET'u

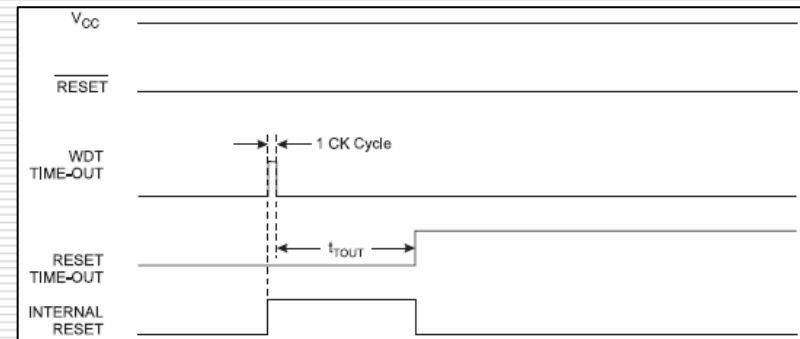


- ❑ Specjalizowane układy Brown-out-Reset (np. DS1811, DS1813),
- ❑ Zabezpieczenie kondensatorem przed spadkami napięcia,
- ❑ Wewnętrzny układ resetu.

1c. Źródła sygnału RESET



- Źródła RESET'u:
 - Power-on Reset,
 - External Reset,
 - Watchdog Reset,
 - Brown-out Reset



1d. Gniazdo dla programatora



Rys. na bazie materiałów firmy Atmel

- Sposoby programowania:
 - Równoległe,
 - Szeregowe (SPI – Serial Programming Interface),
 - JTAG.

- Potrzebne sygnały:
 - MOSI,
 - MISO,
 - SCK,
 - RESET,
 - GND.

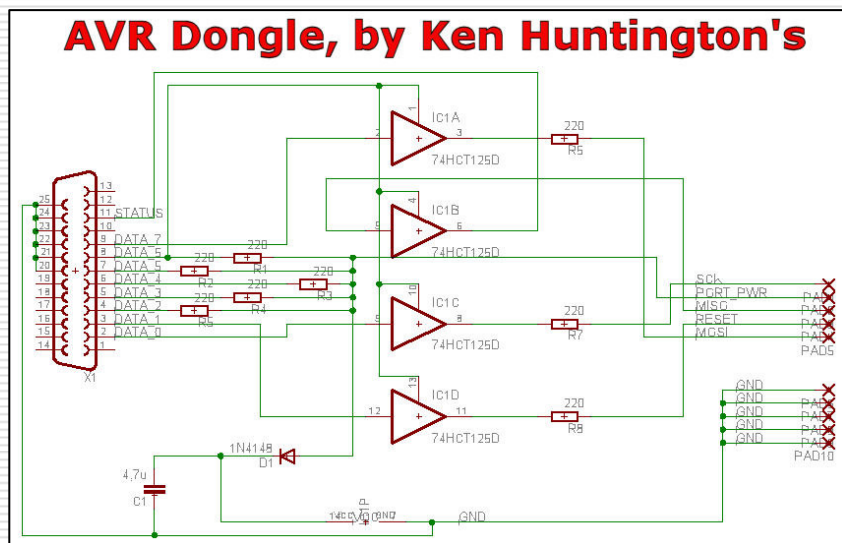
- PROGRAMATOR...

2. PROGRAMATOR – SPI



	1	2	
MISO	○	○	VTG
SCK	○	○	MOSI
RST	○	○	GND
ISP6PIN			

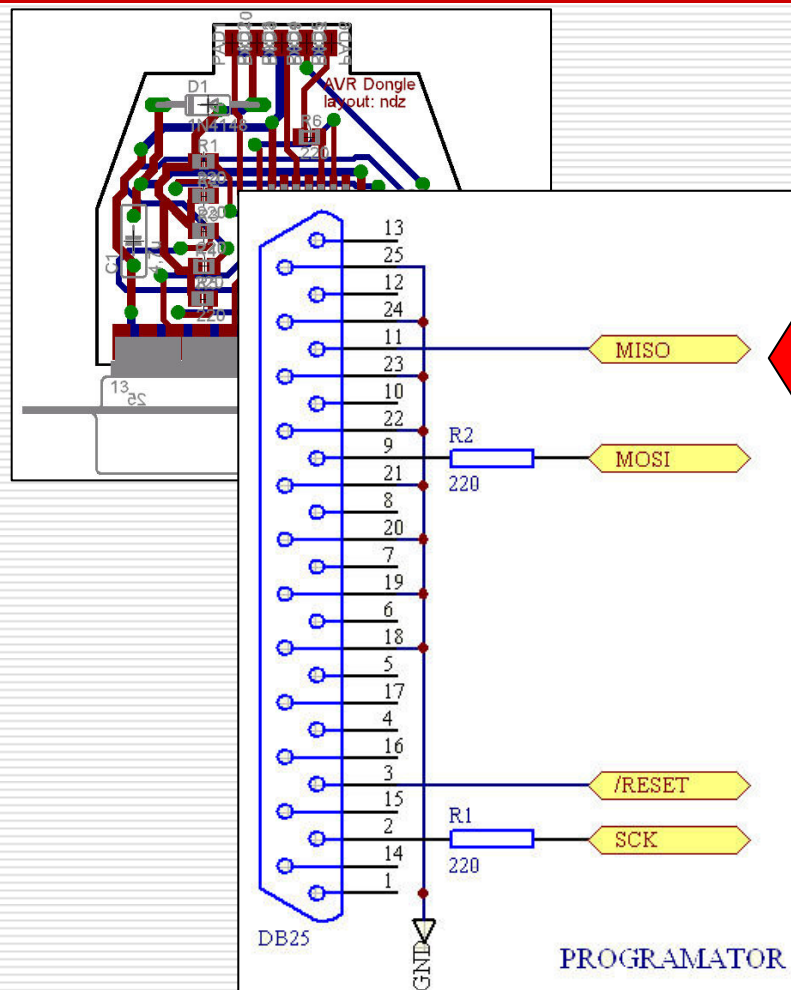
- PROGRAMATOR:
 - Wersja 1:
buforowana, tzw. „elegancka”,
 - Wersja 2: nie
buforowana, tzw. „skuteczna”.



- Popularne gniazdo do programowania.

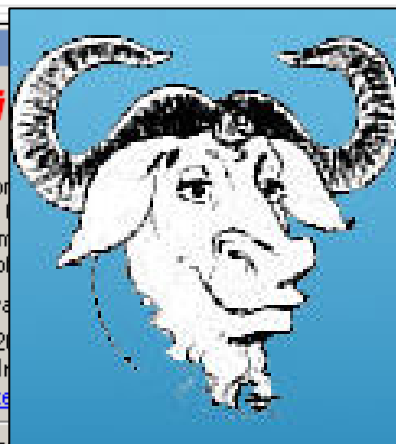
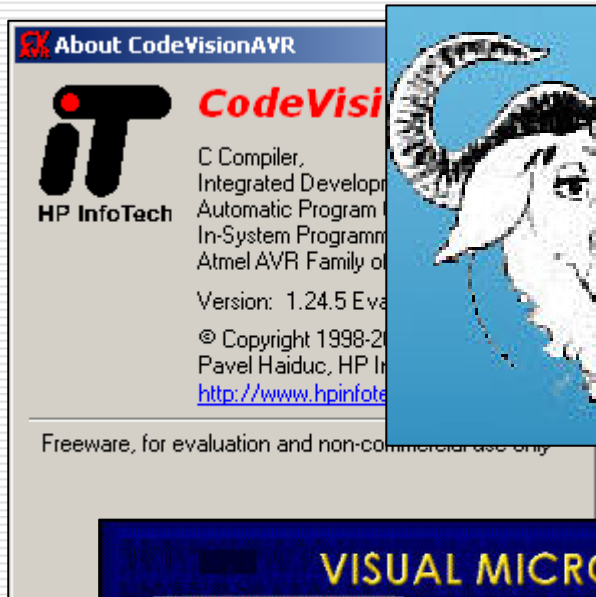
źródło: sp12 Documentation oraz <http://www.avr-asm-tutorial.net>

2. PROGRAMATOR – SPI

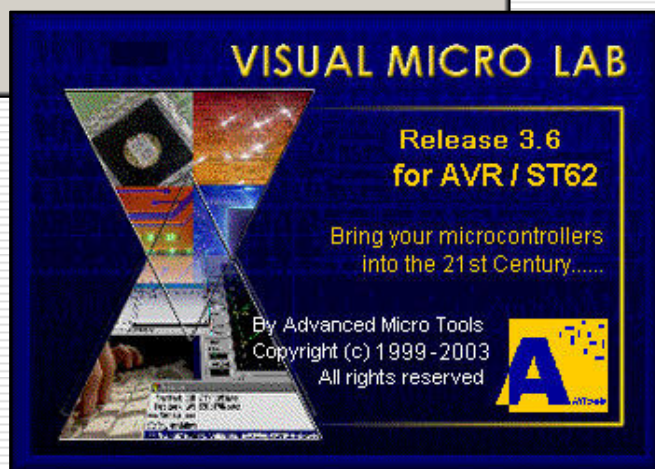


- Najprostsza wersja programatora – niezawodna ale nie buforowana!!!
- Całość zamknąć można we wtyczce DB25 (złącze równoległe LPT).

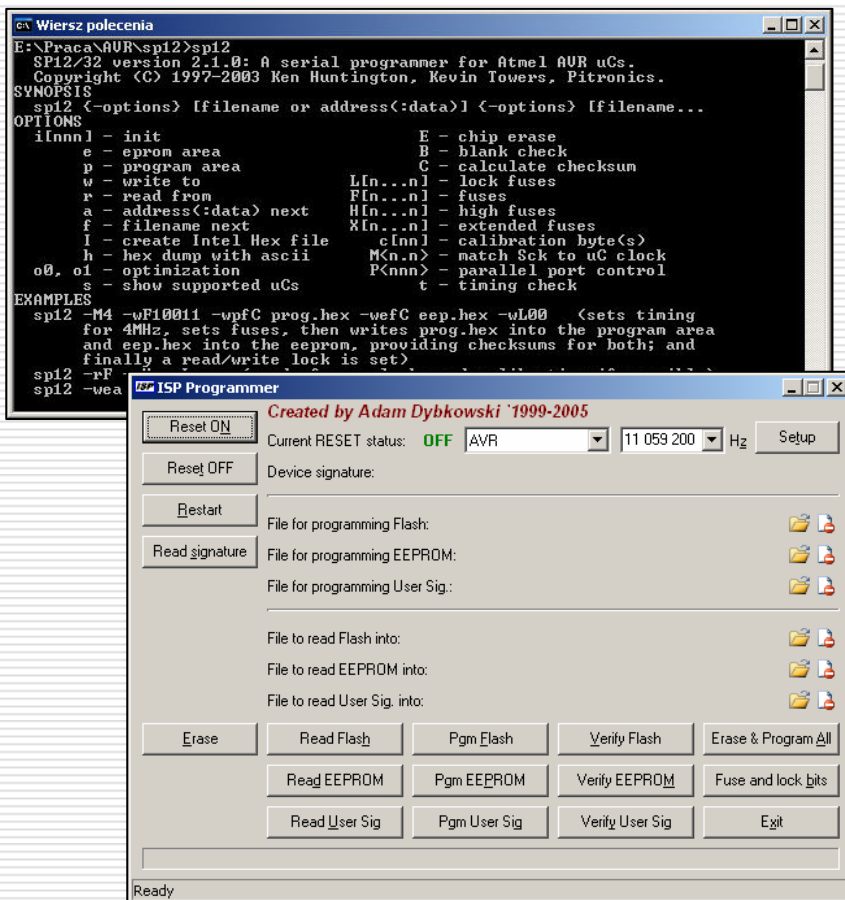
3. Środowiska do programowania C



- AVR-GCC (Open Source)
- WinAVR (korzysta z GCC)
- AvrSide (korzysta z GCC)
- Visual Micro Lab 3.6
(komercyjny, AVR Asembler lub może korzystać z GCC)
- Atmel AVR Studio 4 (AVR Asembler lub może korzystać z GCC)
- CodeVisionAVR 1.24 (pakiet komercyjny, własny kompilator języka C)
- CrossWorks AVR (pakiet komercyjny, własny kompilator języka C)
- Wiele innych...

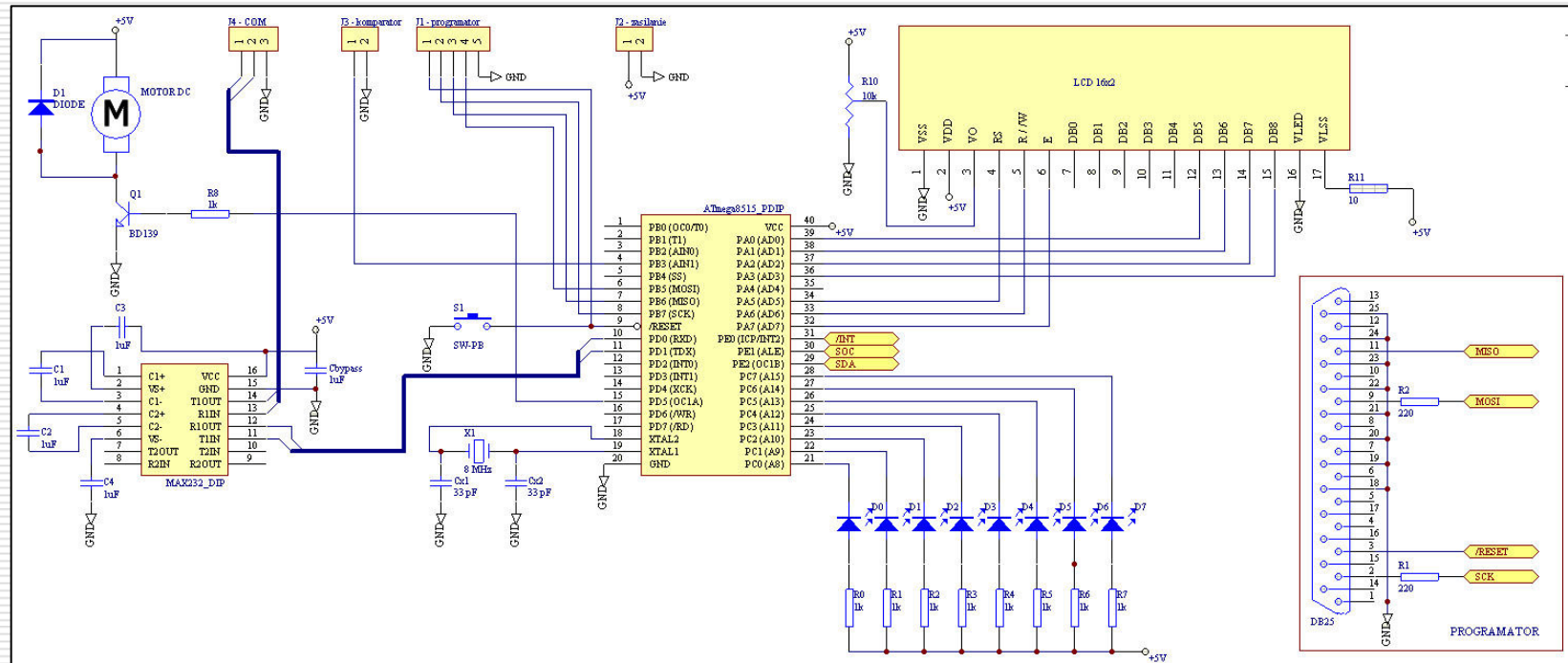


4. Ładowanie pamięci FLASH uC



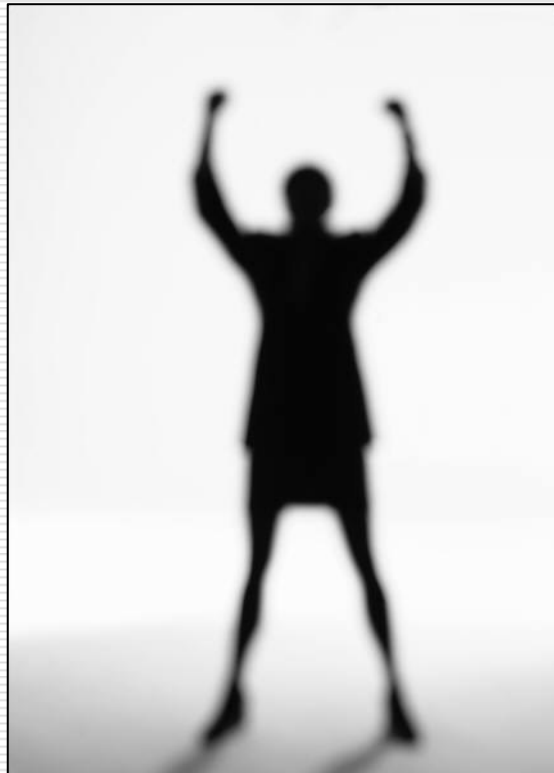
- Sp12,
- PonyProg 2000,
- AVR Studio 4,
- AVR Dude,
- Wiele innych...

Przykładowa płytka „testowa”



- ❑ Gniazda zasilania, programatora, komparatora analogowego,
- ❑ Układ MAX232,
- ❑ Sterowanie LCD 16x2,
- ❑ Sterowanie silnikiem poprzez modulację PWM.

Jak rozpocząć, czyli pierwszy program...



```
#define __AVR_AT90S8515__
#include <io.h>
#include <interrupt.h>
#include <signal.h>

#define LED1_ON (PORTC|=(0x04))
#define LED2_ON (PORTC|=(0x08))
#define LED1_OFF (PORTC&=~(0x04))
#define LED2_OFF (PORTC&=~(0x08))
#define SWITCH1 (PINC&(0x01))
#define SWITCH2 (PINC&(0x02))

int main(void)
{
    DDRC = 0xFC;
    PORTC = 0x03;
    for (;;)
    {
        if (SWITCH1==0) LED1_OFF; else LED1_ON;
        if (SWITCH2==0) LED2_OFF; else LED2_ON;
    }
}
```

Przydatne linki...

PODSTAWOWE:

- ❑ <http://www.atmel.com> – strona producenta AVR,
- ❑ <http://www.avrfreaks.com> – najlepsze forum mikrokontrolerów AVR,

OPROGRAMOWANIE:

- ❑ <http://winavr.sourceforge.net> – strona programu WinAVR,
- ❑ <http://www.nongnu.org/avr-libc/> - strona projektu avr-libc,
- ❑ <http://gcc.gnu.org> – strona kompilatora gcc,
- ❑ <http://sources.redhat.com/binutils/> - strona narzędzi binutils,
- ❑ <http://avrside.fr.pl> – strona środowiska AvrSide.

INNE:

- ❑ <http://www.propox.com> – strona polskiego producenta zestawów ewaluacyjnych,
- ❑ <http://www.btc.pl> – strona wydawnictwa BTC.

Ruszamy w drogę...



□ Dodatkowe informacje:

- www.atmel.com
 - www.avrfreaks.com
-
- www.mikrocontroller.com
 - www.mikrokontrolery.of.pl
 - www.avr-asm-tutorial.net
-
- mipsa.p.lodz.pl