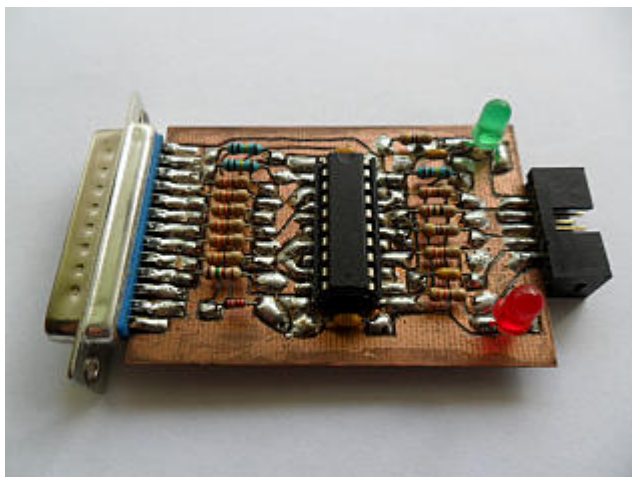


AVRprog 1.01

© 2014 by RomanWorkshop

<http://romanworkshop.blutu.pl/>

Programator mikrokontrolerów AVR



Jest to prosty programator (ulepszony STK200/300) 8-bitowych mikrokontrolerów AVR (ATtiny/ATmega) firmy Atmel, podłączany do portu równoległego (LPT) komputera PC. Służy do programowania mikrokontrolerów w systemie (ISP - In-System Programming), z którego musi być zasilany napięciem stabilizowanym od 3 do 5V. Jest całkowicie odporny na odwrotne podłączenie polaryzacji napięcia zasilającego. Posiada dwie diody LED sygnalizujące włączenie zasilania (zielona) oraz transmisję danych (czerwona). Sygnały programujące są wyprowadzone na standardowe 10-pinowe złącze ISP i powinny być połączone z systemem docelowym przewodami, o długości ok. 30 cm. Wszystkie przewody GND powinny być połączone z

masą programatora i systemu docelowego, aby zapewnić ekranowanie sygnałów programujących. Układ zmontowałem na płytce dwustronnej (16 przelotek) o wymiarach 65x45 mm, na której elementy są montowane powierzchniowo (na górnej warstwie). Jako małą ciekawostkę dodam, że płytkę prototypową wykonałem bez trawienia (mini wiertarką z małym frezem).

Programator współpracuje z następującymi programami sterującymi: ISP Programmer 1.2.0.56, PonyProg2000 2.08c, Bascom AVR 2.0.7.1 oraz AVR-ISP 5.6.0.1.

W razie problemów, należy zmienić w BIOS'ie tryb pracy portu równoległego na "Bi-Directional". System Windows XP może sprawdzać co jakiś czas, czy do portu równoległego nie podłączono drukarki. Sprawdzanie to może powodować nieprawidłową pracę programatora, podłączonego do tego portu. Aby wyłączyć ciągłe sprawdzanie portu równoległego, należy zmodyfikować następującą wartość w rejestrze systemowym:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parport\Parameters]
"DisableWarmPoll"=dword:00000001
```

Zmiana ta wymaga ponownego uruchomienia komputera, ale dzięki niej Windows XP sprawdzi obecność drukarki tylko raz, podczas startu systemu (nie wykryje automatycznie drukarki, podłączonej do portu LPT już po załadowaniu systemu). Aby włączyć ciągłe sprawdzanie portu LPT, należy temu kluczowi ustawić wartość 0 i zrestartować komputer.

Konfiguracja programu ISP Programmer 1.2.0.56

Po instalacji uruchamiamy program i naciskamy przycisk "Setup". W nowym oknie "ISP Programmer Setup", z rozwijalnej listy "ISP cable pinout" wybieramy pozycję "STK200/300 Evaluation Board", z listy "Port with ISP programmer" wybieramy numer portu równoległego, do którego podłączony jest programator, a następnie naciskamy przycisk "OK". Trzeba upewnić się, że pole "Autodetect" jest zaznaczone, a z rozwijalnej listy obok jest wybrana pozycja "AVR". Druga rozwijalna lista "Hz", służy do ustawiania szybkości transmisji danych i w razie błędów programowania, należy wybrać mniejszą wartość. Po podłączeniu programatora do systemu docelowego można nacisnąć przycisk "Read signature", co spowoduje próbę rozpoznania mikrokontrolera przez odczytanie jego sygnatury. Jeśli uda się rozpoznać mikrokontroler, to po napisie "Device signature" pojawią się informacje o nim (model, pojemność pamięci FLASH i EEPROM). Wtedy uaktywnią się też przyciski, umożliwiające wykonywanie operacji takich, jak: odczyt/zapis/weryfikacja, czy kasowanie pamięci mikrokontrolera. Przed odczytem/zapisem pamięci należy wskazać pliki, w których zostaną zapisane dane odczytane z pamięci FLASH/EEPROM oraz pliki, które zawierają dane do zaprogramowania w pamięci FLASH/EEPROM mikrokontrolera. Po naciśnięciu przycisku "Fuse and lock bits", pojawi się nowe okno służące do odczytywania/modyfikowania ustawień konfiguracyjnych mikrokontrolera (tzw. fuse bitów).

Konfiguracja programu PonyProg2000 2.08c

Po instalacji uruchamiamy program i z menu "Setup" wybieramy pozycję "Interface Setup...". W nowym oknie "I/O port setup" zaznaczamy pole "Parallel", z rozwijalnej listy wybieramy pozycję "Avr ISP I/O", zaznaczamy pole z numerem portu równoległego, do którego podłączony jest programator i naciskamy przycisk "OK". Następnie z menu "Setup" wybieramy pozycję "Calibration". W nowym oknie "Yes or No" po upewnieniu się, że żaden program nie obciąża procesora, ani nie używa twardego dysku, naciskamy przycisk "Yes". Po kilku sekundach pojawi się okno "Notice" z informacją o zakończeniu kalibracji, które zamykamy naciskając przycisk "OK". Przed rozpoczęciem programowania, należy dokładnie wybrać model mikrokontrolera za pomocą rozwijalnych list lub z menu "Device/AVR micro". Następnie można już wykonywać operacje takie, jak: odczyt/zapis/weryfikacja, czy kasowanie pamięci mikrokontrolera. Większą część okna programu zajmuje przewijalne pole (bufor), które zawiera dane odczytane z pamięci FLASH i EEPROM mikrokontrolera (różnią się kolorami) i jednocześnie dane do zaprogramowania. Po zaznaczeniu w menu "Edit" pozycji "Edit Buffer enabled", wartość każdej komórki pamięci w buforze można dowolnie modyfikować. Do buforów można również wczytać dane z plików w różnych formatach (w tym HEX). Po wybraniu z menu "Command" pozycji "Security and Configuration Bits...", pojawi się nowe okno służące do odczytywania/modyfikowania ustawień konfiguracyjnych mikrokontrolera (tzw. fuse bitów). Przy próbie odczytania/zapisania mojego mikrokontrolera ATmega168P, pojawił się komunikat o błędzie "Device missing or unknown device (-24)". Po naciśnięciu przycisku "Ignore" odczyt/zapis przebiegał prawidłowo. Jest to spowodowane tylko tym, że program nie rozpoznaje sygnatury zwracanej przez kilka modeli mikrokontrolerów.

Konfiguracja programu Bascom AVR 2.0.7.1

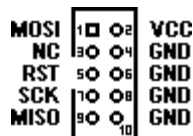
Po instalacji uruchamiamy program i z menu "Options" wybieramy pozycję "Programmer". W nowym oknie "BASCOS-AVR Options" z rozwijalnej listy "Programmer" wybieramy pozycję "STK200/STK300 Programmer", w grupie "Parallel" z listy "LPT-address" wybieramy adres portu równoległego, do którego podłączony jest programator i naciskamy przycisk "OK". Następnie z menu "File" wybieramy pozycję "New" i naciskamy klawisz [F4]. Pojawia się dwa okna informacyjne, które zamykamy naciskając przycisk "OK". W nowym oknie "AVR ISP STK programmer" z rozwijalnej listy "Chip" wybieramy model mikrokontrolera. Po podłączeniu programatora do systemu docelowego można też z menu "Chip" wybrać pozycję "Identify", co spowoduje próbę rozpoznania mikrokontrolera przez odczytanie jego sygnatury. Jeśli uda się rozpoznać mikrokontroler, to z listy automatycznie zostanie wybrany odpowiedni model. Następnie można już wykonywać operacje takie, jak: odczyt/zapis/weryfikacja, sprawdzenie czystości, czy kasowanie pamięci mikrokontrolera. W zakładkach "FlashROM" i "EEPROM", znajdują się przewijalne pola edycyjne (bufory), które zawierają dane odczytane z pamięci FLASH/EEPROM mikrokontrolera i jednocześnie dane do zaprogramowania. Wartość każdej komórki pamięci w buforze można dowolnie modyfikować. Do buforów można również wczytać dane z plików (HEX). Zakładka "Lock and Fuse Bits" służy do odczytywania/modyfikowania ustawień konfiguracyjnych mikrokontrolera (tzw. fuse bitów).

Konfiguracja programu AVR-ISP 5.6.0.1

Po instalacji uruchamiamy program i naciskamy przycisk "Setup". W nowym oknie "ISP Configuration" z rozwijalnej listy "Communications Device", wybieramy numer portu równoległego, do którego podłączony jest programator. Rozwijalna lista "Programmer ISP Speed" służy do ustawiania szybkości transmisji danych i w razie błędów programowania, należy wybrać z niej pozycję "Medium" lub "Slow". Okno zamykamy naciskając przycisk "OK". Po podłączeniu programatora do systemu docelowego z rozwijalnej listy "Device" wybieramy model mikrokontrolera, co spowoduje próbę odczytania jego sygnatury. Jeśli uda się rozpoznać mikrokontroler, to na pasku stanu pojawi się napis "ISP Initialized; Detected Device", a po nim model wykrytego mikrokontrolera. Niestety rozpoznanie mikrokontrolera, nie powoduje w programie automatycznej zmiany modelu wybranego na liście "Device", co w przypadku wybrania złego modelu, wpływa na rozmiary buforów pamięci FLASH/EEPROM, a tym samym powoduje zbędne odczytywanie/zapisywanie nieistniejących obszarów pamięci mikrokontrolera. Dlatego z listy "Device", należy zawsze wybrać taki sam model mikrokontrolera, jaki został wykryty przez program. Następnie można już wykonywać operacje takie, jak: odczyt/zapis/weryfikacja, czy kasowanie pamięci mikrokontrolera. W zakładkach "Flash Memory" i "EEPROM Memory", znajdują się przewijalne pola edycyjne (bufory), które zawierają dane odczy-

tane z pamięci FLASH/EEPROM mikrokontrolera i jednocześnie dane do zaprogramowania. Wartość każdej komórki pamięci w buforze można dowolnie modyfikować. Do buforów można również wczytać dane z plików (HEX). Zakładka "Fuses & Lock Bits" służy do odczytywania/modyfikowania ustawień konfiguracyjnych mikrokontrolera (tzw. fuse bitów).

Sygnały na złączu ISP:



Spis elementów:

REZYSTORY:

R1-R9 - 330 Ω
R10 - 150 Ω
R11-R13 - 100 k Ω
R14 - 10 Ω
R15, R16 - 470 Ω

KONDENSATORY:

C1-C3 - 100 pF
C4 - 100 nF

DIODY:

D1 - 1N4148
LED1 - zielona 5 mm
LED2 - czerwona 5 mm

UKŁADY SCALONE:

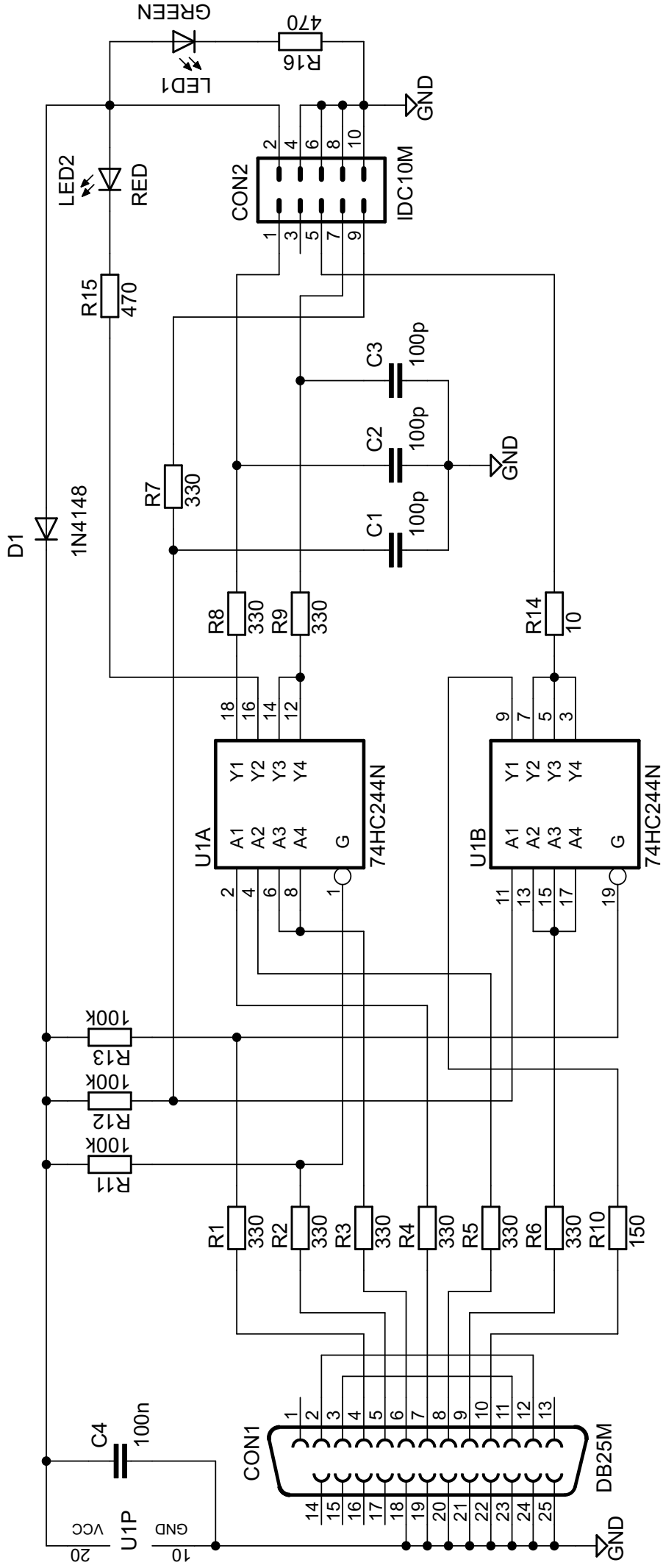
U1 - 74HC244

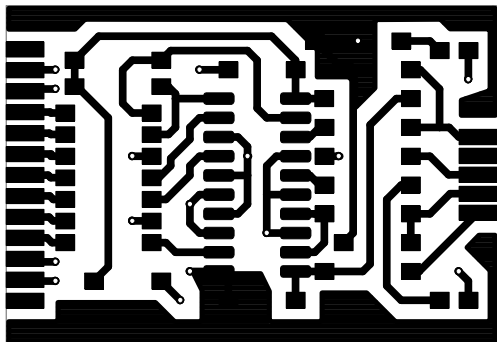
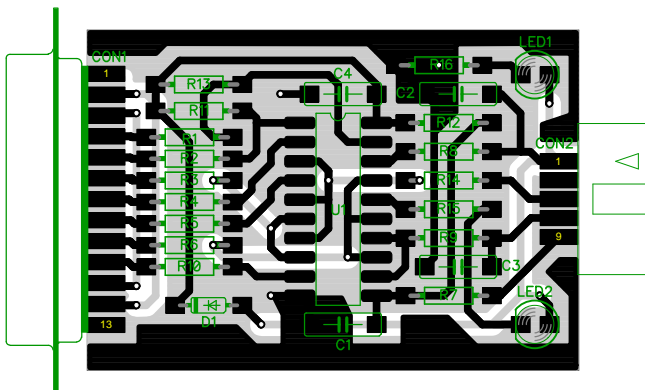
ZŁĄCZA:

CON1 - DB25M proste (męskie)
CON2 - IDC10M proste (męskie)

PODSTAWKI ZWYKŁE:

U1 - DIP20





Wykaz mikrokontrolerów AVR w obudowach DIP:

Model <pamięć FLASH/EEPROM/SRAM, sygnatura> {obudowa} [zegar]

DIP8:

AT90(S/LS)2323 <2k/128/128, \$1E9102*> {U1} [RES/EXT]
AT90(S/LS)2343 <2k/128/128, \$1E9103*> {U1} [INT/EXT]
ATtiny12 <1k/64/-, \$1E9005*> {U1} [RES/INT/EXT]
ATtiny13(A/V) <1k/64/64, \$1E9007> {U1} [INT/EXT]
ATtiny15(L) <1k/64/-, \$1E9006> {U1} [INT]
ATtiny22(L) <2k/128/128, \$1E9106*> {U1} [INT/EXT]
ATtiny25(V) <2k/128/128, \$1E9108> {U1} [RES/INT/EXT]
ATtiny45(V) <4k/256/256, \$1E9206> {U1} [RES/INT/EXT]
ATtiny85(V) <8k/512/512, \$1E930B> {U1} [RES/INT/EXT]

DIP14:

ATtiny24(A/V) <2k/128/128, \$1E910B> {U2} [RES/INT/EXT]
ATtiny44(A/V) <4k/256/256, \$1E9207> {U2} [RES/INT/EXT]
ATtiny84(A/V) <8k/512/512, \$1E930C> {U2} [RES/INT/EXT]

DIP20:

AT90S1200 <1k/64/-, \$1E9001*> {U3} [RES/INT/EXT]
AT90S2313 <2k/128/128, \$1E9101*> {U3} [RES/EXT]
ATtiny2313(A/V) <2k/128/128, \$1E910A> {U3} [RES/INT/EXT]
ATtiny4313 <4k/256/256, \$1E920D> {U3} [RES/INT/EXT]
ATtiny26(L) <2k/128/128, \$1E9109> {U4} [RES/INT/EXT]
ATtiny261(A/V) <2k/128/128, \$1E910C> {U4} [RES/INT/EXT]
ATtiny461(A/V) <4k/256/256, \$1E9208> {U4} [RES/INT/EXT]
ATtiny861(A/V) <8k/512/512, \$1E930D> {U4} [RES/INT/EXT]

DIP28:

AT90(S/LS)2333 <2k/128/128, \$1E9105*> {U5} [RES/EXT]
AT90(S/LS)4433 <4k/256/128, \$1E9203*> {U5} [RES/EXT]
ATtiny48 <4k/64/256, \$1E9209> {U5} [INT/EXT]
ATtiny88 <8k/64/512, \$1E9311> {U5} [INT/EXT]
ATmega8(A/L) <8k/512/1k, \$1E9307> {U5} [RES/INT/EXT]
ATmega48(A/P/PA) <4k/256/512, \$1E9205 (P/PA:\$1E920A)> {U5} [RES/INT/EXT]
ATmega88(A/P/PA) <8k/512/1k, \$1E930A (P/PA:\$1E930F)> {U5} [RES/INT/EXT]
ATmega168(A/P/PA) <16k/512/1k, \$1E9406 (P/PA:\$1E940B)> {U5} [RES/INT/EXT]
ATmega328(P) <32k/1k/2k, \$1E9514 (P:\$1E950F)> {U5} [RES/INT/EXT]

DIP40:

AT89S51 <4k/-/128, \$1E5106**> {U6} [RES/EXT] RST:HIGH
AT89S52 <8k/-/256, \$1E5206**> {U6} [RES/EXT] RST:HIGH
AT89S53 <12k/-/256, \$1E53***> {U6} [RES/EXT] RST:HIGH
AT89S8252 <8k/2k/256, \$1E72***> {U6} [RES/EXT] RST:HIGH
AT89S8253 <12k/2k/256, \$1E73***> {U6} [RES/EXT] RST:HIGH
AT90S4414 <4k/256/256, \$1E9201*> {U6} [RES/EXT]
AT90S8515 <8k/512/512, \$1E9301*> {U6} [RES/EXT]
ATmega161(L) <16k/512/1k, \$1E9401> {U6} [RES/EXT]
ATmega162(V) <16k/512/1k, \$1E9404> {U6} [RES/INT/EXT]
ATmega8515(L) <8k/512/512, \$1E9306> {U6} [RES/INT/EXT]
AT90(S/LS)4434 <4k/256/256, \$1E9202*> {U7} [RES/EXT]
AT90(S/LS)8535 <8k/512/512, \$1E9303*> {U7} [RES/EXT]
ATmega16(A/L) <16k/512/1k, \$1E9403> {U7} [RES/INT/EXT]
ATmega32(A/L) <32k/1k/2k, \$1E9502> {U7} [RES/INT/EXT]
ATmega163(L) <16k/512/1k, \$1E9402> {U7} [RES/INT/EXT]
ATmega164(A/V/P/PA) <16k/512/1k, \$1E940F (P/PA:\$1E940A)> {U7} [RES/INT/EXT]
ATmega323(L) <32k/1k/2k, \$1E9501> {U7} [RES/INT/EXT]
ATmega324(A/V/P/PA) <32k/1k/2k, \$1E9515 (P:\$1E9508, PA:\$1E9511)> {U7} [RES/INT/EXT]
ATmega644(A/V/P/PA) <64k/2k/4k, \$1E9609 (P/PA:\$1E960A)> {U7} [RES/INT/EXT]
ATmega8535(L) <8k/512/512, \$1E9308> {U7} [RES/INT/EXT]

Oznaczenia:

RST:HIGH – układ resetowany wysokim poziomem logicznym

Oznaczenia dotyczące napięcia zasilania:

S – minimum 4V

L, LS – minimum 2.7V

V, P – minimum 1.8V

Oznaczenia dotyczące sygnału taktującego:

[INT] – układ zawiera wbudowany generator

[EXT] – układ może pracować z zewnętrznym sygnałem taktującym

[RES] – układ może pracować z dołączonym rezonatorem kwarcowym

Oznaczenia dotyczące sygnatury układu:

- * – kiedy obydwa bity zabezpieczające są zaprogramowane (Lock mode 3), sygnatury nie da się odczytać szeregowo (Serial low-voltage mode). Wtedy sygnatura zwróci wartość \$000102.
3-bajty sygnatury znajdują się pod adresem \$0001, \$0002 i \$0003.
- ** – podczas odczytu sygnatury piny 16 (P3.6) i 17 (P3.7) muszą mieć stan niski.
3-bajty sygnatury znajdują się pod adresem \$0000, \$0100 i \$0200.
- *** – podczas odczytu sygnatury piny 16 (P3.6) i 17 (P3.7) muszą mieć stan niski.
2-bajty sygnatury znajdują się pod adresem \$0030 i \$0031.

