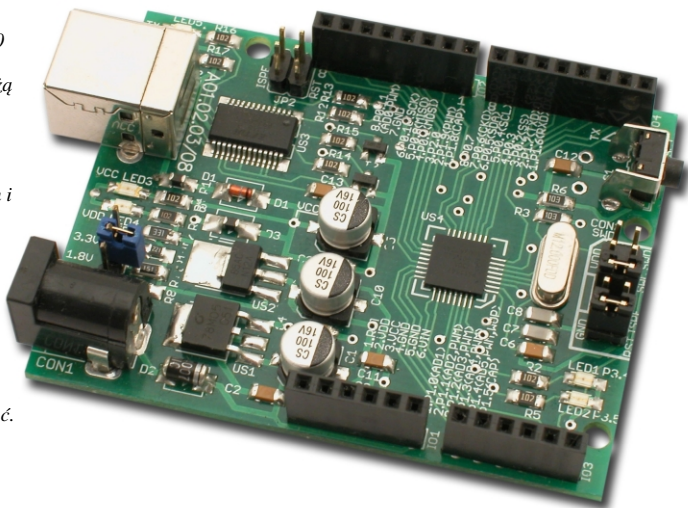


# AVT 1620

## Cortexino Kompatybilna z Arduino płytką z LPC1114 (rdzeń Cortex-M0)

Mikrokontrolery z rdzeniem CORTEX M0 mają szansę zastąpić mikrokontrolery 8 bitowe takie jak AVR. Mają nad nimi dużą przewagę: są w pełni 32 bitowe, szybsze, lepiej wyposażone i co bardzo ważne konkurencyjne cenowo. Dla przykładu, bardzo popularny mikrokontroler AVR ATMEGA8 który ma 8kB flash, 1 kB ram i pracuje z maksymalną częstotliwością 16MHz kosztuje tyle samo lub nawet więcej, niż LPC1114 który ma 4 razy więcej pamięci flash, 8 razy więcej pamięci ram i jest 3 razy szybszy. Wybrany model, LPC1114 jest raczej ubogo wyposażony w porównaniu do innych mikrokontrolerów tego rodzaju, ale dzięki temu odpowiedni do nauki, bo nie odstrasza mnogością nieznanych pojęć.



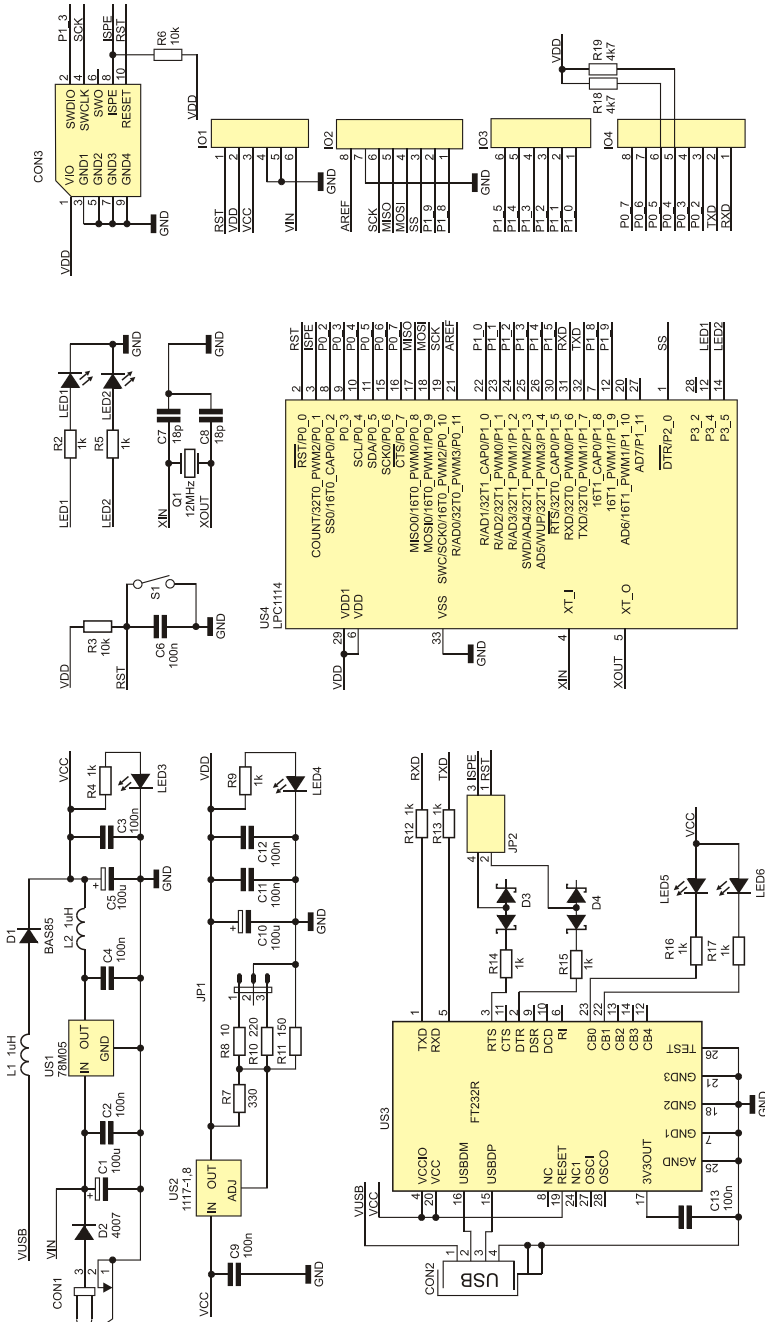
## Właściwości

- mikrokontroler 32-bit LPC1114 z rdzeniem Cortex M0, 32kB flash, 8kB ram, 50MHz;
- wbudowany konwerter USB-UART;
- wbudowany programator ISP z interfejsem USB;
- darmowe środowisko programistyczne;
- rozmieszczenie złączy kompatybilne z Arduino;

## Opis układu

Na płytce znajdują się wszystkie elementy potrzebne do tego, aby zacząć pracę z mikrokontrolerem. Zasilanie może być pobierane z portu USB, należy wtedy pamiętać że prąd nie powinien przekraczać 100mA a napięcie VCC będzie miało wartość ok. 4,5V. Jeżeli przewidujemy większy pobór prądu np. przez peryferia dołączone do płytki, to należy zasilic układ z zewnętrznego zasilacza 7...12VDC. Wtedy napięcie VCC jest dostarczane przez stabilizator US1 i ma dokładnie 5V oraz można je obciążyć do 0,5A co w większości wypadków w zupełności wystarczy. Dodatkowo wszystkie napięcia zasilające są wyprowadzone na złącze IO1. Do zasilania mikrokontrolera potrzebne jest napięcie z przedziału 1,8...3,6V, które dostarcza stabilizator US2. Złącze JP1 pozwala wybrać jedną z trzech wartości 1,8, 2,8 lub 3,3V, poprzez odpowiednie założenie jumpera. Diody LED3 i LED4 sygnalizują obecność napięć zasilających. Układ US3 to popularny konwerter USB-UART, który pełni dwie funkcje. Po pierwsze pozwala programować pamięć mikrokontrolera w trybie ISP. Po drugie pozwala na komunikację z komputerem za pośrednictwem interfejsu szeregowego, co może być wykorzystane jako prosty debugger. Złącze JP2, poprzez założenie dwóch jumperów, pozwala dołączyć sygnały RESET i ISP ENABLE do dodatkowych wyprowadzeń układu FT232, a to zapewnia pełną automatyzację procesu programowania. Diody LED5 i LED6 sygnalizują aktywność interfejsu USB. Przycisk S1 służy do restartowania mikrokontrolera, Q1 jest źródłem sygnału taktującego, diody LED1 i LED2 pełnią rolę sygnalizacyjną, mogą być wykorzystane w dowolny sposób. Porty mikrokontrolera zostały wyprowadzone do

złącze IO2, IO3 i IO4. Złącze CON3 umożliwia dołączenie zewnętrznego programatora/debuggera z interfejsem SWD. Rozmieszczenie gniazd jest kompatybilne z Arduino Delimeanove Board co pozwala na wykorzystanie tych samych modułów rozszerzających możliwości płytki bazowej (tzw. shields). Warto zwrócić uwagę na to, że

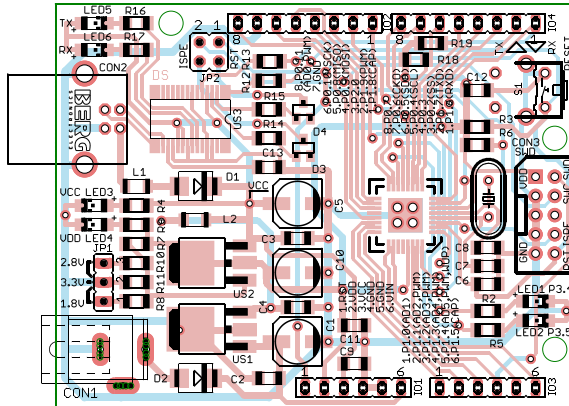


Rys. 1 Schemat elektryczny

PORT0\_4 i PORT0\_5 są także szyną I<sup>2</sup>C i pracują jako wyjścia OD (open drain), dlatego zostały wyposażone w zewnętrzne rezystory podciągające.

## Montaż i uruchomienie

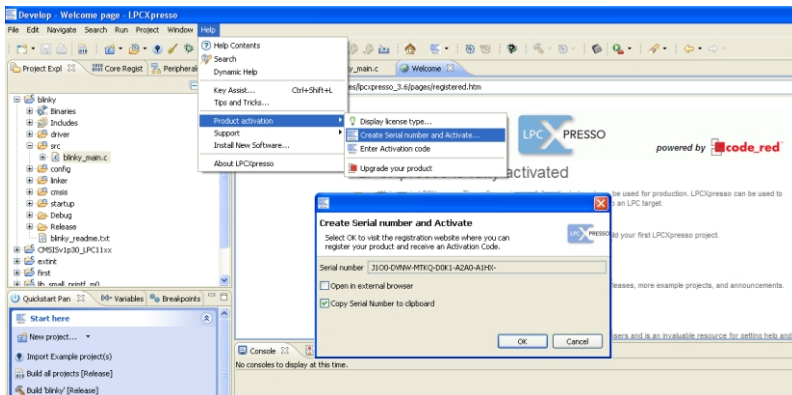
Montaż wykonujemy zgodnie z ogólnymi zasadami według schematu montażowego z rys.2. Wlutowanie mikrokontrolera, który jest w obudowie HQFN33 nie jest trudniejsze od wlutowania układu w obudowie TQFP. Poradzi sobie z tym każdy kto ma odrobinę wprawy w montażu elementów SMD. Szczególną uwagę należy zwrócić na dokładne ułożenie układu na płytce jeszcze zanim zaczniemy go lutować. Wyprowadzenia na każdej krawędzi muszą nachodzić na odpowiadające im pady. Tak ułożony układ przyciskamy delikatnie pęsetą, lutujemy 1...2 piny i kontrolnie sprawdzamy czy nie się nie przesunęło. Następnie przeciągamy grotom po wszystkich krawędziach a potem usuwamy nadmiar cyny, oczywiście nie może zabraknąć topnika. Na koniec należy od strony lutowania obficie zaalac cyną przelotki pod układem, ponieważ tak jest doprowadzona masa układu.



Rys. 2 Rozmieszczenie elementów na płytce drukowanej

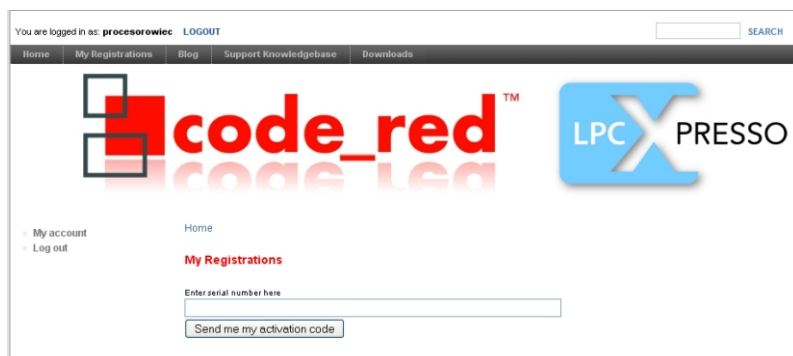
## Środowisko programistyczne

Przygotowanie kompletnego środowiska sprowadza się do zainstalowania dwóch programów. Pierwszy z nich to LPCXpresso - zintegrowane środowisko programistyczne dla mikrokontrolerów LPC oparte na Eclipse. Wersja instalacyjna dostępna jest na stronie <http://lpcxpresso.code-red-tech.com/LPCXpresso/> w zakładce „Download”, po



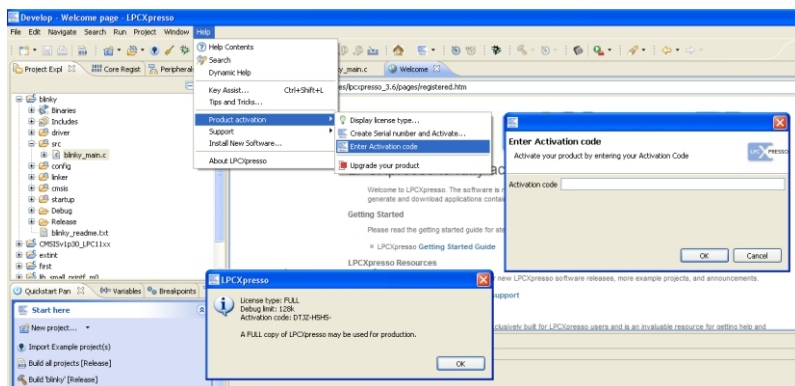
Rys. 3

uprzednim zalogowaniu. Aby stworzyć konto należy kliknąć pole „Create Account” i wypełnić odpowiednie pola swoimi danymi, po zatwierdzeniu, na skrzynkę e-mailową otrzymamy hasło dostępu do naszego konta. Drugi program, który będzie nam potrzebny to Flash Magic – narzędzie do programowania pamięci mikrokontrolera także poprzez interfejs UART z wykorzystaniem bootloadera. Wersja instalacyjna dostępna jest na stronie <http://www.flashmagictool.com/>. Instalacja obu programów nie powinna sprawić problemów i sprowadza się do zaakceptowania warunków licencji i kilku kliknięć „Next”. Po zainstalowaniu uruchamiamy **LPCXpresso**, który powita nas okienkiem z informacją o tym, że ta wersja programu nie jest zarejestrowana i w związku z tym ma znaczne ograniczenia. Rejestracja jest bezpłatna, aby ją wykonać należy w zakładce „Help” wybrać „Product activation” i „Create Serial number and Activate...”. Pojawi się okno, w którym zostanie wygenerowany numer seryjny, zaznaczamy „Copy Serial Number to clipboard” i klikamy „OK” **rys. 3**. Zostanie otworzona strona code-red-tech.com gdzie musimy się zalogować, następnie w zakładce „My Registrations” w polu „Enter serial number here” wkleić wygenerowany numer seryjny **rys. 4**. Kod aktywacyjny zostanie wysłany na naszą skrzynkę e-



Rys. 4

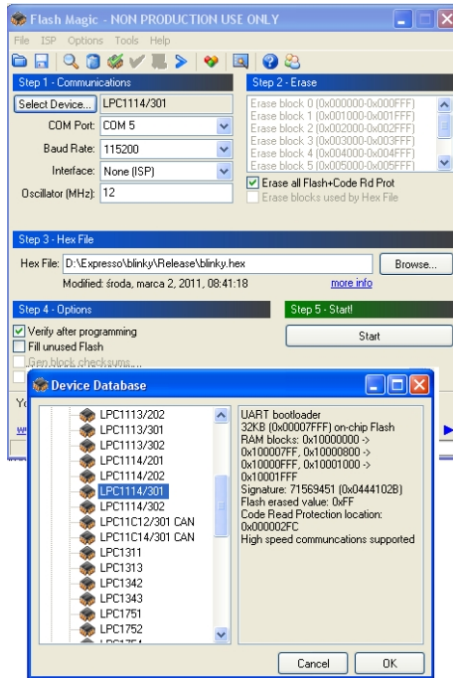
mailową, kopiujemy go i wklejamy w zakładce „Help”, „Product activation” i „Enter Activation Code”, **rys 5**. Wyświetli się informacja, że posiadamy licencję typu „FULL”, która pozwala wykorzystywać oprogramowanie w celach produkcyjnych a jedynym ograniczeniem jest debugowanie kodu do 128kB.



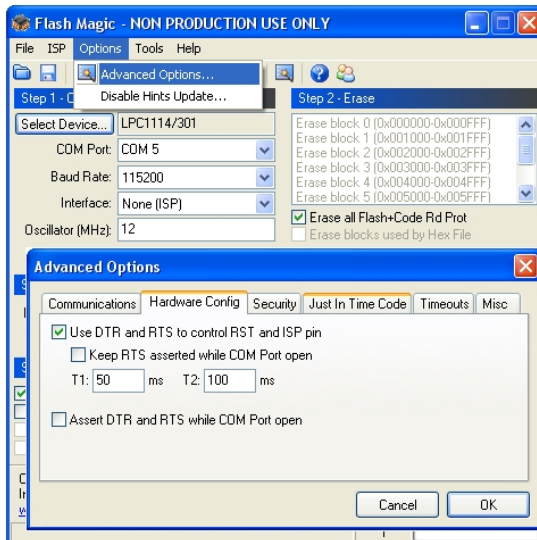
Rys. 5

Program Flash Magic nie wymaga rejestracji, ale posiada jedno ograniczenie – nie może być wykorzystywany w celach produkcyjnych. Jeśli zestaw uruchomieniowy Cortexino jest podłączony do komputera to możemy sprawdzić

poprawność komunikacji. Zworki JP2 powinny być założone, w polu „Selekt Device...” wybieramy LPC1114/301, w polu „COM Port” wybieramy numer portu szeregowego pod którym został zainstalowany zestaw, w polu „Baud Rate” - 115200, w polu „Interface” – None (ISP), w polu „Oscillator(MHz)” – 12, jak na rys. 6. Następnie w zakładce „Options”, „Advanced Options...” , „Hardware Config” zaznaczamy pole „Use DTR and RTS to control RST and



Rys. 6

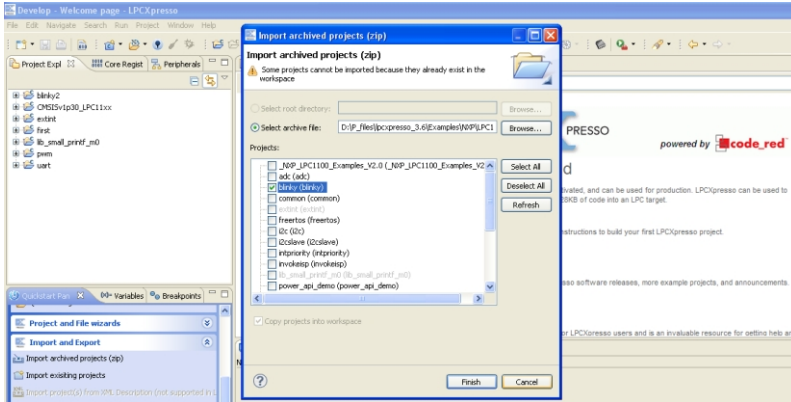


Rys. 7

ISP pin”, rys. 7. Następnie w zakładce „ISP” klikamy na „Read Device Signature” i jeśli wszystko pracuje prawidłowo wyświetli się okno z odczytanymi parametrami.

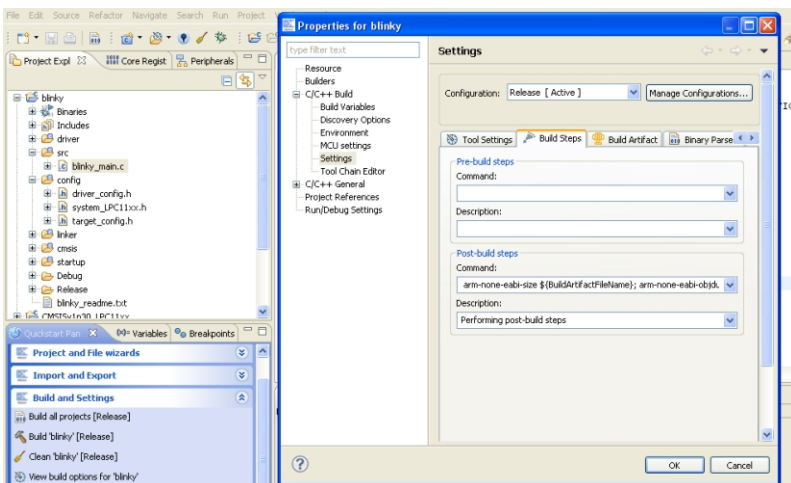
## Pierwszy program

LPCXpresso zawiera wiele przykładów programów, znajdują się one w katalogu programu w podkatalogu „Examples”. Aby otworzyć przykładowy program należy w lewej dolnej części okna odszukać zakładkę „Quickstart Pan” a w niej „Import and Export” i „Import archived projects (zip)”. W wyświetlonym oknie należy odszukać folder „Examples” a następnie podkatalog dla rodziny układów LPC111x i przeszukać istniejące zbiory w poszukiwaniu projektu o nazwie „blinky”, rys 8. W lewej górnej części okna w zakładce „Project Expl” pojawia się



Rys. 8

pogrupowane tematycznie zasoby projektu. Rozwijamy katalog „src”, w którym znajduje się główny plik źródłowy „blinky\_main.c”, jego działanie polega m. in. miganiu diodą led. Aby uruchomić projekt na na płycie Cortexino trzeba wprowadzić małe zmiany. W zasobach projektu, w katalogu „config” otwieramy plik „target\_config.h” i zmieniamy następujące linie:



Rys. 9

```
#define LED_PORT      0
#define LED_BIT      7
```

Zamieniamy na:

```
#define LED_PORT      3
#define LED_BIT      4
```

Teraz projekt jest gotowy do uruchomienia i możemy go skompilować jednak domyślnie zostanie wygenerowany plik bin a Flash Magic wymaga plików hex. Aby zmienić rodzaj pliku wynikowego należy: w zakładce „Quickstar Pan” w kategorii „Build and Settings” otwieramy „View build options for ‘blinkly’”, w otwartym oknie przechodzimy do zakładki „Build Steps”, **rys 9**. Warto teraz ustawić pole „Configuration” na „Release” i ustawić tę opcję jako aktywną „Release [ Active ]”. W ten sposób ustawiamy konfigurację produkcyjną a nie uruchomieniową, co wiąże się m. in. z lepszą optymalizacją pliku wynikowego. Następnie w ramce „Post-build steps” w polu „Command” zmieniamy treść na:

```
arm-none-eabi-size ${BuildArtifactFileName};
arm-none-eabi-objdump -S ${BuildArtifactFileName} >
{BuildArtifactFileBaseName}.lss;
arm-none-eabi-objcopy -O ihex ${BuildArtifactFileName} $
{BuildArtifactFileBaseName}.hex;
```

Temat ten jest dokładnie opisany na stronie code-red-tech w dziale „Support Knowledgebase”.

W końcu możemy skompilować projekt, w zakładce „Quickstar Pan” w kategorii „Build and Settings” klikamy „Build ‘blinkly’ [Release]”. Po krótkiej chwili w zakładce „Console” w dolnej części ekranu dostaniemy podsumowanie, w zasobach projektu pojawi się folder „Release” a w nim plik blinkly.hex. Teraz uruchamiamy Flash Magic, w ramce „Step 3 - Hex File” odszukujemy plik blinkly.hex i klikamy „Start”. Efektem działania programu będzie migająca dioda led na płytce Cortexino.

## Wykaz elementów

### Rezystory:

R2, R4, R5, R9, R12...R17 .....1kW  
R3, R6 .....10kW  
R7 .....330W  
R8 .....10W  
R10 .....220W  
R11 .....150W  
R18, R19 .....4k7W

### Kondensatory:

C1, C5, C10 .....100mF / 16V SMD C  
C2, C3, C4, C6, C9, C11, C12, C13 ...100nF SMD  
C7, C8 .....18pF SMD

### Półprzewodniki:

D1 .....BAS85  
D2 .....1N4007 SMD  
D3, D4 .....BAR43  
LED1...LED6 .....LED SMD  
US1 .....78M05  
US2 .....1117-1.8  
US3 .....FT232R  
US4 .....LPC1114 HQFN33/301

### Pozostałe:

L1, L2 .....Dławik 1...10mH SMD  
S1 .....mikroswitch kątowy  
Q1 .....rezonator 12MHz  
JP1 .....goldpin 1x3 + jumper  
JP2 .....goldpin 2x2 + 2x jumper  
IO1, IO3 .....gniazdo goldpin 1x6  
IO2, IO4 .....gniazdo goldpin 1x8  
CON1 .....GN DC2.1/5.5 kątowe  
CON2 .....USB B kątowe  
CON3 .....goldpin 2x5

Zestaw powstał na podstawie projektu o tym samym tytule opublikowanego w Elektronice Praktycznej 05/11

**ELEKTRONIKA  
PRAKTYCZNA**

[www.ep.com.pl](http://www.ep.com.pl)

Oferta zestawów do samodzielnego montażu dostępna jest na stronie internetowej [www.sklep.avt.pl](http://www.sklep.avt.pl)



tel.: (22) 257-84-50  
fax: (22) 257-84-55

**Producent:**

AVT-Korporacja sp. z o.o.  
ul. Leszczynowa 11  
03-197 Warszawa

**Dział pomocy technicznej:**

tel.: (22) 257-84-58  
[serwis@avt.pl](mailto:serwis@avt.pl)