AVT 1620

Cortexino Kompatybilna z Arduino płytka z LPC1114 (rdzeń Cortex-M0)

Mikrokontrolery z rdzeniem CORTEX M0 mają szanse zastapić mikrokontrolery 8 bitowe takie jak AVR. Maja nad nimi duża 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 pamieci ram i jest 3 razy szybszy. Wybrany model, LPC1114 jest raczej ubogo wyposażony w porównaniu do innych mikrokontrolerów tego rodzaju, ale dzieki temu odpowiedni do nauki, bo nie odstrasza mnogością nieznanych pojeć.



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 złącz 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 zasilić układ z zewnętrznego zasilacza 7...12VDC. Wtedy napięcie VCC jest dostarczane przez stabilizator US1 i ma dokładnie 5V oraz można je obciażyć do 0,5A co w wiekszości wypadków w zupełności wystarczy. Dodatkowo wszystkie napięcia zasilające są wyprowadzone na złączu 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 sygnalizacyjna, mogą być wykorzystane w dowolny sposób. Porty mikrokontrolera zostały wyprowadzone do

złącz 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

 $PORTO_4$ i $PORTO_5$ są także szyną I²C 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 nic się nie przesunęło. Następnie przeciągamy grotem po wszystkich krawędziach a potem usuwamy nadmiar cyny, oczywiście nie może zabraknąć topnika. Na koniec należy od strony lutowania obficie zalać 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 - zintegrorowane 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-milową 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 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-

You are logg	ged in as: procesorowiec LOGOUT	SEARCH
Home	My Registrations Blog Support Knowledgebase Downloads	
	code_re	
My acc	count Home	
= Log ou	d My Registrations	
	Enter serial number here	
	Send me my activation code	

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 debagowanie kodu do 128kB.

Conception Provide Conception Conce			
File Edit Navigate Search Run Project Window	Help		
Project Expl 23 In Core Regist 7 Peripheral	Help Contents Search Dynamic Help Key Assist Ctrl+Shift+L Tras and Table	D D D D C C C C C C C C C C C C C C C C	╊╡℁∊⋓∊ ∶©∣Q₊∶ ∤⋪∙∶⇔∙⇔
B W Binaries ⊕ S Includes B Ø driver ⊕ Ø ac	Product activation Support Install New Software	Display license type Create Serial number and Activate Erker Activation code	Enter Activation code
Binky_man.c Config C	About LPC/presso	Upgrade your product	Activate your product by entering your Activation Code
Control Contro Control Control Control Control Control Co		generate and download applications co Getting Started	n n Inter
binky_readme.txt e		Hesse read the getting started guide to * LPCXpresso Getting Started Guide LPCXpresso Resources	e OK Cancel
* 🖂 lin anal mintf mil	EPCX presso		new LPCXpresso software releases, more example projects, and announcements.
Quidistart Pan 23 00 Variables % Breakpoints Start here	B Ucense type: FULL Debug Imit: 128k Activation code: DTJZ-HSH5-		upport
Mew project The second se	A FULL copy of LPC/presso me	ay be used for production.	Luderthy best for Linuxertose users and a an invaluable resource for definit help and
Build blinky [Release]			

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", "Advenced Options…", "Hardware Config" zaznaczamy pole "Use DTR and RTS to control RST and

tash Mag	ic - NON PRODUCT		ONLY		×
File ISP Opti	ons Tools Help				
۵ 🖬 🔍	🗃 🍏 🖌 📕 🗲	*	2		
Step 1 - Comm	inications		Step 2 - Erase		
Select Device.	. LPC1114/301		Erase block 0 (0x000	000-0x000FFF)	^
COM Po	t COM 5	~	Erase block 2 (0x002	000-0x001FFF)	
Baud Rat	e: 115200	*	Erase block 3 (0x003 Erase block 4 (0x004	000-0x003FFF) 000-0x004FFF)	
Interfac	e: None (ISP)	~	Erase block 5 (0x005	000-0x005FFF)	<u> </u>
Oscillator (MH:	:): 12		Erase all Hash+Co Erase blocks used	de Hd Prot by Hex File	
Step 3 - Hex Fi	e				
Hex File: D:\F	xnresso\blinkv\Belease	hlinku hex		Browse	
Modi	ied: środa. marca 2. 201	1.08:41:18	mor	re info	
Step 4 Detien			Shan E. Sha	at	
Step 4 - Option	•		Step 5 - Star	u.	
Fill unused Fi	ogramming ach			Start	
Gen.block.cl	iecksums				
Gen.block.cl	ecksums Database				
Gen Meck of	PC1113/202 PC1113/202 PC1113/202 PC1113/202 PC1113/202 PC1114/201 PC1114/201 PC1114/202 PC1114/202 PC1114/202 PC1114/202 PC1114/202 PC1114/201 PC1313 PC1313 PC1313 PC1314 PC1314 PC1314 PC1314 PC1314 PC1314 PC1315 PC1314 PC1315 PC1315 PC1314 PC1315 PC135		APT bootbader 26 (buo0007FF) on 24 blocks: 0:100000 100007FF, 0:100001 (00007FF, 0:100001 (00001FFF, 0:100010 gnatue; 7155345) 200002FC 2000002FC gh speed communcal	-chip Flash 00 -> 100 -> 100 -> 100 -> 100 -> k0441028) F cocation: tions supported	

Rys. 6





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 "Exapmles". Aby otworzyć przykładowy program należy w lewej dolnej części okna odszukać zakładkę "Quickstatrt 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 LPC11xx i przeszukać istniejące zbiory w poszukiwaniu projektu o nazwie "blinky", **rys 8**. W lewej górnej części okna w zakładce "Project Expl" pojawią się

🖀 Develop - Welcome page - LPCXpresso		
File Edit Navigate Search Run Project Window Help	Import archived projects (zip)	3 8 * 1 © Q. • 1 ∕? • 1 ∕= • ↔ •
Project Expl 🛛 🗰 Core Regist 💏 Peripherals 🔍 🗌	▲ Some projects cannot be imported because they already exist in the workspace	
8 년 binky2 8 년 OKISV(p30_PC11xx 8 년 Strint 8 년 first 8 년 By smal_printf_m0	Select root directory: Browse Ø Select archive Fle: DrJP_fles](pcopresso_3.6)Examples/V6P/LIPC1 Browse Projekta: Projekta: DrJP_fles](pcopresso_3.6)Examples/V6P/LIPC1 Browse	PRESSO
a ⊴open ≊ Sont	MP UPC1002_Densemble_V2.0. (MPP_UPC1100_Enamples_V2.00) Select AI	d hvalut, and can be used for production. LPCQreases can be used to BKB of code in the ALC larget. advactions to build your first LPCQreases project.
Quidstart Par X M- Variables On Breakpoints Project and File wizards	Copy projects into workspace	
Import and Export		or LPCXoresso users and is an invaluable resource for cetting help and
Timport exisiting projects Timport exisiting projects Timport project(s) from WML Description (not supported in L	? Finish Cancel	

pogrupowane tematycznie zasoby projektu. Rozwijamy katalog "scr", w którym znajduje się główny plik źródłowy "blinky_main.c", jego działanie polega m. in. miganiu diodą led. Aby uruchomić projekt na na płytce Cortexino trzeba wprowadzić małe zmiany. W zasobach projektu, w katalogu "config" otwieramy plik "target_config.h" i zmieniamy następujące linie:

File Edit Source Reflactor Navigate Search Run Project	E Properties for blinky	
The Let Source Relation Newpole Search Run Project V C - Garden Search Relation Newpole Search Run Project V Project Expl 23 III Core Regist 2 Perphered C C - Briefs C -	Properties for blinky Type filer text Pasource Bulders Bulders Decovery Options MCU settings Tool Chan Editor Project References Run/Debug Settings	Settings
Release Dinky_readme.txt CMSTSv1v90_LPC11vv		Description: Performing post-build steps
Quickstart Pan X (M= Variables) Breakpoints A Poject and File wizards X		
🖺 Import and Export 🛞		
Build and Settings		
Nij buid al projects (kelease)		
Clean 'blinky' [Release] Si View build options for 'blinky'	?	OK Cancel

Rys. 9

Rys. 8

#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 bulid options for 'blinky'", w otwartym oknie przechodzimy do zakładki "Build Steps", **rys 9.** Warto teraz ustawić pole "Configuration" na "Release" i ustawić tę opcje 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-buil stesps" 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 'blinky' [Release]". Po krótkiej chwili w zakładce "Console" w dolnej części ekranu dostaniemy podsumowanie, w zasobach projektu pojawi się folder "Release" a w nim plik blinky.hex. Teraz uruchamiamy Flash Magic, w ramce "Step 3 - Hex File" odszukujemy plik blinky.hex i klikamy "Start". Efektem działania programu będzie migająca dioda led na płytce Cortexino.

Wykaz elementów

Rezystory:

R2, R4, R5, R9, R12…F	3171kW
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
С7, С8	18pF SMD
Półprzewodniki:	
D1	BAS85
D2	1N4007 SMD
D3, D4	BAR43
LED1LED6	LED SMD
US1	78M05
US2	1117-1.8
US3	FT232R
US4	LPC1114 HQFN33/301

Pozostałe:

L1, L2	Dławik 110mH SMD
S1	mikroswitch kątowy
Q1	rezonator 12MHz
JP1	goldpin 1x3 + jumper
JP2	goldpin 2x2 + 2x jumper
101, 103	gniazdo goldpin 1x6
102, 104	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



www.ep.com.pl

Oferta zestawów do samodzielnego montażu dostępna jest na stronie internetowej www.sklep.avt.pl



Dział pomocy technicznej:

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