

BF100 – linuksowy ARMputer, część 2

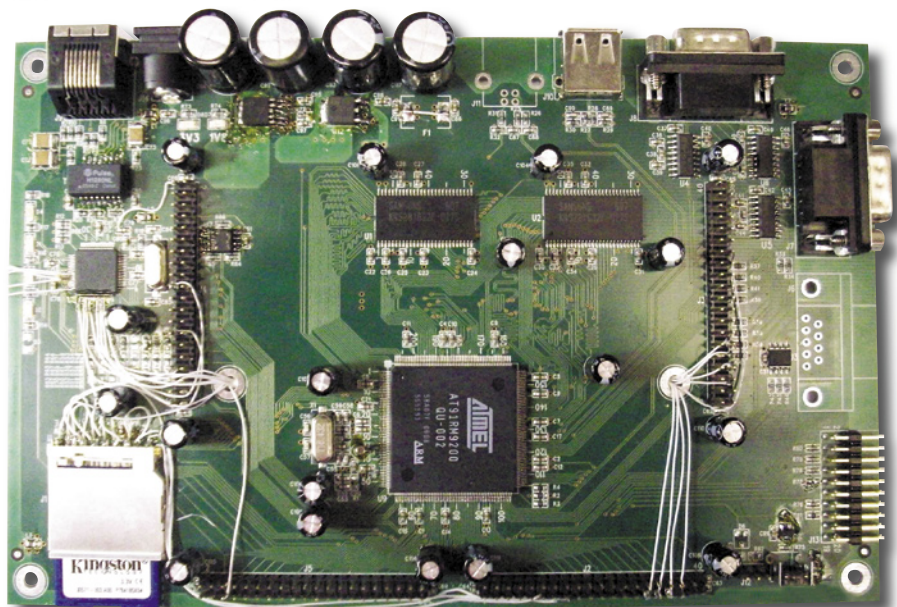
AVT-5128

Pomimo konstrukcyjnych niedociągnięć, ARMputer cieszy się dużym powodzeniem wśród naszych Czytelników. W drugiej (ostatniej) części artykułu przedstawiamy sposób jego montażu i uruchomienia, a także sposobu przygotowania instalacji Linuxa, która jest jednym z ważniejszych „smaczków” całego opracowania.

Rekomendacje:

opisany w artykule komputer jest dobrze wyposażoną platformą sprzętową, dla której przygotowaliśmy implementację systemu operacyjnego Linux.

Jego wykorzystanie przynosi programistom w świat aplikacji innych pisanych „wprost” na mikrokontrolery. Jeszcze kilka lat temu takie możliwości można było zakwalifikować do kategorii opowiadań science-fiction.



Montaż i uruchomienie

Pomimo iż modelowe płytki ARMputera z założenia miały zostać zmontowane automatycznie, podczas konstruowania ARMputera wykorzystano tylko takie elementy, które dadzą się zmontować w warunkach amatorskich. Zatem całość przy odrobinie wprawy możemy zmontować w warunkach amatorskich. Największą trudność podczas montażu może sprawić mikrokontroler AT91RM9200, który ma 208-wyprowadzeniową obudowę TQFP, niemniej jednak przy zachowaniu odrobiny ostrożności, nie będzie to zadanie niemożliwe do wykonania.

Montaż rozpoczynamy od zamontowania elementów biernych, takich jak rezystory, czy kondensatory, które występują w obudowach do montażu powierzchniowego 0603. Należy zwrócić szczególną uwagę na rezystory R47...R71 znajdujące się pomiędzy kartą MMC a układem STE100, ponieważ musimy tutaj wlutować tylko rezystory 0 Ω (zwory): R59, R52, R54, R48, R47, R66, R64, R71, natomiast pozostałe pola lutownicze zostawić nieobsadzone. Wspomniane wcześniej rezystory stanowią matrycę krosującą, służącą do konfiguracji układu STE100p. Po zmontowaniu biernych elementów SMD przystępujemy do

montażu półprzewodników, poczynając od małych układów takich jak MAX3232, MAX485, poprzez pamięci SDRAM i układ STE100p, na procesorze AT91RM9200 kończąc. Podczas montażu układów z wieloma wyprowadzeniami, tak jak STE100p (U8) czy AT91RM9200 (U9), należy najpierw je wcześniej odpowiednio wypozycjonować, a następnie przytwierdzić do podłoża lutując narożne piny. Po odpowiednim pozycjonowaniu na całe pole lutownicze możemy nałożyć topnik RMA7, który najlepiej nadaje się do tego celu, a następnie nabierając odrobinę cyny w miarę szybkim ruchem zalutować cały rząd wyprowadzeń. Skrajne wyprowadzenia mogą się nam zlutować razem, niemniej jednak nie mamy się czym przejmować i nadmiar cyny należy odessać za pomocą popularnej plecionki.

Po zakończeniu lutowania należy jeszcze sprawdzić pod lupą czy wyprowadzenia nie są ze sobą powziewierane. Po zamontowaniu układów półprzewodnikowych możemy zamontować transformator TR1, a następnie pozostałe złącza oraz kondensatory elektrolityczne, które znacząco wystają ponad płytkę. Podczas projektowania płytki niestety wkradło się kilka błędów, więc podczas montażu gniazda pa-

PODSTAWOWE PARAMETRY

- Mikroprocesor AT91RM920 (ARM920T)
- 32 MB SDRAM
- 128 kB Data Flash SPI
- Interfejs Ethernet
- 2xRS232 (jeden z kompletem linii)
- Interfejs MMC/SD
- Interfejsy USB-device i USB-host
- Interfejs RS485
- Zasilanie 5 V/800 mA (w tym 500 mA rezerwy dla USB-host)

mięci MMC należy odgiąć do góry wszystkie piny. W związku z inną numeracją użytego w projekcie *footprintu* dla osiągnięcia poprawności połączeń nastąpiło przenumerywanie pinów złącza SD. Zatem musimy z padów złącza karty MMC/SD oraz z gniazd J2 i J5 doprowadzić sygnały za pomocą druczika kynarowego następująco: do pierwszego pinu od lewej strony złącza karty należy doprowadzić sygnał ze złącza J2 pin 13. Do drugiego pinu od lewej należy doprowadzić sygnał ze złącza J2 pin 15. Do 3 pinu złącza karty należy doprowadzić sygnał z padu karty oznaczonego CMD. Do 4 pinu należy doprowadzić sygnał z padu VSS, natomiast do pinu nr 5 doprowadzamy sygnał z padu oznaczonego VDD. Do 6 pinu karty dołączamy sygnał z padu oznaczonego CLK, natomiast do pinu 7 doprowadzamy sygnał z padu VSS. Do pinu 8 dołączamy sygnał z padu DAT, natomiast do pinu 9 dołączamy sygnał ze złącza J2 pin 11. Również rezystor R77 musimy dolutować z drugiej strony płytki – nie do linii 3,3 V, a GND. Niestety trochę błędów wkradło się także w okolicach układu STE100p, dlatego musimy przeciąć połączenie doprowadzające sygnał do złącza J4 pin 27, a następnie za pomocą kynaru dołączyć go do pinu 48 układu STE100p. Następnie za pomocą kynaru ze złącz J2, J3, J4 musimy doprowadzić kilka sygnałów do układu STE100p w następujący sposób:

```
PIN29/J2 – pin57/U8
PIN31/J2 – PIN58/U8
PIN33/J2 – PIN52/U8
PIN35/J2 – PIN44/U8
PIN5/J3 – PIN43/U8
PIN19/J4 – PIN53/U8
PIN11/J3 – PIN49/U8
PIN7/J3 – PIN48/U8
PIN9/J3 – PIN59/U8
PIN27/J4 – PIN60/U8
```

Po zmontowaniu wszystkich elementów i doprowadzeniu brakujących połączeń możemy przystąpić do uruchamiania płytki. W tym celu do złącza J14 dołączamy napięcie zasilające o wartości 5 V ustawiając ograniczenie prądowe na wartość około 200 mA. Jeżeli ograniczenie prądowe nie zadziała należy zmierzyć napięcia na wyjściu stabilizatora LF33 (U11), które powinno wynosić około 3,3 V oraz napięcie na wyjściu stabilizatora LF18 (U12), którego wartość powinna być zbli-

żona do 1,8 V. Natomiast jeżeli, któreś z napięć ma nieprawidłową wartość lub układ pobiera więcej niż 150...200 mA, należy jeszcze raz sprawdzić poprawność montażu wszystkich elementów najlepiej pod lupą, ponieważ przy tak drobnym rastrze wyprowadzeń mogą pojawić się zwarcia. Po usunięciu ewentualnych problemów z zasilaniem należy sprawdzić za pomocą oscyloskopu obecność sygnałów zegarowych o częstotliwości 32768 Hz na wyprowadzeniu XOUT32 (pin 35) oraz 18,432 MHz na wyprowadzeniu XOUT (31). Następnie należy sprawdzić obecność sygnału zegarowego w układzie STE100p na wyprowadzeniu X2 (11), którego częstotliwość powinna wynosić 25 MHz. Jeżeli wszystkie sygnały zegarowe są poprawne, możemy uznać, że wstępne uruchomienie płytki powiodło się i możemy przystąpić do dalszego procesu uruchamiania, który będzie polegał na wgraniu do pamięci BIOS ARMputera bootloadera inicjalizującego oraz bootloadera głównego U-BOOT. Wszystkie programy niezbędne do ożywienia naszej płytki możemy pobrać ze strony <http://bryndza.ep.com.pl> z działu *Projekty->ARMputer 9200*. Należy pobrać stamtąd pliki: *loader.bin*, *u-boot.bin*, *rm9200load.tgz* oraz *nfsroot-boof.tgz*. Istnieją dwie metody wgrania bootloadera:

- możemy wykorzystać specjalny program *rm9200load* dla systemu Linux lub
- standardowy program terminalowy, na przykład *minicom*, czy *hyperterminal*.

W niniejszym opisie skupimy się na rozwiązaniu z wykorzystaniem programu *rm9200load*. Program ten powinien również się dać skompilować w środowisku Cygwin dla Windows, niemniej jednak nigdy ta opcja nie była testowana przez autora.

Aby zaprogramować pamięć Flash ARMputera należy dołączyć port szeregowy DBG (J7) do portu szeregowego komputera za pomocą kabelka krosowego. Następnie należy uruchomić dowolny program terminalowy na przykład *Minicom*, czy *Hyperterminal* i skonfigurować połączenie poprzez port szeregowy z prędkością 115200 bodów, 8 bitów danych, 1 bit stopu, z wyłączoną kontrolą przepływu.

Po ustanowieniu połączenia i wciśnięciu klawisza RESET

w ARMputerze, na terminalu powinna być widoczna pojawiająca się w odstępach sekundowych litera C, co świadczy że procesor RM9200 rozpoczął wykonywanie programu ładującego znajdującego się w jego wewnętrznej pamięci ROM i oczekuje na przesłanie programu w protokole X-Modem. Po upewnieniu się, że procesor i interfejs portu szeregowego działa prawidłowo opuszczamy program terminalowy, a następnie rozpakowujemy oraz kompilujemy program *rm9200load*.

W tym celu w powłoce wydajemy następujące polecenia:

```
tar xvfz rm9200load.tgz
cd rm9200load
make
```

```
make install
```

Efektom będzie rozpakowanie i zainstalowanie w systemie programu umożliwiającego wygodne wgrywanie programów do pamięci BIOS. Po zainstalowaniu programu *rm9200load* należy wgrać pierwszy loader (*loader.bin*) do wewnętrznej pamięci RAM procesora, za pomocą polecenia *rm9200load /dev/ttyS0 loader.bin*. Gdy wgrywanie programu zakończy się powodzeniem, należy uruchomić program terminalowy (115200/n/8/1), a następnie wcisnąć dowolny klawisz. Na ekranie powinno ukazać się menu pierwszego bootloadera:

```
Darrell's loader – Thanks to the
u-boot project
```

```
Version 1.0. Build Dec
```

```
8 2007 20:06:15
```

```
Modified to BOFF board by Lucjan
```

```
Bryndza <lucjan.bryndza@ep.com.pl>
```

```
DRAM:32MB
```

```
1: Upload Darrell's loader to Dataflash
```

```
2: Upload u-boot to Dataflash
```

```
3: Upload Kernel to Dataflash
```

```
4: Start u-boot
```

```
5: Erase bootsector in flash
```

```
6: Memory test
```

Ponieważ jest to pierwsze uruchomienie bootloadera, należy wcisnąć klawisz 6, co spowoduje uruchomienie testu pamięci SDRAM. Jeżeli pamięć działa prawidłowo, po chwili na ekranie pojawi się następujący komunikat:

```
Testing RAM, Detected 32MB ==>
OK
```

Jeżeli pamięć działa prawidłowo, przystępujemy do wgrania pierwszego loadera do pamięci DataFlash. W tym celu w terminalu wciskamy klawisz 1 (*Upload Darrell's loader*

List. 1.

```
Dataflash read successful: Starting U-boot
U-Boot 1.3.1 (Dec 14 2007 - 15:46:45)
DRAM: 32 MB
Atmel: Flash: 0 kB
DataFlash:AT45DB161
Nb pages: 4096
Page Size: 528
Size= 2162688 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C000317F (RO) Darrell loader
Area 1: C0003180 to C001F73F (RO) U-boot
Area 2: C001F740 to C002183F Environment
Area 3: C0021840 to C01FFFFFF Kernel
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
boff>
```

to Dataflash). W oknie terminala powinna pojawiać się w odstępach sekundowych litera C. Następnie opuszczamy terminal i w powłoce wywołujemy program ładujący następująco: `rm9200load /dev/ttyS0 loader.bin`. Teraz należy uruchomić program terminalowy i wcisnąć RESET na płytce. Na terminalu zamiast literki C powinno pojawić się menu pierwszego loadera uruchamiającego się z pamięci DataFlash. Kolejną czynnością, jaką musimy wykonać jest zaprogramowanie drugiego bootladera U-BOOT (plik `u-boot.bin`). W tym celu w powłoce wydajemy polecenie: `rm9200load /dev/ttyS0 u-boot.bin 2`, a następnie na płytce wciskamy klawisz RESET, co powoduje rozpoczęcie procesu programowania pamięci Flash bootladerem U-BOOT. Prawdopodobność programowania możemy zweryfikować uruchamiając terminal i wciskając klawisz RESET na płytce. Na terminalu powinna pokazać się wówczas linia komend głównego bootladera U-BOOT (**list. 1**).

Po wgraniu UBOOT-a musimy ustawić zmienne środowiskowe, które są niezbędne do uruchomienia Linuxa. W tym celu w terminalu wydajemy następujące polecenia:

```
setenv bootdelay 3
```

```
setenv ethaddr 00:01:02:03:04:05
```

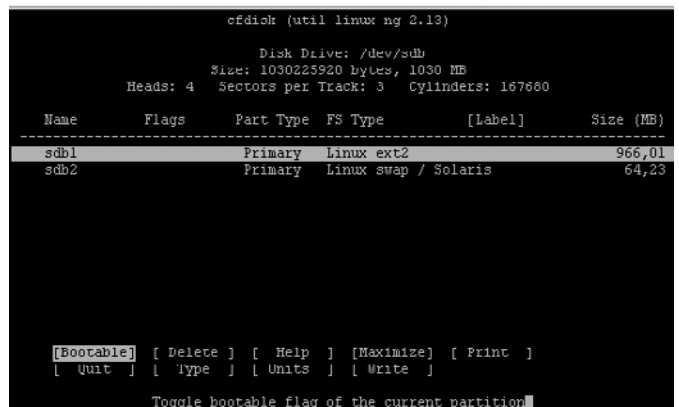
```
setenv bootfile vmlinux.
img setenv nfsargs mem=32M
console=ttyS0,115200n8 root=/dev/
nfs nfsroot=192.168.16.101:/var/nfs,
timeo=200,retrans=500 ip=:::eth0:
on setenv ipaddr 192.168.16.222
setenv serverip 192.168.16.101
setenv bootargs mem=32M
rootfstype=ext2 root=/dev/mmcblk0p1
console=ttyS0,115200n8 rootdelay=1
init=/sbin/init setenv bootcmd mm-
cinit; ext2load mmc 0 0x21000000
/boot/at91boff -linux.bin; bootm
0x21000000
saveenv
```

W zależności od własnych preferencji możemy wpisać odpowiedni adres sprzętowy MAC, adres IP płytki, czy adres serwera NFS dostosowując go do konfiguracji sieci. Następnie należy sprawdzić poprawność działania sieci w uboocie. W tym celu w wierszu polecenia loadera wydajemy polecenie `dhcp`,

co po chwili powinno zaowocować wyświetleniem adresu IP przydzielonym przez serwer DHCP. Po zakończeniu konfiguracji U-BOOT-a, ostatnią czynnością, jaką musimy zrobić jest przygotowanie partycji na karcie MMC oraz instalacja systemu `boff-linux`. Aby zainstalować system na karcie należy podłączyć do komputera zewnętrzny czytnik kart MMC (lub wykorzystać czytnik wbudowany), włożyć do niego kartę MMC, a następnie określić identyfikator urządzenia pod

jącą partycję należy usunąć nakierowując strzałkami na tą partycję, a następnie wybierając opcję `[Delete]`. Kolejną czynnością jest utworzenie partycji głównej EXT2 oraz partycji wymiany LINUX SWAP. W tym celu w programie wybieramy opcję `Create New`, podajemy rozmiar partycji tak, aby zostało 64 MB wolnego miejsca i wciskamy przycisk OK. Następnie ponownie wydajemy opcję `Create New` i na pozostałym miejscu tworzymy kolejną partycję. Następnie musimy zmienić typ ostatniej partycji na LINUX SWAP. W tym celu wybieramy opcję `Type`, zatwierdzamy domyślnie 82 (Linux SWAP) i wciskamy `Enter`. Teraz należy zapisać układ tablicy na kartę, co możemy zrobić za pomocą opcji `Write`. Prawdopodobnie stworzony układ partycji na karcie powinien wyglądać tak jak na **rys. 10**.

Jeżeli układ partycji jest prawidłowy, przystępujemy do sformatowania partycji głównej oraz partycji wymiany wydając następujące polecenia w powłoce:



```
cfdisk (util linux ng 2.13)

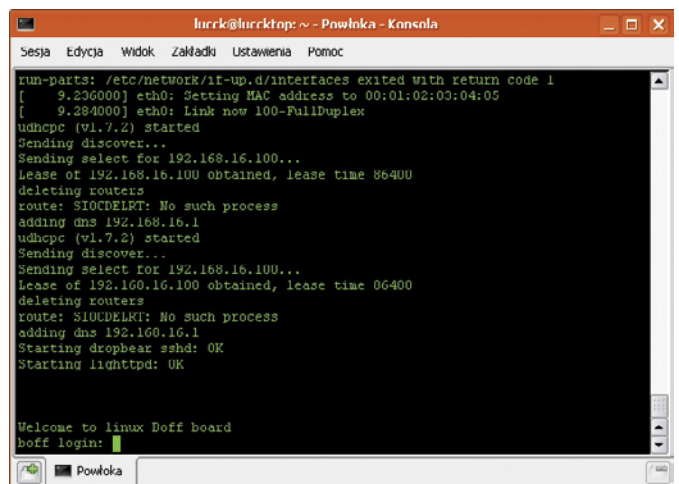
Disk Drive: /dev/sdb
Size: 1030225920 bytes, 1030 MB
Heads: 4 Sectors per Track: 3 Cylinders: 167680

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sdb1      Primary    Linux     linux ext2   [Label]      965,01
sdb2      Primary    Linux swap / Solaris

[Bootable] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ]   [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition
```

Rys. 10. Układ partycji na karcie utworzony przez użytkownika



```
lurck@lurcktop: ~ - Powloka - Konsola

Sesja Edycja Widok Zakladki Ustawienia Pomoc

run-parts: /etc/network/if-up.d/interfaces exited with return code 1
[  9.236000] eth0: Setting MAC address to 00:01:02:03:04:05
[  9.284000] eth0: Link now 100-FullDuplex
udhcpd (v1.7.2) started
Sending discover...
Sending select for 192.168.16.100...
Lease of 192.168.16.100 obtained, lease time 86400
deleting routers
route: SIOCDLMT: No such process
adding dns 192.168.16.1
udhcpd (v1.7.2) started
Sending discover...
Sending select for 192.168.16.100...
Lease of 192.168.16.100 obtained, lease time 06400
deleting routers
route: SIOCDLMT: No such process
adding dns 192.168.16.1
Starting dropbear sshd: OK
Starting lighttpd: OK

Welcome to linux boff board
boff login: █
```

Rys. 11. Widok procesów uruchamiającego się systemu

WYKAZ ELEMENTÓW

Rezystory

R1...R3, R5: 1 kΩ

R4, R12, R13, R33...R42, R55...R58,
R61, R65, R67, R69, R81: 0 Ω

R6...R11, R20, R21: 51 Ω

R14...R18, R22, R43...R45, R73, R74:
100 Ω

R19: 5 kΩ

R23, R24: 75 Ω

R25...R28: 33 Ω

R29...R31: 15 kΩ

R32: 22 kΩ

R46, R68, R72, R75...R80, R84...R86:
10 kΩR47...R54, R59, R60, R62...R64, R66,
R70, R71: – zworki konfiguracyjne
0 Ω

R82,R83: 47 kΩ

Kondensatory

C1...C51, C60...C70, C75...C82, C85,
C86, C91, C95...C99: 100 nF

C52, C58: 470 pF

C53, C59: 4,7 nF

C54...C57: 10 pF

C71,C100...C116: 100 μF/16 V

C72...C74: 1 nF/2 kV

C83, C84: 22 pF

C87, C88: 15 pF

C89, C90: 47 pF

C92...C94, C117: 1000 μF>/16 V

Półprzewodniki

D1: LED czerwona

D2,D7: LED zielona

D3: LED pomarańczowa

D4: LED niebieska

D5, D8: LED żółta

D6: BAT54S

U1, U2: K4S281632

U3: MAX485CSA

U4...U6: MAX3232

U7: AT45DB161

U8: STE100P

U9: AT91RM920

U10: MCP100T-3

U11: LF33

U12: LF18

Inne

F1: bezpiecznik MFRG500

J1: złącze karty MMC

J2...J5: łączówka IDC40

J6...J8: gniazdo DB9_M

J9: złącze RJ-45

J10: gniazdo USB-A

J11: gniazdo USB-B

J12: goldpin

J13: łączówka IDC20

J14: wylącznik

L1...L5: dławik BLM21PG22 Murata

S1: TACT SWIT

TR1: transformator LAN H1260

X1: rezonator kwarcowy 18,432 MHz

X2: rezonator kwarcowy 32,768 kHz

X3: rezonator kwarcowy 25 MHz

```
mkfs.ext2 /dev/sdb1
```

```
mkswap /deb/sdb2
```

```
sync
```

Gdy już mamy przygotowany układ partycji, należy wyjąć, a następnie ponownie włożyć kartę pamięci do czytnika. Po ponownym włożeniu karty powinien pojawić się monit systemu o zamontowanie nowej partycji, na co się zgadzamy. Karta pamięci będzie najprawdopodobniej widoczna w ścieżce `/media/disk` (dystrybucja `ubuntu`). Ostatnią czynnością, jaką należy zrobić jest rozpakowanie głównego systemu plików Linuxa na partycję główną wydając polecenie: `sudo tar xvzf rootfs.arm.tar.gz -C /media/disk`.

Po rozpakowaniu systemu plików należy odmontować kartę pamięci za pomocą polecenia `umount /media/disk` lub za pomocą narzędzia graficznego, a następnie wyjąć tak przygotowaną kartę pamięci z czytnika. To już w zasadzie wszystkie czynności niezbędne do uruchomienia `boff-linuxa`, możemy zatem do płytki ARMputera włożyć kartę pamięci MMC, podłączyć kabel do gniazdka sieciowego, dołączyć płytke przez port szeregowy do komputera PC z uruchomionym programem terminalowym, a następnie włączyć napięcie zasilające. W oknie terminala powinniśmy zaobserwować procesy uruchamiającego się systemu (rys. 11).

Co dalej?

Po wykonaniu czynności opisanych w poprzednim podpunkcie powinniśmy mieć w pełni uruchomiony i działający system Linux na naszym ARMputerze. Możemy zatem zalogować się do niego zarówno za pomocą terminala poprzez port szeregowy, jak i poprzez sieć za pomocą protokołu SSH, wydając polecenie `ssh root@adres_ip_armputera`. Jako login podajemy nazwę `root`, a jako hasło `zaqwsx`. Możemy uruchamiać dowolne zainstalowane

programy dostępne w dystrybucji na przykład popularny manager plików `mc` przypominający Norton Commandera. W dystrybucji zainstalowano także serwer WWW `lighttpd`, którego główny katalog z dokumentami znajduje się w pliku `/var/www`. Nie jesteśmy oczywiście ograniczeni tylko do programów, które zostały domyślnie zainstalowane. Ponieważ mamy do dyspozycji `cross-compiler`, bez problemu możemy instalować ze źródeł różne programy lub tworzyć własne, co będzie tematem kolejnych artykułów.

Lucjan Bryndza, EP
lucjan.bryndza@ep.com.pl

Chciałbym podziękować współtwórcom ARMputera BF100:

Wojtkowi Zieleźnemu – za opracowanie schematu oraz dobór elementów do projektu,

Jerzemu Jurkowskiemu – za projekt PCB, w który włożył wiele godzin pracy.

R	E	K	L	A	M	A
ROBOKITY VELLEMAN						
		BOTBALL Kod handlowy: KSR9 Cena: 161 zł				
					HEXAPOD Kod handlowy: KSR3 Cena: 78 zł	
Zestawy do samodzielnego montażu						
www.sklep.avt.pl					tel. 022 257 84 50	