

# Emulacja Raspberry Pi

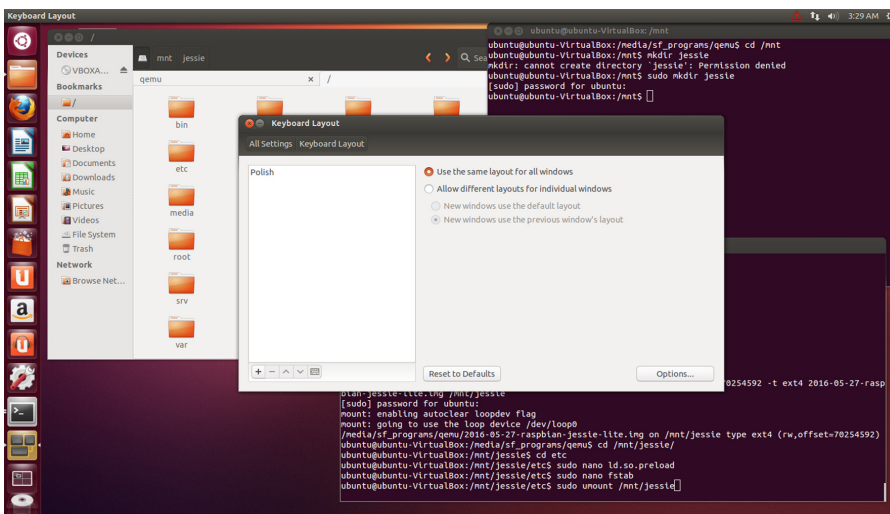
**Większość projektów opartych na minikomputerach, a opisywanych w „Elektronice Praktycznej”, bazuje na Raspberry Pi. Komputer ten bardzo łatwo dostać, a jego niewielka cena sprawia, że jest w zasięgu nawet początkujących elektroników. Omawiane projekty można jednak – przynajmniej częściowo – sprawdzić jeszcze niższym kosztem, a nawet bez potrzeby dotykania jakichkolwiek płytek drukowanych. Wystarczy skorzystać z emulatora Raspberry Pi.**

Zakup Raspberry Pi nie stanowi problemu. Aktualnie sprzęt ten jest szybko dostępny, a 200 zł za pełnoprawny, miniaturowy komputer to naprawdę niedużo. Bywa jednak tak, że czy to ze względów logistycznych, czy nawet z wygody, a być może z powodów finansowych, zakup i obsługa fizycznego RPI nie wydają się dobrym rozwiązaniem. Alternatywą będzie użycie darmowego emulatora do stworzenia wirtualnego środowiska, które pod wieloma względami będzie identyczne z Raspberry Pi. W niniejszym artykule pokazujemy, jak to zrobić i jak pokonać cały szereg trudności, wiążących się z tą czynnością.

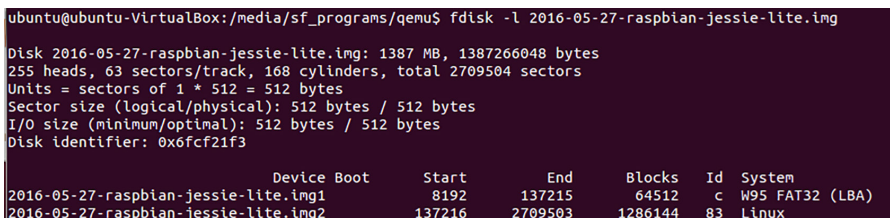
## Kłopoty z ARM-em

Podstawowym problemem, związanym z uruchamianiem wirtualnego Raspberry Pi, jest obsługa procesora tego minikomputera. Układy Broadcom w RPI mają rdzenie ARM, które nie są powszechnie implementowane przez większość popularnych programów do tworzenia maszyn wirtualnych. Dlatego wybraliśmy Qemu – bezpłatny program, wspierający m.in. 32- i 64-bitowe architektury x86, PowerPC, SPARC i m.in. ARM. O ile nie sprawdza się dobrze do uruchamiania wirtualnych systemów MS Windows oraz wcale nie jest wygodny w obsłudze, o tyle jest w stanie emulować jednostkę ARM1176 w stopniu wystarczającym do uruchomienia Raspbiana.

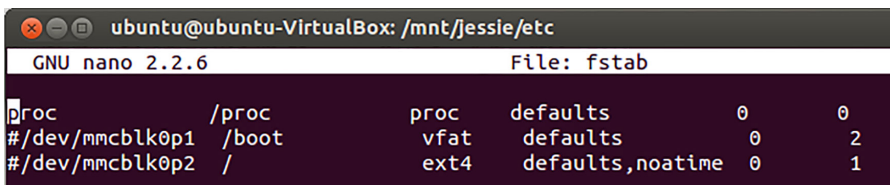
Do uruchomienia emulatora potrzebny jest jego główny program oraz obraz dysku z systemem, który ma być włączony i plik jądra Linuksa, adekwatny do danego systemu. Strona domowa Qemu to <http://qemu.org>, ale przekierowuje ona bezpośrednio na materiały wiki, podczas gdy skompilowane pliki binarne pod MS Windows (a zakładamy, że taki właśnie system operacyjny jest typowym środowiskiem pracy inżyniera elektronika) można pobrać spod adresu <https://qemu.weilnetz.de/>. Zarówno wersja 32-, jak i 64-bitowa będzie wystarczająca. My sięgnęliśmy po kompilację 2.7.0-rc3 z 17 sierpnia 2016 r. (<http://tinyurl.com/h9sbvsvf>), czyli najnowszą w momencie pisania tego artykułu. W sieci można znaleźć informacje, że obsługa Raspberry Pi przestaje działać w nowszych wersjach tego



Rysunek 1. Zmiana ustawień klawiatury w wirtualnym Ubuntu na VirtualBoxie



Rysunek 2. Wynik wywołania polecenia fdisk pozwala przyjrzeć się układowi partycji w obrazie dysku



Rysunek 3. Zmiana zawartości pliku fstab w dystrybucji Raspbiana Jessie konieczna, by móc uruchomić system

oprogramowania, ale w praktyce wieści te wynikają z niezrozumienia różnych trudności związanych z uruchomieniem nowego Raspbiana na Qemu, a my w niniejszym artykule je rozwiązujemy. Stąd można przyjąć, że korzystanie z nowszych wersji Qemu nie powinno pogorszyć sytuacji, a może nawet ją ułatwić.

Najnowszy obraz dysku systemu operacyjnego Raspberry Pi można pobrać z oficjalnej strony fundacji RPI (<https://goo.gl/8h18uP>). Aktualnie jest to dystrybucja Jessie, bazująca na analogicznie nazwanym Debianie (Debian 8) i udostępniona 27 maja 2016

roku. Dostępna jest też „lżejsza” postać tej dystrybucji – Raspbian Jessie Lite. Obie one pracują z jądrem w wersji 4.4, co warto zapamiętać. Gdyby w przyszłości aktualne wersje zostały zastąpione nowszymi, niekompatybilnymi z opisaną tutaj metodą, archiwalne obrazy dystrybucji można pobrać pod adresem <http://goo.gl/1ZQQUM> w przypadku pełnego Raspbiana lub <http://goo.gl/VvPDvj>, gdy potrzebny jest Raspbian Lite.

W tym momencie musimy zwrócić uwagę na jeden istotny szczegół, a mianowicie na dużą różnicę pomiędzy Raspbianem Jessie a poprzednią generacją tego systemu,

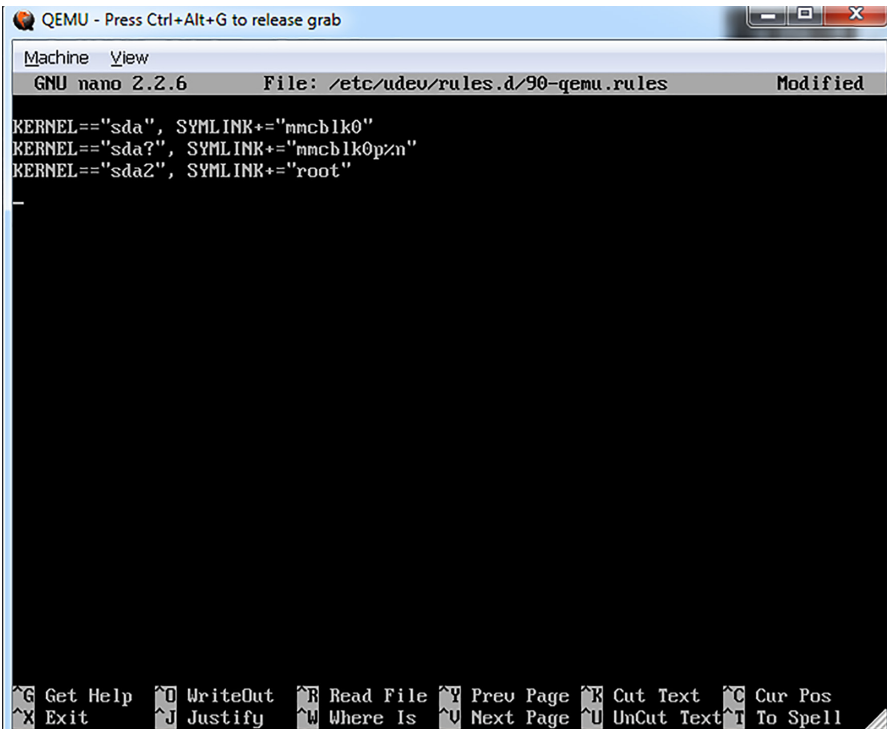
czyli Raspbianem Wheezy. Ten starszy bazuje na zupełnie innej wersji jądra i jest znacznie mniej problematyczny do uruchomienia. Jeśli więc do zaplanowanych testów wystarczy starsza wersja systemu operacyjnego, można sięgnąć po Raspbiana Wheezy np. z 16 lutego 2015 r. i w ten sposób uniknąć kilku dodatkowych kroków. Dowolna wersja Raspbiana Jessie będzie natomiast wymagała dodatkowych modyfikacji. Opisujemy je poniżej.

## Kuracja Jessiego

Próba uruchomienia wirtualnego Raspberry Pi z Raspbianem Jessie powoduje błąd na poziomie jądra systemu (kernel panic), pojawiający się w chwili, gdy podczas startu system próbuje załadować treść z karty pamięci. Jessie domyślnie wczytuje bibliotekę odpowiedzialną za mapowanie pamięci, która w przypadku niedoskonałego emulatora, jakim jest Qemu i w obliczu braku realnej karty SD, powoduje problem. W związku z powyższym,

by pomyślnie uruchomić nową wersję Raspbiana, konieczne staje się wyłączenie tego elementu w procesie uruchamiania. Niestety, stosowna operacja jest dosyć skomplikowana.

Obraz dysku dla Raspberry Pi to plik .img (spakowany zipem), w formacie raw z partycjami ext4. Typowe narzędzia pod Windows dosyć słabo radzą sobie z jakkolwiek edycją tego typu plików, a jeśli już, to ich instalacja dosyć mocno ingeruje w system. W praktyce, by skutecznie i pewnie wprowadzić zmiany w procedurze uruchamiania systemu ukrytego w obrazie dysku, należy posłużyć się Linuxem. Ze względu na użycie systemu plików ext4 konieczne jest wybranie dosyć nowej dystrybucji. Dlatego zamiast polecić pobranie prostego, miniaturowego Linuxa i włączenie go z użyciem i tak już zainstalowanego Qemu, sugerujemy inne rozwiązanie. Zakładamy przy tym, że czytelnik nie ma do dyspozycji realnego komputera z Linuxem i nie chce restartować swojego, by uruchomić go z nośnika flash lub płyty z systemem. Alternatywną (i dosyć uniwersalną) metodą jest użycie gotowego obrazu systemu wirtualnego. Osoby, które preferują inny sposób uruchomienia Linuxa mogą pominąć następną dział.

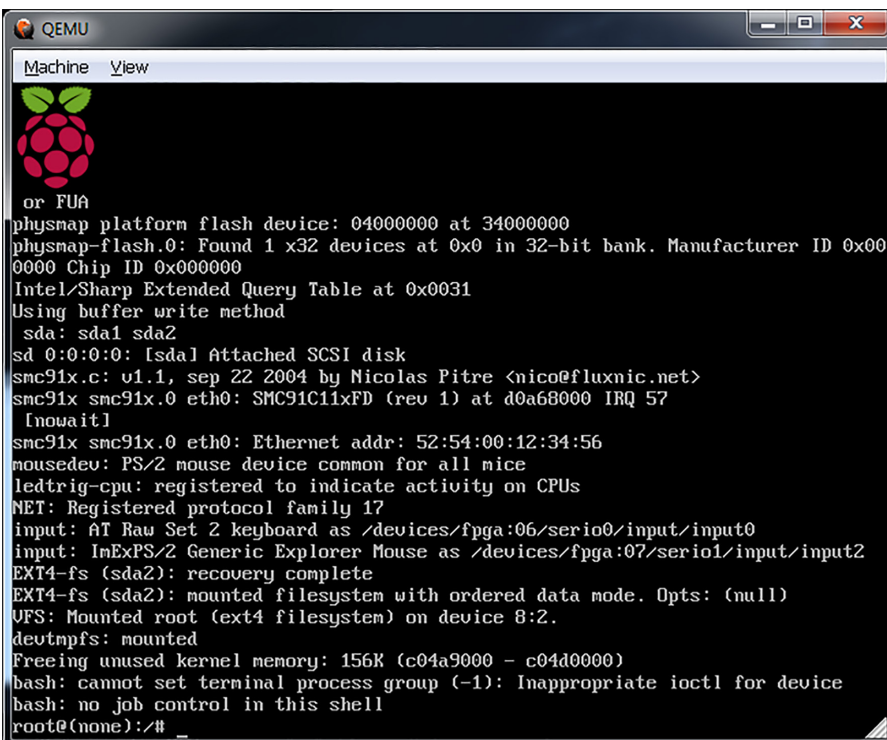


```

QEMU - Press Ctrl+Alt+G to release grab
Machine View
GNU nano 2.2.6 File: /etc/udev/rules.d/90-qemu.rules Modified
KERNEL=="sda", SYMLINK+="mncblk0"
KERNEL=="sda?", SYMLINK+="mncblk0pzn"
KERNEL=="sda2", SYMLINK+="root"
-
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^X Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^I To Spell

```

Rysunek 4. Treść pliku tworzonego, by nośniki emulowane przez Qemu jako podłączone przez USB były widziane jako karty SD



```

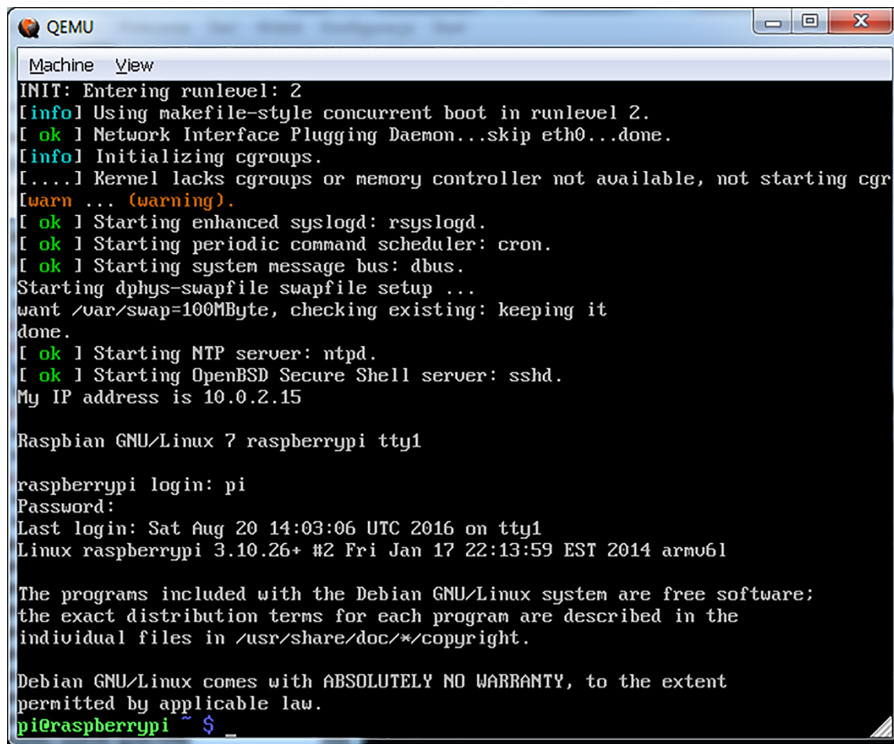
QEMU
Machine View
or FUA
physmap platform flash device: 04000000 at 34000000
physmap-flash.0: Found 1 x32 devices at 0x0 in 32-bit bank. Manufacturer ID 0x00
0000 Chip ID 0x000000
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
sda: sda1 sda2
sd 0:0:0:0: [sda] Attached SCSI disk
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
smc91x smc91x.0 eth0: SMC91C11xFD (rev 1) at d0a68000 IRQ 57
[nowait]
smc91x smc91x.0 eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
ledtrig-cpu: registered to indicate activity on CPUs
NET: Registered protocol family 17
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2
EXT4-fs (sda2): recovery complete
EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) on device 8:2.
devtmpfs: mounted
Freeing unused kernel memory: 156K (c04a9000 - c04d0000)
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/#

```

Rysunek 5. Qemu uruchomiony z jądrem 4.4 i podmontowanym obrazem Raspbiana Jessie, ale jeszcze przed uruchomieniem samego Raspbiana

## Instalacja wirtualnego Linuxa

Obraz względnie nowego, wirtualnego systemu z rodziny Linuxów można pobrać spod adresu <https://virtualboxes.org/images/>. Są tam pliki dla bezpłatnego i wygodnego w użyciu środowiska VirtualBox, który wielu czytelników może mieć już zainstalowany, a jeśli nie, to z łatwością poradzi sobie z jego instalacją. Sam Oracle VirtualBox można znaleźć pod adresem: <https://www.virtualbox.org/>. My wybraliśmy Ubuntu, a więc podstronę <https://virtualboxes.org/images/ubuntu/>. Konkretnie sięgnęliśmy po plik z Ubuntu 12.10 na komputerze x86. Jest on dostępny pod adresem <http://goo.gl/tSHnZ1> i ma 613 MB. Należy go pobrać i rozpakować, po czym spośród zawartości spróbować otworzyć ten z rozszerzeniem .vbox. Spowoduje to uruchomienie VirtualBoxa z Ubuntu, do którego już po chwili będzie można się zalogować. Hasło użytkownika Ubuntu to w tym przypadku „reverse”. Następnie w menu Devices -> Shared Folders -> Shared Folders Settings... należy dodać folder do grupy Machine Folders. Wybrany folder musi zawierać plik obrazu Raspbiana Jessie, który chcemy poddać kuracji. Warto też zaznaczyć opcje „Auto-Mount” i „Make Permanent”, ale na pewno nie należy zaznaczać „Read-only” – potrzebujemy bowiem dokonać zmian w pliku, do którego dostęp otrzymujemy. Przy okazji można też zwiększyć ilość pamięci wideo i RAM, przydzielonej wirtualnej maszynie oraz warto skorzystać z opcji Devices -> Insert Guest Additions CD Image..., by mieć pewność, że integracja pomiędzy systemem plików rzeczywistego komputera a maszyny



Rysunek 6. Pomyślnie uruchomiony Raspbian Wheezy, oparty na Debianie 7

wirtualnej będzie działać z dowolną wersją systemu hosta.

Pewnym utrudnieniem jest fakt, że wspomniany obraz domyślnie ma ustawioną włoską klawiaturę, której mapowanie znacznie różni się od polskiej/amerykańskiej. Można to łatwo zmienić, klikając na ikonkę włącznika/ustawień w prawym górnym rogu ekranu i wybór opcji System Settings.... W dziale Hardware znajduje się ikona Keyboard, po której naciśnięciu przechodzi się do okna z linkiem „Layout Settings”. Po jego kliknięciu należy nacisnąć znak plusa w kolejnym oknie, wybrać stosowną klawiaturę (np. polską) i zatwierdzić wybór (rysunek 1). Można też usunąć klawiaturę włoską, klikając na nią i naciskając znak minusa.

W tym momencie warto dla pewności zrestartować system wirtualny, po czym konieczne staje się dodanie aktualnego użytkownika (ubuntu) do grupy vboxsf, by mieć dostęp do folderu współdzielonego z rzeczywistego komputera, w którym znajduje się obraz Raspbiana Jessie. W tym celu należy w terminalu maszyny wirtualnej Ubuntu wykonać polecenie:

```
sudo adduser ubuntu vboxsf
```

i w razie potrzeby podać hasło dla potwierdzenia. Po tej operacji konieczny jest ponowny restart, a po nim w systemie plików, pod adresem /media, będzie widoczny podmontowany katalog współdzielony z obrazem Raspbiana.

### Edycja zawartości obrazu Jessiego

Następnym krokiem jest przygotowanie miejsca do zamontowania partycji z obrazu Raspbiana w uruchomionym Linuksie, np.

zawiera dwie partycje, z czego druga (2016-05-27-raspbian-jessie-lite.img2) rozpoczyna się od sektora 137216. W przypadku innych obrazów wartość ta będzie się nieco różnić. Natomiast wielkość sektora to 512 bajtów. Oznacza to, że druga partycja rozpoczyna się od 70254592 bajtu (137216×512). Mając tę informację, możemy ją podmontować, pomimo że znajduje się wewnątrz pliku obrazu. Należy użyć polecenia **mount** z uprawnieniami roota:

```
sudo mount -v -o offset=70254592 -t ext4 2016-05-27-raspbian-jessie-lite.img /mnt/jessie
```

Naturalnie dokładna wielkość offsetu i nazwa pliku będą inne dla każdego obrazu. Jeśli wszystkie dane zostały poprawnie obliczone, w katalogu /mnt/jessie będzie można znaleźć zawartość drugiej partycji obrazu, a więc tej, na której znajdują się wszystkie ustawienia systemowe.

Problematiczne komendy wydawane podczas uruchamiania Raspbiana Jessie należy wyomentować. Należy przejść do katalogu /mnt/jessie (jeśli tam podmontowaliśmy partycję):

```
cd /mnt/jessie
```

a następnie wejść do katalogu z ustawieniami:

```
cd etc
```

Pierwszy plik do edycji to **ld.so.preload**. Można skorzystać z edytora **nano**, ale konieczne z uprawnieniami roota:

```
sudo nano ld.so.preload
```

Należy wyomentować wszelkie znajdujące się w środku linijki, poprzez dostawienie znaku „#” na ich początku. W naszym przypadku była tam jedna linijka:

```
/usr/lib/arm-linux-gnueabi/hf/libarmmem.so
```

poprzez utworzenie podkatalogu **jessie** w katalogu /mnt. W tym celu należy wywołać w terminalu polecenia:

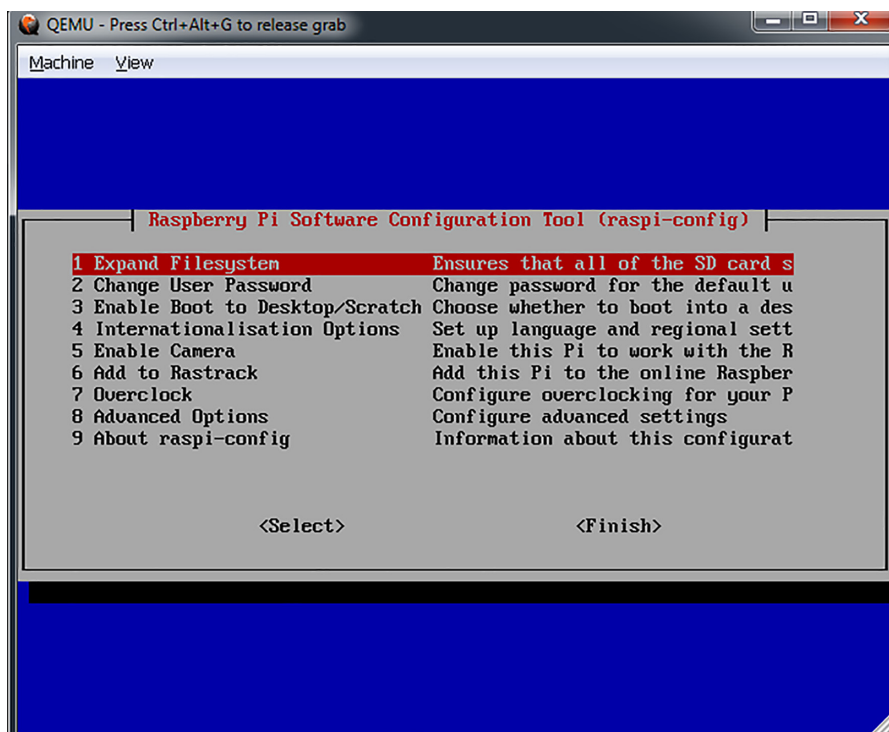
```
cd \mnt
```

```
sudo mkdir jessie
```

Poleceniem **fdisk** należy przeanalizować strukturę pliku z obrazem. Na przykład:

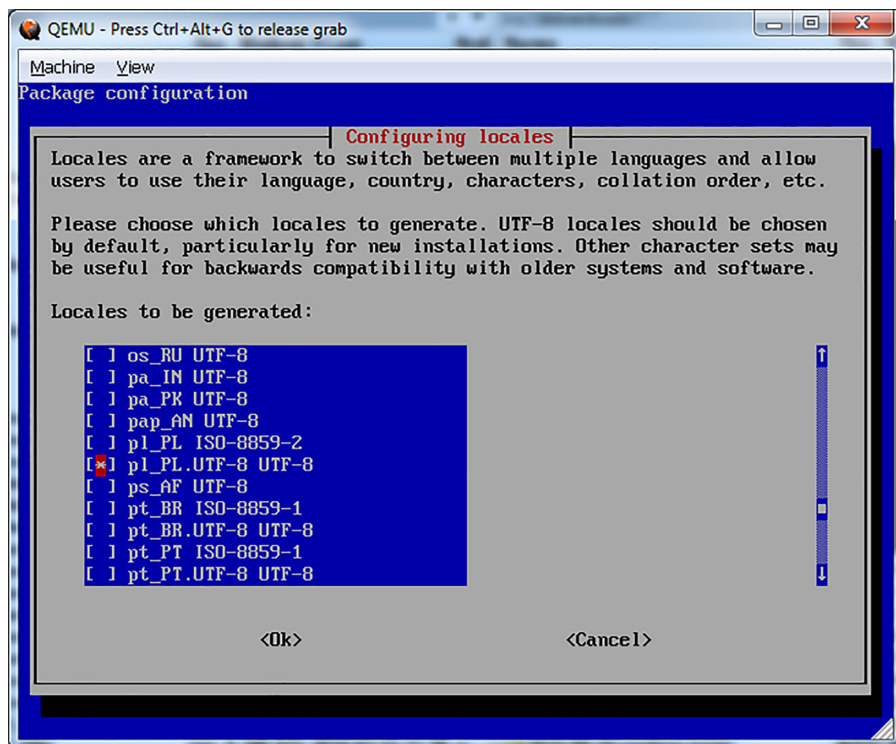
```
fdisk -l 2016-05-27-raspbian-jessie-lite.img
```

W przypadku Raspbiana Jessie Lite z 27 maja 2016 wygląda ona tak, jak na **rysunku 2**. Jak widać, plik



Rysunek 7. Menu raspi-config w wersji z Raspbiana Wheezy





Rysunek 8. Ustawienia lokalne Raspbiana

Z edytora wychodzimy kombinacją CTRL+x, potwierdzając zapis zmian.

Drugi plik do edycji to **fstab**:

```
sudo nano fstab
```

w którym należy wycommentować wszystkie linie, zawierające ciąg o fragmencie „mmcblk” (rysunek 3). Chodzi o to, by system nie wieształ się w momencie próby podmontowania zawartości karty pamięci, gdyż w systemie jest ona widziana jako pendrive. Po zapisaniu zmian należy wyjść z podmontowanych podkatalogów np. poprzez:

```
cd \
```

po czym odmontować partycję:

```
sudo umount /mnt/jessie
```

W efekcie na komputerze realnym mamy zmodyfikowany obraz Raspbiana Jessie, który nie będzie powodował problemów przy uruchamianiu w Qemu.

## Uruchomienie wirtualnego Raspberry Pi

Mając obraz kompatybilny z emulatorem, czy to domyślną wersję Wheezy, czy zmodyfikowaną Jessie, można przejść do jego uruchamiania. Konieczne jest tylko jeszcze pobranie odpowiedniego pliku jądra systemu.

Różne wersje pliku jądra, przygotowanego na potrzeby Qemu i Raspberry Pi, można pobrać spod adresu: <https://goo.gl/E3Wn71>. Dla Wheezy'ego wystarczy wersja 3.10.25, ale dla Jessie potrzebny będzie nowszy. W momencie pisania tego tekstu, najnowsza opublikowana tam wersja to 4.4.12 i z niej właśnie korzystamy. Po jej pobraniu należy wgrać ją do katalogu, w którym zainstalowany został Qemu. Do tego samego katalogu warto też skopiować przygotowany obraz Raspbiana.

Uruchomienie Raspbiana na PC z Windowsem wymaga podania do Qemu szeregu parametrów oraz wstępnej konfiguracji. Na początek zalecaną składnią będzie:

```
qemu-system-arm.exe -kernel
kernel-qemu -cpu arm1176 -m
256 -M versatilepb -no-reboot
-serial stdio -append „root=/
dev/sda2 panic=1 rootfstype=ext4
rw init=/bin/bash” -hda
2015-02-16-raspbian-wheezy.img
```

Opcja `-kernel` wskazuje plik jądra Linuksa. Opcja `-cpu` podaje, która wersja procesora ma być symulowana. Opcja `-m` informuje, że emulowana ma być maszyna z 256 MB pamięci RAM. To mało, ale niestety Qemu działa nieco wadliwie w przypadku emulacji ARMA gdy podajemy większą wartość, system prawdopodobnie się nie uruchomi. Parametr `-hda` wskazuje plik z obrazem, który ma być potraktowany jako dysk podłączony do wirtualnego komputera, czyli w przypadku RPI – zawartość karty SD.

Dla systemu Raspbian Jessie linia polecenia może wyglądać następująco:

```
qemu-system-arm.exe -kernel
kernel-qemu-4.4.12-jessie -cpu
arm1176 -m 256 -M versatilepb
-no-reboot -serial stdio
-append „root=/dev/sda2 panic=1
rootfstype=ext4 rw init=/bin/bash”
-hda 2016-05-27-raspbian-jessie-
lite.img
```

Jednakże tak naprawdę nie jest to jeszcze uruchomienie prawdziwego Raspbiana, a jedynie środowiska linuksowego z dostępem do systemu plików z użyciem powłoki `/bin/bash`. (rysunek 5)

Dalsze operacje zależą od wybranej wersji Raspbiana. W przypadku Wheezy trzeba po włączeniu środowiska linuksowego w Qemu edytować plik `/etc/ld.so.preload`, podczas gdy w Jessie zostało to już zrobione wcześniej. Poleceniem:

```
sudo nano /etc/ld.so.preload
```

należy otworzyć plik i wycommentować jedyną tam linię tekstu.

Kolejną operacją trzeba wykonać zarówno we Wheezy, jak i Jessie. Należy utworzyć plik `/etc/udev/rules.d/90-qemu.rules`, który spowoduje utworzenie symbolicznych linków do urządzeń tak, by systemowi operacyjnemu wydawało się, że ma do czynienia z prawdziwą kartą MMC, podczas gdy nasz symulator widzi niektóre nośniki jako urządzenia USB. Dlatego konieczne staje się użycie polecenia:

```
nano /etc/udev/rules.d/90-qemu.
```

```
rules
```

i wpisanie treści:

```
KERNEL=="sda", SYMLINK+="mmcblk0"
K E R N E L = = " s d a ? " ,
SYMLINK+="mmcblk0p%n"
KERNEL=="sda2", SYMLINK+="root"
```

Powinno to wyglądać tak, jak na rysunku 4. Po zapisaniu pliku należy wyjść z emulatora poleceniem „exit”.

Mając tak przygotowany system, można już pokusić się o uruchomienie właściwego Raspbiana. W tym celu konieczne jest zmodyfikowanie parametrów wywołania Qemu. Dla Wheezy będzie to:

```
qemu-system-arm.exe -kernel
kernel-qemu -cpu arm1176 -m 256
-M versatilepb -no-reboot -serial
stdio -append „root=/dev/sda2
panic=1 rootfstype=ext4 rw” -hda
2015-02-16-raspbian-wheezy.img
a dla Jessie:
qemu-system-arm.exe -kernel
kernel-qemu-4.4.12-jessie -cpu
arm1176 -m 256 -M versatilepb
-no-reboot -serial stdio
-append „root=/dev/sda2 panic=1
rootfstype=ext4 rw” -hda
2016-05-27-raspbian-jessie-lite.
img
```

Jak widać, usunięta została opcja `init=/bin/bash`.

System powinien uruchomić się, pomimo wyświetlenia szeregu błędów. Istnieje też bardzo duża szansa, że zawartość pliku `/etc/udev/rules.d/90-qemu.rules` została wyczyszczona, stąd trzeba ją sprawdzić i ew. uzupełnić, tak jak na rysunku 4. Może się okazać, że tym razem wywołanie polecenia `nano` dla tego pliku będzie wymagało uprawnień roota (czyli wywołanie rozpoczynamy od `sudo`). Pojawia się tu pewne dodatkowe utrudnienie. Raspbian domyślnie uruchamia się z klawiaturą ustawioną na układ brytyjski, a więc inną niż w USA czy w Polsce. Cudzysłów potrzebny w modyfikowanym pliku `90-qemu`.

rules uzyskuje się na niej poprzez naciśnięcie SHIFT+2. Po zapisaniu zmian można ponownie wyłączyć system.

Kolejne uruchomienie emulatora da różne rezultaty, w zależności od wersji systemu. W przypadku Wheezy na ekranie pojawią się opcje konfiguracyjne Raspberry Pi, które należy po prostu zamknąć i wymusić ponowny restart systemu. Po nim będzie już można w pełni zalogować się do RPI z użyciem loginu „pi” i hasła „raspberrypi” (rysunek 6).

W przypadku Jessie nie będzie to tak dobrze działało, gdyż system nie jest w stanie automatycznie podmontować pierwszej partycji obrazu nośnika do katalogu /boot, co sprawia że polecenie **raspi-config** nie działa poprawnie. Partycję tę można jednak podmontować ręcznie, wpisując:

```
sudo mount /dev/mmcblk0p1 /boot
```

Po czym można wywołać ekran ustawień komendą:

```
sudo raspi-config
```

## Rozszerzenie systemu plików

Aby móc sensownie wykorzystywać wirtualne Raspberry Pi, wypada jeszcze sprawić, by na wirtualnej karcie pamięci, włożonej w ten wirtualny sprzęt, było odpowiednio dużo miejsca na programy. Operacja ta rozpoczyna się od powiększenia samego pliku obrazu systemu i w gruncie rzeczy można ją przeprowadzić już w chwilę po pobraniu pliku z obrazem. My w celu powiększenia pliku użyjemy narzędzia z pakietu Qemu. Służy temu polecenie:

```
qemu-img.exe resize
```

```
2015-02-16-raspbian-wheezy.img  
+5G
```

gdzie oczywiście **2015-02-16-raspbian-wheezy.img** to nazwa pliku z obrazem, a ostatni parametr określa, o ile rozmiar obrazu ma być zwiększony.

Następnie należy ponownie uruchomić emulator. Komendy po zalogowaniu będą zależeć od wersji systemu. We Wheezy należy jedynie stworzyć dowiązanie:

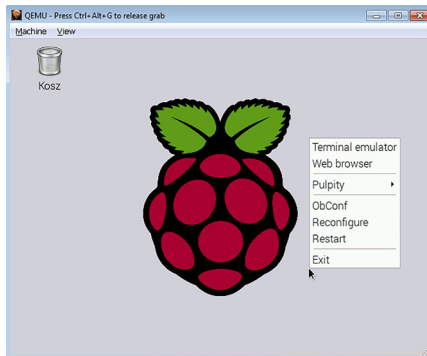
```
sudo ln -snf mmcblk0p2 /dev/root
```

W Jessie również wypada stworzyć to dowiązanie, ale dodatkowo trzeba także ponownie ręcznie podmontować /dev/mmcblk0p1 do katalogu /boot oraz zmodyfikować działanie programu **raspi-config**, gdyż domyślnie będzie on myślał, że nośnikiem jest pendrive, a nie karta pamięci i nie będzie umiał rozszerzyć systemu plików. W tym celu należy wywołać ciąg poleceń:

```
sudo bash
```

```
sed -E 's/mmcblk0p?/sda/' /usr/  
bin/raspi-config | bash
```

Gdyby się okazało, że system cały czas pracuje z brytyjską klawiaturą, w której znak „|” znajduje się pomiędzy lewym SHIFT-em a klawiszem „z”, można go wprowadzić



Rysunek 9. Raspbian Wheezy z trybem graficznym

przytrzymując ALT i wpisując 124 na klawiaturze numerycznej.

Po tych operacjach można, niezależnie od wersji systemu, wywołać polecenie:

```
sudo raspi-config
```

i wybrać pierwszą z opcji, czyli „Expand Filesystem”. Następnie należy ponownie zrestartować wirtualną maszynę (rysunek 7). Uwaga – kolejne uruchomienie może potrwać nieco dłużej.

## Dodatkowe ustawienia i aktualizacja

Dla polskich użytkowników korzystne będzie zapewne zmienienie ustawień międzynarodowych, by pozbyć się brytyjskiej klawiatury, a nawet by zmodyfikować język opcji w niektórych programach. Operacja ta jest bardzo prosta. Wystarczy po zalogowaniu do systemu wejść w **raspi-config** z uprawnieniami roota (w przypadku Jessie ponownie trzeba podmontować partycję /dev/mmcblk0p1 do /boot) i wybrać czwartą pozycję z menu (Internationalisation Options). Najpierw najlepiej wybrać opcje lokalizacji (rysunek 8), a kolejnym razem układ klawiatury.

Opcjonalnie można też zaktualizować cały system tak, by pakiety miały najnowsze wersje dla danej dystrybucji. Niestety, może się okazać, że niektóre zmiany spowodują pewne problemy, ale i tak pokazujemy, jak to zrobić. Wystarczy bowiem wpisać jedynie:

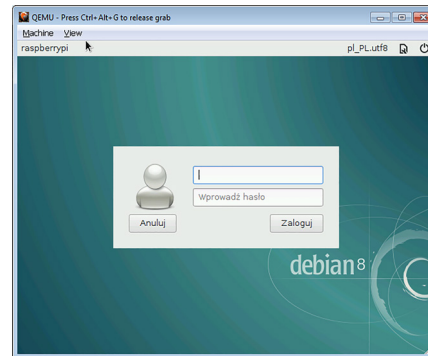
```
sudo apt-get update && sudo apt-get dist-upgrade -y
```

i odpowiadać na ewentualne pytania, zadawane przez instalator.

## Środowisko graficzne

W końcu emulator może też pracować ze środowiskiem graficznym Raspbiana. Można je ustawić jako domyślne poprzez wybór trzeciej opcji w menu **raspi-config**. Ma ona różne nazwy w różnych wersjach systemu, ale dotyczy ustawień bootowania. Wybór „desktop” jako domyślnego ekranu startowego spowoduje, że po starcie systemu będzie uruchamiane środowisko graficzne.

W zależności od wersji systemu operacyjnego, może pojawić się jednak pewien dodatkowy problem. Niekiedy okazuje się,



Rysunek 10. Raspbian Jessie z trybem graficznym

że w Raspbianie Wheezy (rysunek 9) górny pasek menu zaczyna migać. Można wyeliminować to zjawisko, zupełnie go wyłączając, poprzez usunięcie pliku: **/home/pi/.config/lxpanel/LXDE-pi/panels/panel**. Służy temu polecenie:

```
rm /home/pi/.config/lxpanel/  
LXDE-pi/panels/panel
```

Ponieważ spowoduje to, że pulpit będzie pozbawiony jakichkolwiek sensownych kontrolek, warto też już w trybie graficznym kliknąć prawym przyciskiem na ekranie i wejść do ustawień, by móc wskazać, że kliknięcia prawym przyciskiem będą pozwalały na uruchamianie terminala, czy np. przeglądarki internetowej. Przydatna będzie też wskazówka, jak poradzić sobie z systemem z niedziałającym widokiem pulpitu graficznego. Wystarczy przytrzymać prawy ALT i nacisnąć któryś klawisz funkcyjny, by przenieść się do innego ekranu terminala. Ekran graficzny jest domyślnie osiągany w Raspbianie dla kombinacji ALT+F7 (rysunek 10).

W razie dalszych problemów z trybem graficznym, najlepiej będzie zmienić wersję zainstalowanego **lightdm**, który zań odpowiada.

## Podsumowanie

Uruchomienie wirtualnego Raspberry Pi to chyba najbardziej programowy z projektów Soft, jaki może przyjść do głowy. Dla wielu osób cały zachód, związany ze sprawieniem, by Raspbian poprawnie działał, może nie być wart potrzebnego na to czasu, ale należy sobie uświadomić, że w ten sposób można np. na jednym mocniejszym komputerze uruchomić kilka instancji Raspberry Pi, działających z jakimś programem, przystosowanym pod kątem właśnie tej dystrybucji systemu. Co prawda brak fizycznych wyprowadzeń IO jest dużym ograniczeniem, ale z drugiej strony, wirtualne RPI może być wygodnym narzędziem do testowania aplikacji. My na pewno zachowamy sobie jeden przygotowany obraz pliku z Raspbianem dostosowanym do emulatora oraz skrót startujący emulator, by nie trzeba było za każdym razem ręcznie wpisywać wszystkich parametrów wywołania.

**Marcin Karbowniczek, EP**