FOLIA 180

Annales Universitatis Paedagogicae Cracoviensis

Studia Technica VIII (2015)

Piotr Migo, Henryk Noga

Didactic aspects of using Raspberry Pi minicomputer in school science lab

Linux Raspbian Booting

Upon turning on the system, components are loaded, in order [1]:

- the startup initiates the GPU module with the ROM memory. Core ARM CPU and SDRAM are inactive at the beginning. SoC of the ROM contains the level 1 bootloader recorded by the manufacturer, which is loaded into the L2 cache and run by the GPU,
- the level 1 bootloader initiates SD card mounted VFAT partition of the directory /boot with 2 levels bootloader recorded in bootcode.bin. GPU is loaded into memory and L2 cache runs bootcode.bin,
- bootcode.bin activates SDRAM, then GPU loads the bootloader called level 3 start.elf to the /boot directory (this is the GPU firmware). Start.elf RAM divides memory between ARM CPU and the GPU,
- the level 3 bootloader initiates start.elf at the ARM CPU core, as well as subsequently reads configuration files: config, cmdline.txt, and boot kernel from kernel.img, the Linux kernel can be viewed later in the console by the dmesg command, which displays the kernel buffer. 3 files of start/boot directory operate, namely: start.elf (spilt by 2 to ARM and GPU). We can chose the GPU firmware in config.txt by entering these lines:

start_file=start*.elf

fixup_file=fixup*.elf

- config.txt –this file affects the performance of the hardware Pi platform – configuration of input and output devices and the speed of the BCM2835. Display settings are groups by parameters: overscan_ * framebuffer_ * sdtv_ * hdmi_ * config_hdmi_boost. It impacts the running of the system by commands: disable_commandline_tags, cmdline, kernel, ramfsfile, init_uart_baud. For overclocking

[1] Retrieved from http://thekandyancode.wordpress.com/2013/09/21/how-the-raspberry-pi-boots-up/

the platform, the settings might be changed by: arm_freq, gpu_freq, core_freq, h264_freq, isp_freq, sdram_freq, init_uart_clock, init_emmc_clock and a group over_voltage_ *,

– cmdline.txt – contains the Kernel-mode line, which can be passed to options in the kernel at boot Raspbiana.

If you do not feel strong about the System Configuration console commands or by editing some configuration files, you can use the raspi-config tool that automatically appears when you first start Raspberry Pi:

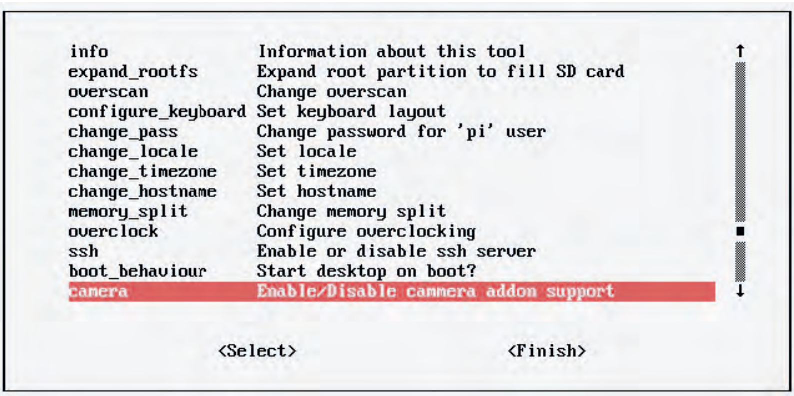

Fig. 1. Main screen of the raspi-config tool

With this tool, you can: extend the size of the root partition, change the local settings: the date, interface language, set the password, the division of RAM, over-clocking, enable the SSH server.

Operations on partitions and files in Linux Raspbian

Let us consider access to the data. Virtual directory allows disk access /dev that contains files that are corresponding to different devices. As for the media, mostly commonly used names like: /dev/hdX (ATA drives and CD-ROM), /dev/sdX (for SATA drives, memory cards and USB stick) or sometimes /dev/mmcblkX (memory cards). The X at the end is the drive number (the hard disk names depends on the order of connection to the disk controller), and will be a, b, c, d etc., e.g.:

/dev/hda1 – first partition (primary) at ATA Disc – primary master,
/dev/sdb4 – fourth partition (primary) at SATA disc – primary slave,
/dev/hdc – entire ATA disc – secondary master,
/dev/sdd6 – sixth partition (logic) at SATA disc – secondary slave.

As we can see there is still marking at the end of the partition number. The digits from 1 to 4 are primary partitions, and if there are more than 4 partitions on the disk, the last of the primary partition is called an extended partition, and the ones from 5 and above are called logical partitions. It is worth recalling that each partition must be formatted before use (and later at the end of the mount ("pinned to the root of the file system – usually /mnt or /media). If we understand this philosophy of installation in Linux, we are not going to have major problems with not only local resources formatted in the native file system (ext, vfat, etc.), but also with network resources (protocols SMB, NFS, etc.).

Below are a few commands showing working with partitions and file systems in Raspbian (most of the commands must be executed with root rights, so we use sudo suffix):

Tab. 1. Example commands for managing partitions and file systems in Raspbian

Example commands	Results
$ sudofdisk–l	Show
$ sudocfdisk /dev/mmcblk0	Manage SD Partitions (e.g. Create partitions)
$ sudo mkfs.ext4 /dev/mmcblk0p3	Format 3 SD partitions as EXT4 3
$ sudomkdir /mnt/disk3 $ sudo mount /dev/mmcblk0p3 /mnt/disk3	Create catalogue /mnt/disk3 and Mounting 3 SD Card partitions in it
$ sudochgrp –R users /mnt/disk3 $ sudochmod –R g+w /mnt/disk3	Share mounted partitions to others users (group of users becomes owner of the catalogue /mnt/disk3 and grant write rights)
$ sudonano /etc/fstab	Edit /etc/fstab to self mounted partitions during Pi start
Write in /etc/fstab file (use [Tab] between columns): /dev/mmcblk0p3[Tab] /mnt/disk3[Tab] ext4[Tab] defaults[Tab] 0[Tab] 2	
$ sudomkdir /mnt/pen $ mount -t vfat /dev/sda1 /mnt/pen	Mount 1 partition from pen drive connected to the Pi platform

$ sudo apt-get install ntfs-3g
$ sudomkdir /mnt/muzyka
$ sudo ntfs-3g /dev/sda2 /mnt/muzyka
	Use storage with the NTFS system in read/write

$ sudomkdir /mnt/video
$ sudosmbclient -L 10.0.0.125
$ sudomount -t cifs↵//10.0.0.125/video /mnt/video
	Use network resource available by SMB (creating the mount directory, viewing shared resources on the machine with the specified IP address)

$ sudomkdir /mnt/pen
$ sudomount -t vfat /dev/sda1 /mnt/pen
$ cd /mnt/pen
$ sudodd of=kopia.img if=/dev/mmcblk0↵bs=2M
	Create SD card copy image to the kopia.img file on a stick drive (i.e. a copy of the bit-by-bit) for example. Transfer the system to a larger SD card

$ sudodd of=/dev/mmcblk0 if=kopia.img↵bs=2M	Same as above: kopia.img recorded on the new SD card

Managing the file, users and permissions in Raspbian

Having some knowledge of the file systems in Linux Raspbian and being able to mount a directory on the root filesystem, we can move on to discuss the file commands (create, read, copy, move, rename, delete, etc.), User management, and important issues, broadcasting rights to individual data file. All console commands are presented as a real-world examples of their use, rather than their general syntax.

Tab. 2. Sample command file operations and users permissions

Example commands	Results
$ ls /mnt/pen	Display folder content /mnt/pen

$ cd /home/pi
$ cd /
$ cd ..
$ cd ~
	Move to the /home/pi directory
	Move to the root directory /
	Move to the directory one level up
	Move to the user's home directory

$ cp /home/pi/led.py /mnt/pen	Copy led.py from Home folder to a pen drive
$ mv /home/pi/logo.png /mnt/pen $ mv logo.png img1.png	Move logo.png to a pen-drive -change name to img1.png
$ rm img1.png $ rm –r /home/pi/test	Delete img1.png file Delete /home/pi/test catalogue along with the content
$ sudomkdir /mnt/disk4	Create /mnt/disk4 catalogue
$ ls –l /home/pi	Display /home/pi content with owner information and file size and modification date
$ sudochmodug=rw ~/.skrypt	Give read-write permissions to the owner group
$ sudochmod go= -R /boot	Receive any permissions to /boot and it content for all users except for the owner
$ sudochmodg-w,o+r .skrypt	Receive the right to make changes by the group and give read access for other users in the file .skrypt
$ sudochgrp users logo.png	Assign rights of the logo.png file to a group of users
$ sudochmod 600 .skrypt	Receive .skrypt file permissions for the group and other users, and give read and write permissions to the owner
$ sudochmod 4711 .skrypt	Give the .skrypt file script permissions, full permissions for the owner and execute permission for the group of other users
$ sudouseradd –m –G adm,audio,users ⏎ robert	Create a robert user, his home directory and assign to selected groups
$ sudopasswdrobert	Set password for Robert user

Basic user in the Raspberry Pi system, has a name of "Pi" with a password "raspberry pi|". We can create new users with the password command useradd and passwd.

Managing the Distribution of the Raspbian software

Raspbian distribution image prepared by the developers contains the most popular and necessary software needed for everyday work and creating projects. However, adapting a system to meet the requirements of the constructor is difficult – you need to enter a lot of improvements into the system, which would take too much SD card space. The advantage of Linux is its scalability, so we have an opportunity to build a different distribution, like small embedded systems, through workstations with software utility up to large clusters of computed servers.

Systems based on Debian have a very good package manager – apt-get, that will meet our requirements for tracking all software installed on our system, installing, updating and deleting specific packages and update the entire Raspbian kernel. We use apt-get quite often taking care of updating the cache, because it just checks for particular software package (buffer works with server software). Another issue related to the software is a fact that many packages require the existence of certain other packages or libraries. This is called depending on their checking and watching apt-get manager. Let's look at a few examples of the apt-get package manager:

Tab. 3. List of commands for packages manager

Example commands	Results
$ sudoapt-getupdate	Update the application package manager
$ sudo apt-get install mc	Install Midnight Commander package or its update
$ sudoapt-getupgrade	Update all packages in the system
$ sudo apt-cache search scratch	Search buffer for the keyword *scratch* in the title or file description
$ sudo apt-get remove mc	Delete Midnight Commander package
$ sudo apt-get purge gimp	Remove GIMP package

Network management In Raspian distribution

The service network through wired and wireless interfaces with a focus stack TCP / IP (the IP addressable network interfaces, numbering service ports TCP and UDP layers and configuring servers at the application layer) is an important element of modern operating systems. We are in a favourable position that our platform, the Raspberry Pi, has Linux as an operating system that works across multiple web servers and routers in the world, it has been tested for several years by developers and hackers. We will focus on network configuration and server settings, and control of the GPIO ports. We will also discuss the commands and operations.

Tab. 4. Commands and configuration files for network operation in Raspbian Linux

Example commands	Results
$ ifconfig	Network interface statistics (like MAC and IP addresses, masks, subnet, broadcast etc.). Normally we see eth0 and lo
$ sudoifdown eth0 $ sudoifup eth0	Disable the Ethernet interface Enable the Ethernet interface
$ ping –c 10 google.pl	Send 10 packages to google.pl

$ sudonano /etc/network/interfaces auto lo iface lo inet loopback iface eth0 inet static address 10.1.1.200 netmask 255.0.0.0 gateway 10.1.1.1 allow-hotplug wlan0 iface wlan0 inetdhcp wpa-ssidSAMPLE_SSID wpa-psk SAMPLE_PASSWD	Edit the configuration file with the parameters of the network interfaces. We want to get a static IP address for the wired interface eth0i DHCP service and WPA encryption for wireless wlan0.
$ sudonano /etc/resolv.conf nameserver10.1.1.20 nameserver10.1.1.21	Edit the configuration file to manually set the servers name
$ sudo /etc/init.d/networking restart	Restart the Raspbian network environment after editing configuration files

$ dmesg | grep ^usb
$ sudo apt-cache search ralink
$ sudo apt-get install firmware-ralink
	Install the USB wireless interface Ralink chipset (finding the messages starting wlan card install the driver)

$ sudoiwlistscan	Scan the environment in order to find wireless hot-spots

More detailed application and other necessary commands, we will use in practical scenarios in which the main elements will be video-tutorials with related descriptions and listings terminal commands and programs in the selected language.

Description of several software libraries

In the last part we will focus on programming languages of the Raspberry Pi platform, and especially on programming libraries that helps use the GPIO ports and communication buses. Although Raspbian system supports many programming languages, ones which are desirable contain a lot of ready-made libraries to help you create projects, have community support, and many examples of active construction. We will mention only three of them and describe their characteristics, and we will deliberately skip the present of their syntax:

– Scratch- it is a tool created for all. Very good at the first contact with programming – a colourful interface encourages the creation of multimedia projects: simple games, presentations and cartoons, as well as interfaces to the external sensors and motors. Scratch interface is divided into several panels: pallet, blocks, a list of objects that make up the program, the stage and the construction of the program window.
– Python – the primary language supported by the Raspberry Pi (the platform name consists of two parts – the “Raspberry” follows the tradition of giving the names of fruits to computer systems, the second part of “Pi” is a nod to the Python language). It is a high-level language. Another feature is the abundance of the Python standard libraries characterized by clarity and conciseness of code. As far

as programming is concerned, paradigms are closer to object-oriented programming. Also, it is characterized by a dynamic system type and automatic memory management. If the language is dynamic, it is often and only performed partially. used as a script language. The Python interpreter analyzes each expression, maps and variables in the memory every time you start the program, Python programs are slower than their counterparts such as low-level compiled one in CC/C ++ – C language is a high level structure language and its introducing object-oriented paradigms. Both versions are compiled languages, so the speed execution of the finished program is high. It has a modern data structures and a rich set of operators, closely associated with Unix, which has thousands of libraries, allowing for low-level operations (indicators). But it also has disadvantages: high flexibility of writing the code, the ability to build a long confusing expressions. Here are some programming libraries prepared for the Pi platform:

RPi.GPIO package for Python

This package adds to Python's GPIO ports controlling programs class (only control ports, support for SPI, I2C, PWM hardware not included). The table below shows the installation process and basic use of RPi.GPIO:

Tab. 5. Instalation and simple use of RPi.GPIO class in Python Programs

Examples commands	Results
$ wgetadres_pobierania	Download package archive
$ tar xvzf RPi.GPIO-0.5.6.tar.gz	Unpack the package
$ cd RPi.GPIO-0.5.6	Enter the main catalogue
$ sudo apt-get install python-dev	Extra install additional libraries
$ sudo python setup.py install	Compile and install the RPi.GPIO
# alternately we change the voltage. On 12 pin connector from P1 to 0V for 3,3V:	

```
import RPi.GPIO as GPIO                # use library To control GPIO
import time                            # use date library
GPIO.setmode(GPIO.BOARD)               # marking porst same like P1 connector
GPIO.setup(12, GPIO.OUT)               # exit pin 12

While True:                            # doing all time (ctr+z finishes)
        GPIO.output(12,False)          # set pin 12 to 0V
        time.sleep(2)
        GPIO.output(12,True)           # set pin 12 to 3,3V
        time.sleep(4)
```

# Generate signal PWM 4s period. On 12 pin of P1 connector:	
import RPi.GPIO as GPIO	# use libl. To control. Gpio
GPIO.setmode(GPIO.BOARD)	# marks ports as P1 on PCB
GPIO.setup(12, GPIO.OUT)	# make pin 12 output
LED01 = GPIO.PWM(12, 0.25)	# set execution. PWM for 4 s periods.
LED01.start(1)	# generate PWM signal
input(‚Zakończ Enterem >>>')	# Enter – finish…
LED01.stop()	# stop PWM signal

Library for Python comes with a RaspiRobotBoard module

This is a library that works with RaspiRobotBoard expansion module. The library itself provides you with: configuration GPIO pins directions cooperating with the module controlling the operation of two DC motors, 2 LEDs, 2 digital inputs and ultrasonic rangefinder. Necessary operations are presented in the table below:

Tab. 6. Instalation and use of RaspiRobotBoard library in version 1

Example commands	Results
$ sudo apt-get install python-rpi.gpio	Install the GPIO package
$ sudo apt-get install python-serial	Install the Serial package
$ wgetadres_pobierania/rrb-x.y.z.zip	Download the compressed library
$ unziprrb-x.y.z.zip	Unzip the archive
$ cd rrb-x.y.z	Go to library catalogue
$ sudo python setup.py install	Compile and install the RaspiRobot library
The illustrative program in Python with comments RaspiRobotBoard control module:	
from raspirobotboard import *	# use Raspirobot lib
rob = RaspiRobot()	# set in memory work object rob
rob.set_led1(1)	# turn on LED1
rob.set_led2(0)	# turn off LED2
rob.set_oc1(1)	# switch on Open Collector
rpb.set_oc1(0)	# switch off Open Collector
rob.forward()	# 2 engines rotate in normal direction
directionrob.left()	# left engine – reverse, right – forward
rob.right()	# left engine – forward, right reverse
rob.stop()	# 2 engines stop
rob.get_range_inch()	# ultrasound range finder.
rob.sw1_closed()	# read the SW1 button position

GPIO control via the Internet using WebIOPi framework

At the end of the theoretical part of the work we will present an interesting framework – WebIOPi, which allows for creating solutions for drivers controlled through a browser or other network applications. It is written in Python, supports UART communication bus, SPI, I2C, and approx. 30 devices that can be connected to the Raspberry Pi (D converters, IO port expanders, sensors), enables you to build Web UI interfaces based on Javascript/HTML protocol, supports COAP to connect electronic devices via the Internet. The table below presents the basic information about WebIOPi framework: installation, commissioning and default Web UI interface to control the GPIO ports (you can connect any device by appropriate buffers):

Tab. 7. Basic information about WebIOPi framework: installation, commissioning and default Web UI interface to control the GPIO ports

Example commands	Results

$ wget adres/WebIOPi-x.y.z.tar.gz
$ tar xvzf WebIOPi-x.y.z.tar.gz
$ cd WebIOPi-x.y.z
$ sudo ./setup.sh
$ sudowebiopi -d -c /etc/webiopi/config
$ sudo /etc/init.d/webiopi start
$ sudo /etc/init.d/webiopi stop
	download WebIOPi package
	unpack WebIOPi package
	go to library location
	run the package
	start the package with default config
	
	Start the service
	Turn of the service

Type http://rpi_adres_ ip:8000 in the web browser and you will see UI with: IN – this tag is port in entry mode OUT – it is a port tag with port in output mode Black tag on P1 connector means low level on the output port Orange tag on p1 connector means high state (3.3V) on the output port We change the state of the tag by clicking the mouse on it	

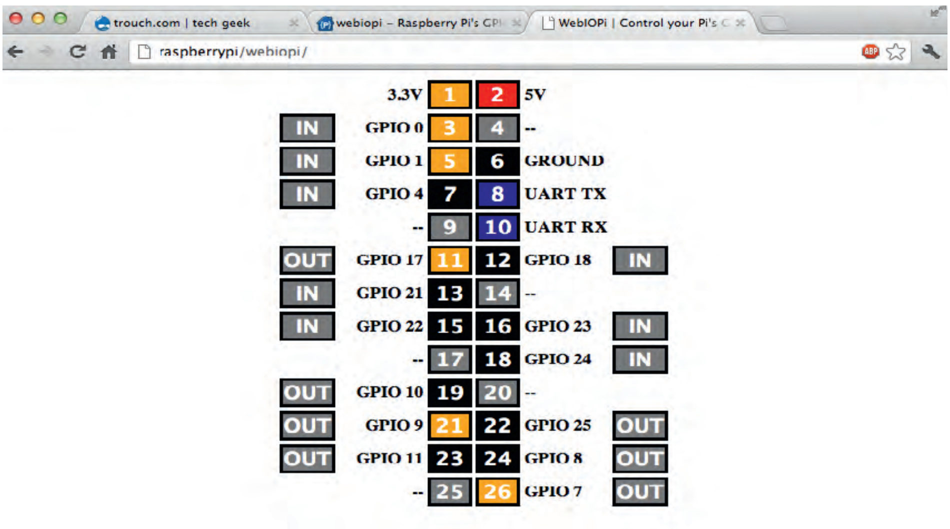

Fig. 2. Interface design control and default GPIO pins after running WebIOPi

Conclusion

It seems that the introduction of the Raspberry Pi minicomputer into the educational process will show students that technical education should not stop at simple computer use for basic programs, popular software with GUI interface and most importantly – it cannot be done in isolation from other fields of technology [7, 8].

References

[1] Dylak S., Ubermanowicz S., *Strategie nauczania-uczenia się infotechniki*, Wyd. Fundacja Wolnego Oprogramowania, Poznań 2014.

[2] Gurbiel E., Hardt-Olejniczak G., Kołczyk E., Krupicka H., Sysło M.M., *Technologia informacyjna*, WSiP, Warszawa 2002.

[3] Juszczyk S., *Technologia Informacyjna w edukacji.* In: (ed.) S. Juszczyk, *Metodyka nauczania informatyki w szkole*, Wyd. Adam Marszałek, Toruń 2001.

[4] Juszczyk S., *Systemy komputerowego wspomagania procesów kształcenia*. In: (ed.) S. Juszczyk, *Metodyka nauczania informatyki w szkole*, Wyd. Adam Marszałek, Toruń 2001.

[5] Kiełbasa M., Depešová J., Noga H., *Szkolny klimat dla twórczości technicznej*. In: *Edukacja--Informatyka-Technika*, Rzeszów 2014.

[6] Piaskowska-Silarska M., Gumuła S., *Social, economic and environmental aspects of municipal waste treatment by thermal methods*. XXIV DIDMATTECH 2011. *Problems in teachers education*. Wydawnictwo Pedagogical University of Cracow – J. Selye Uniwersity In Komárno. Cracow 2011.

[7] Siemieniecki B., *Kognitywistyczne aspekty technologii edukacyjnej – kierunki badań*. In: B. Siemieniecki, *Technologia informacyjna w Polskiej edukacji*, Wyd. Adam Marszałek, Toruń 2012.

[8] Vargová M., *Technology Education in Basic and Upper Secondary Schools*. – Slovak Republic. In: *UNESCO – The Development of new Approaches in Technology and Vocational Education in the Countries in Transition – the Countries of Central Europe and Souht Africa*. An Internation Pilot Project. Participation Programme for Years 2002–2003. No. 183 711 16 ONG. 2003.

[9] http://www.wired.co.uk/news/archive/2013-10/18/eben-upton-rasp-berry-pi

[10] http://en.wikipedia.org/wiki/Raspberry_Pi

[11] http://pclab.pl/art50706.html

[12] http://www.petervis.com/Raspberry_PI/BCM2835_Block_Diagram

[13] http://elinux.org/RPi_VerifiedPeripherals

[14] http://www.wvshare.com/product/RPi-B-CN.htm

[15] http://hallard.me/arduipi-getting-started/

[16] http://raspberryalphaomega.org.uk/2013/02/06/automatic-raspberry-pi-board-revision-detection-model

[17] http://fritzing.org

[18] http://www.raspberrypi.org/downloads/

[19] https://launchpad.net/win32-image-writer

[20] http://thekandyancode.wordpress.com/2013/09/21/how-the-rasp-berry-pi-boots--up/

[21] https://code.google.com/p/webiopi/wiki/DEVICES

[22] http://www.raspberrypi.org/model-b-revision-2-0-schematics/
[23] http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Robotics/RaspiRobot_ Board.pdf
[24] https://github.com/Elektordi/pi-accesscontrol/blob/master/doc/rp BreakOutV0_4_sch.pdf
[25] http://botland.com.pl/moduly-rozszerzajace-raspberry-pi-b/2066-pi-clock-zegar-rtc--do-raspberry-pi.html

Abstract

Raspberry Pi minicomputer can be used in vocational training. It is hard to get quick benefits of the computer based on Linux System if one has no experience and proper knowledge with it. However, the time spent on learning may result in some profits [2, 3]. If students have enough patience, the Raspberry Pi ARM Linux OS, programming support, network support, TCP / IP, GPIO ports will become interesting options [1, 5].
I think that the Raspberry Pi platform is an interesting place of work connected with other divisions of technology [4, 6].

Key words: minicomputer, *Raspberry Pi*, technical education

Piotr Migo, Henryk Noga
Pedagogical University of Cracow
Institute of Technology
ul. Podchorążych 2
30-084 Kraków, Poland