On teaching Raspberry Pi for Undergraduate University Programmes

Septimiu Mischie
Faculty of Electronics and Telecommunications
Politehnica University of Timisoara
Timisoara, Romania
septimiu.mischie@upt.ro

***Abstract*—This paper presents some aspects regarding the teaching Raspberry Pi in an undergraduate university programme. Laboratory lessons and diploma projects are considered. Basic knowledge about this system, such as using the GPIO (General Purpose Input Output) in different ways, remote accessing by a PC or recording by two types of camera are presented first. Then, more advanced applications as web server or stereo vision are presented. By these applications, especially those based on video cameras, we bring some news compared to what currently exists in the literature.**

Keywords—Raspberry Pi; linux; Phython; GPIO; video camera; stereo vision; internet; web server

I. INTRODUCTION

The first release of the Raspberry Pi system was launched in 2012. From then until today, other versions were launched, especially recently. However, the first release, Raspberry Pi 1 model B, was for a long time the most powerful one. The resources of this model are the following: a 700 MHz single core ARM1176 processor, 512 MB of SDRAM, a video Core IV (actually these three are inside the chip BCM2835), two USB 2.0 ports, a 100 Mb/s Ethernet port, a HDMI video output, a special connector (CSI, camera serial interface) for video camera and an extension connector having 26 pins for GPIO and serial interfaces [1, 2]. The operating system is a Linux distribution called Raspbian and it is stored on the SD card. The first contact of my students with Raspberry Pi was during the traditional *Hard and Soft* contest from Suceava, Romania in May 2013. Due to my students' interest for Raspberry Pi, we decided to introduce this system in our curricula, both in laboratory lessons and diploma projects. However, things were not so easy, at least in laboratory lessons because we had to replace other lessons. Now, after three years following that event, we introduced five double lectures in the second semester of our programme of study and also we achieved more diploma projects. In the meantime, the use of Raspberry Pi systems has taken great amplitude because there is a large community of users that make their project public. Hence, our idea to popularize Raspberry Pi among students has proved to be a good initiative.

Raspberry Pi has all the necessary characteristics to be used like a PC, but having smaller resources. It also allows interfacing with a lot of devices, such as switches, LEDs, sensors or actuators, like a microcontroller based system. Finally, by means of some suitable adapters it has wireless connectivity (Bluetooth, WiFi) and it can also take pictures. These facilities, as well as its small size and its processor type, make Raspberry Pi be more similar to a smartphone. In a few words, Raspberry Pi has features from a PC, microcontroller and smartphone. Therefore, in order to be used to its entire capacity, it would be best to make use of most of the characteristics of each of these three devices.

There are other systems similar to Raspberry Pi, in terms of size and which also use Linux based operating systems. Sitara AM 335x Starter Kit has in addition a touch screen display. We popularized it among students for diploma projects and research papers [3]. Unfortunately, it does not become so used as Raspberry Pi. Beagle Bone Black (BBB) is comparable from most points of view with Raspberry, but having fewer facilities in video processing [4]. BBB also has a powerful community of users but it is not as popular as Raspberry Pi.

This paper is organized as follows. Basic knowledge about Raspberry Pi and the content of laboratory lessons are presented in section II. Some of the most important results obtained during diploma projects are presented in the section III, while the conclusions are drawn in the section IV.

II. USING RASPBERRY PI IN LABORATORY LESSONS

Laboratory lessons about Raspberry Pi were introduced in the second semester of the Electronics and Telecommunications undergraduate programme of study. To have a better understanding of these lessons, the students need prerequisite knowledge about programming in C language and Electric and Electronic Circuits. The laboratory class is performed with 16-18 students, who are subsequently divided into pairs. Each pair uses an experimental setup containing one PC with internet access and one Raspberry Pi system with a screen, a keyboard and a mouse, and also a Wi-Fi adapter, two types of video cameras and a USB hub.

A. Lesson 1: Using Raspberry Pi as a standalone PC. Basics about Linux

In this lesson, students are introduced to Raspberry Pi. The system needs a DC voltage of 5V from a suitable adapter and

978-1-5090-3748-3/16/$31.00 ©2016 IEEE

it is connected to a screen by a HDMI to VGA adapter. A mouse and a keyboard are also connected by means of the two USB ports.

Raspberry Pi can be used in both graphical user interface and through Linux commands. Among the most usual Linux commands, such as *pwd*, *cd* or *ls* we also presented the command *apt-get install* to install some new commands as in the following example:

```
$ sudo apt-get install scrot
```

Execution of such command needs internet access. In this case the new command *scrot* (screen shot) allows capture the screen. The prefix *sudo* comes from *s*uper *u*ser *do* and sometimes needs a password.

A more challenging task is to turn ON or OFF or make to blink the only LED of Raspberry Pi that can be controlled by the user. This task requires among the usual command *cd* other commands, such as *echo* or *cat* as well as the command *chmod* to establish the permissions for a file. Generally, the LEDs files are inside the directory */sys/class/leds*. The only available file is *led0*. By using the command

```
$ cat /sys/class/leds/led0/trigger
```

the following content is displayed

```
none [mmc0] timer heartbeat
```

which means that the source that triggers the *led0* is *mmc0* (multimedia card 0). This source can be modified to be *timer* by the command

```
$ echo timer > trigger
```

As a consequence, the *led0* starts to switch between ON and OFF, with each of the two states having the default value of 500 ms. The previous command can be executed only if the next command is previously executed

```
$ sudo chmod a+w trigger
```

that gives write (*w*) permissions for all (*a*) users for the file trigger.

To modify the default values for the ON and OFF times, the files *delay_on* and *delay_off* which are included in the directory */sys/class/leds/led0* after the content of the file *trigger* become *timer*, need to be modified.

B. Lesson 2: Using the GPIO by Linux commands

The simplest possibility of Raspberry Pi to interface with the outside is the GPIO pins. Thus, it contains a 26-pin GPIO connector. There are more possibilities to access GPIO, one of them being by Linux commands. To implement the *output* function, a few LEDs can be connected to the appropriate pins. To implement the *input* function a push button which applies 1 or 0 logic to the appropriate pin is used.

In order to access the GPIO pins there is another directory, */sys/class/gpio* which contains by default the files *export, gpiochip0, unexport*. By writing the desired pin number, e.g. 24, into the file *export*, a new directory, *gpio24* is created inside the directory */sys/class/gpio*. This directory will contain the following files:

```
active_low device direction edge subsytem uevent value
```

Two of these files, *direction* and *value*, will be used in the following examples.

Thus, the pin can be configured as an *output* by

```
$ echo out > direction
```

Then, the desired value, 1 or 0 can be sent to that pin by

```
$ echo 1 > value
```

If the value of *direction* was established to *in*, the pin is an *input* and its state can be read by (the answer can be 1 or 0)

```
$ cat value
```

If a pin is no longer used it can be disabled by

```
$ echo 24 > /sys/class/gpio/unexport
```

C. Lesson 3 and 4 Using GPIO in Phyton and C

Although we can create executable files which contain Linux commands, the best solution for implementing more complicated tasks is to use programming languages as Python or C. In principle, both of these languages can implement the same tasks. However, there are some differences.

The Raspbian operating system contains support for Python by default. The only thing that must be done is to import the GPIO library by the instruction `import RPi.GPIO as GPIO`. There are two ways to select the pin numbers. The first corresponds to the BCM2835 numbering, and is selected by the instruction `GPIO.setmode(GPIO.BCM)`. The second corresponds to the natural numbers (1-26) of the GPIO connector, the instruction being `GPIO.setmode(GPIO.BOARD)`.

One possibility to access GPIO by C language is by using the WiringPi package. To install it, the following Linux commands must be executed:

1. `$ sudo apt-get install git-core`
2. `$git clone git://git.drogon.net/wiringPi`

The last command has as effect making the directorry `/Home/Pi/wiringPi`

3. `$ cd /Home/Pi/wiringPi`
4. `$ ./build`

To initialize the WiringPi package, the function `wiringPiSetup()` must be executed.

Table I presents the most important actions that can be executed in an application that requires accessing the GPIO in both Python and WiringPi.

Python is a scripting language, which means that it does not need compilation. In contrast, running an application in WiringPi needs the compilation step first. Thus, the execution time of a WiringPi application is shorter than a Python application. This can be proven by implementing a program that commands by 1 and 0 an output pin in an infinite loop in both languages. In each case a square signal having a 0.5 duty cycle is obtained, but the frequency is higher when the program is executed in C language.

TABLE I. ACCESING GPIO IN PYHTON AND WIRINGPI

Action/Language	**Pyhton**	**WiringPi**
Pin as output Pin as input	GPIO.setup(24,GPIO.OUT) GPIO.setup(23,GPIO.IN)	pinMode(5,OUTPUT) pinMode(4,INPUT)
Establish the logic level for an output pin	GPIO.output(24,1)	digitalWrite(5,1)
Read the logic level of an input pin	GPIO.input(23)	digitalRead(4)

D. Lesson 5: Using video cameras with Raspberry Pi. Remote access of Raspberry Pi

Raspberry contains the CSI connector where the dedicated Raspberry Camera can be connected. The camera must be enabled by the command `$ sudo raspi-config` and then the option 5, `Enable camera`. After this, the system is rebooted. There are two Linux commands for this camera, one for taking photos and another one for recording, e.g.:

`$ raspistill -o file_name.jpg`, saves a photo in file_name.jpg.

`$ raspistill -t 4000`, turn on the camera for 4 sec. and sends the video signal to HDMI output.

`$ raspivid -t 10000 -o clip_name.mp4`, records 10 sec. and saves it in the file clip_name.mp4.

To play this file, a player such OMXplayer must be first installed. Then, the next command allows playing

```
$ omxplayer -o hdmi clip_name.mp4
```

Having two USB ports (a USB hub can be used to extend the number of ports), also other webcams can be connected to the Raspberry Pi. One of them is Logitech C270. To use this camera the following command must be executed:

```
$ sudo apt-get install fswebcam
```

Then, taking photos can be done by the command

```
$ fswebcam -r 640x480 -- save file_name.jpg
```

To record video by Logitech C270, the application `ffmpeg` must be installed. More commands are necessary for this, only one is presented for space reasons:

```
$ git clone git:// source.ffmpeg.org/ ffmpeg.git ffmpeg
```

Then, the following command allows recording video:

```
$ ffmpeg -f v4l2 -r 25 -s 640x480 -i /dev/video0 video.avi
```

It is not necessary to use a computer screen, a keyboard and a mouse to develop applications on Raspberry Pi. Thus, it can be connected in the same local network by a PC and then it can be remotely accessed. For this purpose, the *SSH (SecureShell) server* application must be enabled on Raspberry Pi, while a *SSH client* application must be installed and then executed on the PC. The simplest solution for a SSH client on Windows is to install the *putty* application (it can be free downloaded from http://www.chiark.greenend.org.uk /~sgtatham/putty/ download.html).

To connect Raspberry Pi to a local network there are two possibilities: connect it to a router by (i) a suitable cable that use the Ethernet port or (ii) by a WiFi adapter (WiPi) inserted in a USB port. To see the IP address of Raspberry Pi, the `$ ifconfig` command can be executed. Another possibility is to use the configuration web page of the router (192.168.1.0).

If the PC is connected in the same local network with Raspberry Pi, the putty.exe can be executed, and the IP of the Raspberry Pi must be introduced in the appropriate field while the port number is 22. Following this, a window is opened where the login and the password of Raspberry must be introduced. This way, we have access to the terminal to send Linux commands to Raspberry Pi.

To exchange data with Raspberry PI, the *FileZilla* free client application can be installed on the PC. This application requires the IP of Raspberry Pi, the login, the password and the port number (22). In this way, the pictures, movies or other files saved on Raspberry Pi can be moved on the PC.

III. USING RASPBERRY PI IN DIPLOMA PROJECTS

A. Raspberry Pi as a web server

The first diploma project presents implementing a web server on Raspberry Pi. The following resources are connected to the Raspberry Pi 2 model B in this case: a Logitech C270 webcam, an analog temperature sensor together with an analog to digital converter with SPI interface and four leds. Thus, the facilities of this server which can be started by introducing the address http://rpiprojects.ddns.net in a web browser, fig. 1 are the following:

Turn ON or OFF each led by a left click of the mouse on the appropriate symbol.

Displays and update at each second the temperature.

Displays live video from the place where Raspberry Pi is located.

Thus, the LED Control Section of the web page shows us that two LEDs are ON and the other two are OFF and the live video confirms this.

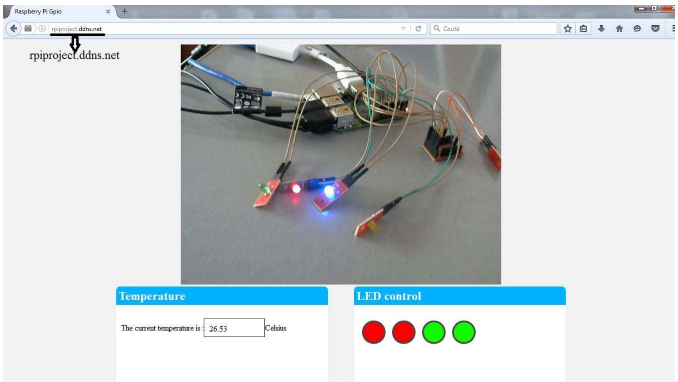

Fig. 1. Raspberry Pi as web server

One of the most challenging parts of this web server has been the integration of the video streaming in the web browser. Both Raspberry Pi and the PC we want to access the web server need to have VLC Player installed. Raspberry Pi sends the video stream to the port 8160. The file index.php of the web server must contain among others the following program line:

```
<embed id="vlcEmb" type="application/x-google-vlc-plugin" version="VideoLAN.VLCPlugin.2" auto play="yes" loop="no" width="480" height="270" target=http://192.168.1.121:8160/ ></embed
```

B. Using Raspberry Pi to compute the distance to any point from a scene by stereo vision

The second diploma project presents an application where two Logitech C270 webcams are connected to the USB ports of Raspberry Pi 2 model B. This assembly can measure and display the distance to any point from the scene that is captured by one of the webcams. This idea is based on the fact that the images captured by the two webcams are not the same. Each point of an image has its corresponded point in the other image, horizontally shifted by a number of pixels. This shift is called disparity. It can be proven that the distance from that point to the camera plane is inversely proportional with that shift [5]. The only restriction is that this point should have a corresponding one in the other image. This is true for most of the points except for those on the left border of the left image which have no correspondent on the right image. The two webcams are located in the same plane having a distance of 20 cm between them.

To implement this idea, a C++ program which uses Open CV library is run on Raspberry Pi. Also the Qt creator is used for displaying the results in a graphical form. Mainly, the following steps are executed:

1. Acquiring several sets of images by the two webcams. They must contain a checkerboard pattern having an odd number of squares on one side and an even number of squares on the other side [6].

2. Implementing the calibration of the webcams using the images acquired on the step 2. In this way, the intrinsic and extrinsic parameters of the webcams are determined [6].

3. Choosing one set of the images previously captured or acquiring other two images and compute the disparity map of the left image. For this purpose, the Semiglobal Matching algorithm is used [7]. The disparity map is displayed in a graphical interface where the user can adjust some parameters to get a better disparity map.

4. Using the disparity map and intrinsic and extrinsic parameters to construct the tridimensional scene. Thus, each point of one of the left image will get the third coordinate, z, among the first two x and y.

5. Using a graphical interface to display the image and selecting a point by means of a cursor; computing and displaying the distance from the plane of the two cameras to that point. Fig. 2 and 3 presents two different examples. The red cross represents the cursor.

IV. Conclusion

This paper has presented some basics applications of Raspberry Pi as well as more advanced two. The basics consist of the usage of the GPIO, the cameras or the remote access by SSH. The advanced one uses the facilities such as web server, live streaming or stereo vision. As a future work, we have more ideas. Regarding laboratory lessons we want to interface Raspberry Pi with a smartphone, because all the students have such devices, and also with some movement sensors as gyroscope and accelerometer. Regarding the diploma projects we want to improve the algorithm that compute the distance by optimizing the method which compute the disparity map and thus it is allowed us to be near to a real time algorithm.

Fig. 2. The diparity of the point is 90 pixels and the distance is 119 cm

Fig. 3. The disparity of the point is 62 pixels and the distance is 170 cm

References

[1] T. L. Warner, "Hacking Raspberry Pi", Que Publishing, 2014.

[2] G. Girling, "Raspberry Pi Manual: A practical Guide the Revolutionary small Computer", Haynes, 2014.

[3] C. Moşneang, S. Mischie, and R. Pazsitka, "Integrating the accelerometer of the AM335X Sitara Starter Kit in a QT application," Proceedings of the 6th EDERC, 2014, pp. 225-229.

[4] D. Molloy, "Exploring BeagleBone",John Wiley & Sons Inc. 2014.

[5] Guanyu Yi, *High-Quality, Real-Time HD Video Stereo Matching on FP*GA, Master thesis, Delft University of Technology, 2011.

[6] MathWorks - http://www.mathworks.com, Computer Vision System Toolbox Users Guide

[7] H. Hirschmuller, Stereo Processing by Semiglobal Matching and Mutual Information, IEEE Trans. on Pattern Analysis and Machine Inteligence, vol. 30, February 2008, pp. 328-340.