ACADEMIC JOURNAL ON COMPUTING, ENGINEERING AND APPLIED MATHEMATICS, VOL. 01, NO. 01, March 2020

Technical report of Raspberry Pi prototype for lectures in public universities

Tiago da Silva Almeida[1]

[1] *Universidade Federal do Tocantins, Computer Science, Tocantins, Brazil*

Reception date of the manuscript: 11/02/2020
Acceptance date of the manuscript: 27/02/2020
Publication date: 09/03/2020

Abstract— This paper presents a low-cost prototype for lectures in public universities based on Raspberry Pi. It is described how to connect and configure the prototype using an infrared remote control. Technologies applied in education are widely explored in recent researches and can be held to lower the cost. The proposed prototype is 86% cheaper on average (compared to ordinary computer) and can be used for automation of the classrooms besides lectures. For example, the prototype can be additionally used to access control, environment monitoring, and management of the environment utilization by the users.

Keywords—Raspberry Pi, Prototype, Lectures

I. Introduction

With the advent of the Internet of Things (IoT), System-on-Chip (SoC) devices are widely used in many applications for different areas. SoCs can be used from military, medical, commerce, logistics, entertainment and educational applications.

In many educational institutions in Brazil, the allocation of resources for the development of activities is a bottleneck. This directly impacts the quality of the applied activities. Thus, low-cost resources are important to the maintenance of the activities.

For the simple activity of lectures, the most used equipment is a multimedia projector and a computer. At public universities, costs on average R$ (Reais) 5,518.12[1] and computer costs on average R$ 4,369.00[2]. This is a high cost for public universities in periods of limited spending.

Many research papers have been done using Raspberry Pi due to its facility to work and implement solutions. For example, [1] describes a data processing system based on Raspberry Pi to inertial sensors. The paper is just an introduction and does not bring any results. [2] developed a touchscreen platform for rodent testing to allow cognitive testing based on Raspberry Pi and Arduino. This has advantages compared to the standard maze apparatuses typically employed in rodent behavioral testing, according to the authors.

[3] proposed the extension of the AP (Access Point) configuration algorithm to deal with this dynamic nature (users often repeat joining and leaving the network) using Raspberry Pi for the AP. [4] proposed a renewable energy monitoring system for data acquisition and transmission applied to real-time cloud monitoring of a decentralized photovoltaic plant also based on Raspberry Pi embedded in each plant.

In the same field, [5] used a Raspberry Pi as a web server for home automation, with all transducers (sensors and actuators) connect by the internet to the Raspberry Pi.

On the other side, a lot of papers focus on “how to use new technologies or devices to improve education and solve its issues”. For example: [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] and [16].

In this context, this paper proposes a prototype for uses in lectures based on Raspberry Pi. The prototype was built based on the work of [17]. Section II is explained how the prototype was connected and configurated. Section III concludes the paper and makes some discussion about the prototype.

II. Methods and materials

The materials and devices used for the prototype were:

- 1 -A Raspberry Pi 3 B with an 32 GB SD card (R$ 360,00);
- 2 - Li-ion Battery 2.200 mAh (R$ 50,00 each);
- 1 - TP4056 Battery Charger Module (R$ 6,00);

Contact data: Tiago da Silva Almeida, Quadra 109 Norte, Avenida NS-15, ALCNO-14, Plano Diretor Norte, Palmas/TO, Brazil, tiagoalmeida@uft.edu.br

[1]According to data from the Brazilian federal government portal, available for consultation at: `http://paineldeprecos.planejamento.gov.br/analise-materiais`.

[2]Also according to data from the Brazilian federal government portal, available for consultation at: `http://paineldeprecos.planejamento.gov.br/analise-materiais`.

- 1 - 1838B IR receiver (R$ 7,00);
- 1 - Universal Remote for datashow (R$ 130,00).

The IR (InfraRed) sensor has just three pins, which was connect with three pins on the GPIO (General-Propose Input Output) connector in Raspberry Pi (VCC, GND, and SIGNAL). The connection is: a) VCC pin of IR receiver to 3.3V pin of Raspberry Pi; b) GND pin of IR receiver to any Ground pin of Raspberry Pi; c) SIGNAL pin to GPIO 18 of Raspberry Pi.

Each button of an IR remote control has a string of specific encoding. When a button is pressed, the IR transmitter in the remote control was send out the corresponding IR encoding signals. On the other side, when the IR receiver receives certain encoding signals, it will decode them to identify which button is pressed.

To this prototype, it was used the `LIRC`[3] library to read infrared signals returned by buttons of the remote control and translate them to button values. Then, it was used `pylirc` (Python) to simplify the process for reading values from the remote control.

It's important to highlight that `LIRC` library runs without problems with Raspbian Jessie operational system until 2017. For newer versions of Raspbian is required other configuration of the libraries.

To configure the Raspberry Pi for remote control is required these steps:

1. Download the `LIRC` library using the following command from terminal:

```
sudo apt-get install lirc
```

2. Add the two lines below to `/etc/modules`. This will start the modules on boot of Raspbian. Pin 18 will be used to take the output from the IR sensor. Use the following command:

```
sudo nano /etc/modules
```

And add this two lines in the file opened:

```
lirc\_dev
lirc\_rpi gpio\_in\_pin=18
```

Also, edit the `/boot/config.txt` file. Open the file using the command:

```
sudo nano /boot/config.txt
```

And add the following line to the file:

```
dtoverlay=lirc-rpi,gpio\_in\_pin=18
```

3. Edit `/etc/lirc/hardware.conf` using the command:

```
sudo nano /etc/lirc/hardware.conf
```

And type the following code exactly as shown below.

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching
    lircd
LIRCD_ARGS="--uinput"
# Don't start lircmd even if there seems to be
     a good config file
# START_LIRCMD=false
# Don't start irexec, even if a good config
    file seems to exist.
# START_IREXEC=false
# Try to load appropriate kernel modules
LOAD_MODULES=true
# Run "lircd --driver=help" for a list of
    supported drivers.
DRIVER="default"
# usually /dev/lirc0 is the correct setting
    for systems using udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"
# Default configuration files for your
    hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""
```

Type the command to reboot.

4. To perform a quick test to check if `LIRC` is working, it is required to stop the `LIRC` daemon and start `mode2`. `mode2` shows the pulse/space length of infrared signals.

To start the `LIRC` in `mode2` type the following command:

```
sudo /etc/init.d/lirc stop
mode2 -d /dev/lirc0
```

5. The `irrecord` command helps to discover the IR codes used by the remote control and assist with creating a `lircd.conf` file which will be used by `LIRC`.

To record the IR code type the following command from the terminal:

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

Once started, `irrecord` is going to show detailed instructions on how to configure the remote control. Each button should be a pre-defined name. Running `irrecord-list-namespace` is going to display a list of available names. It can be set any name for any button from the `irrecord-list-namespace` list. After successfully recording the configuration file, it looks like the following listing:

[3] `LIRC` is a package that allows you to decode infrared signals of many (but not all) commonly used remote controls. LIRC runs as a daemon that will decode IR signals received by the device drivers and provide the information on a socket. We will then write a program in the user space to monitor this socket for input events using the LIRC client library.

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
#
# this config file was automatically generated
# using lirc-0.9.0-pre1(default) on Mon Dec 16
    20:26:03 2019
#
# contributed by
#
# brand: /home/pi/lircd.conf
# model no. of remote control:
# devices being controlled by this remote:
#

begin remote

name /home/pi/lircd.conf
bits 16
flags SPACE_ENC|CONST_LENGTH
eps 30
aeps 100

header 9093 4490
one 645 1625
zero 645 494
ptrail 647
pre_data_bits 16
pre_data 0x8C73
gap 108500
toggle_bit_mask 0x0

begin codes
KEY_LEFT 0x13EC
KEY_UP 0x639C
KEY_RIGHT 0x936C
KEY_DOWN 0xE31C
KEY_ENTER 0x53AC
KEY_TAB 0xE11E
KEY_ALTRELEASE 0x31CE
KEY_ALTPRESS 0xF10E
KEY_CLOSE 0x916E
KEY_PRESENTATION 0xD32C
KEY_ESC 0x33CC
KEY_POWER 0x817E
end codes

end remote
```

Now replace the existing configuration file (which is most likely empty) with the created new one, by the following command:

```
sudo cp lircd.conf /etc/lirc/lircd.conf
```

6. Restart LIRC by the following command:

```
sudo /etc/init.d/lirc restart
```

irw can be used to test the new configuration file. irw sends data from Unix domain socket to stdout.

7. At this stage was created a virtual keystroke from the remote control. It was used the Python (pylirc) code to do this. pylirc is LIRC Python wrapper and it is required to access LIRC from Python programs. To install pylirc should be typed following command in the terminal:

```
sudo apt-get install python-pylirc
```

Create a lircrc.conf file. The lircrc file is used to map the key symbols defined in lircd.conf to application-specific strings. Thus, this file cannot be configured until lircd has been configured to provide proper key symbols as displayed by irw. A lircd was already configureted in the previous step.

The lircrc file should be placed in the home directory as /.config/lircrc. Optionally you can create a system-wide configuration file located in /etc/lirc/lircrc which will be used when no lircrc file can be found in the /home directory.

The syntax of the lircrc file consists of one or more of the following constructions:

```
begin
prog = ...
remote = ...
button = ...
[button = ...] (optional, for key sequences)
repeat = ...
delay = ...
ignore_first_events = ...
config = ...
[config = ...] (optional, for toggle button
    behaviour)
mode = ...
flags = ...
end
```

The complete lircrc file that was prepared for the prototype is as follows:

```
begin
button = KEY_ESC
prog = irexec
repeat = 0
config = esc
end
begin
button = KEY_TAB
prog = irexec
repeat = 0
config = tab
end
begin
button = KEY_LEFT
prog = irexec
repeat = 0
config = left
end
begin
button = KEY_RIGHT
prog = irexec
repeat = 0
config = right
end
begin
button = KEY_DOWN
prog = irexec
repeat = 0
config = down
end

begin
button = KEY_UP
```

```
prog = irexec
repeat = 0
config = up
end
begin
button = KEY_ALTRELEASE
prog = irexec
repeat = 0
config = altrelease
end
begin
button = KEY_ALTPRESS
prog = irexec
repeat = 0
config = altpress
end
begin
button = KEY_CLOSE
prog = irexec
repeat = 0
config = close
end
begin
button = KEY_PRESENTATION
prog = irexec
repeat = 0
config = presentation
end
begin
button = KEY_POWER
prog = irexec
repeat = 0
config = power
end
```

8. PyUserInput is a cross-platform Python module to take control of the mouse and keyboard in Python. PyUserInput is registered on PyPI (Python Package Index) and updated periodically, so tools such as pip can be used to install. Type the following command from terminal:

```
pip install PyUserInput
```

Here is the complete python program to emulate keyboard from remote key-press:

```
#!/usr/bin/env python

import pylirc , time
from pykeyboard import PyKeyboard
from pymouse import PyMouse
import os

k = PyKeyboard()
m = PyMouse()

blocking = 0;

x = 0
y = 0

def setup():
    pylirc.init("irexec")

def key_press(config):
    if config == 'esc':
        k.tap_key(k.escape_key)
        print 'ESC key'
    if config == 'tab':
        k.tap_key(k.tab_key)
        print 'TAB key'
    if config == 'presentation':
        k.press_key(k.control_key)
        k.press_key(k.shift_key)
        k.tap_key('p')
        k.release_key(k.shift_key)
        k.release_key(k.control_key)
        print 'Presentation mode in Okular'
    if config == 'power':
        os.system('sudo shutdown now')
        print 'Power Down'
    if config == 'close':
        k.press_key(k.control_key)
        k.tap_key('q')
        k.release_key(k.control_key)
        print 'Closing Okular'
    if config == 'altrelease':
        k.release_key(k.alt_key)
        print 'Releasing ALT'
    if config == 'altpress':
        k.press_key(k.alt_key)
        print 'Pressing ALT'

def loop():
    while True:
        s = pylirc.nextcode(1)
        while(s):
            for (code) in s:
                print 'Command: ', code["config"]
                key_press(code["config"])
            if(not blocking):
                s = pylirc.nextcode(1)
            else:
                s = []

def destroy():
    pylirc.exit()

if __name__ == '__main__':
    try:
        setup()
        loop()
    except KeyboardInterrupt:
        destroy()
```

9. Save the file as remote.py to the directory /home/pi in the Raspberry Pi. Make the file executable by the following command:

```
sudo chmod +x remote.py
```

10. Open the autostart file by the command:

```
sudo nano /.config/lxsession/LXDE-pi/autostart
```

Add the following two lines at the end of the file:

```
@pcmanfm /home/pi/media --wallpaper-mode=tile
@python /home/pi/remote.py
```

So far, the flash drive with the PDF presentation is going to load when the Raspbian startup. To do so, it is automatic open up the /home/pi/media directory to choice the flash drive. The presentation file is open with Okular PDF reader and it is pre-installed in Raspberry Pi. At the same time, it is also started remote.py file.

TABLE 1: POWER SPECIFICATION OF ALL RASPBERRY PI MODELS.

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	400mA
Raspberry Pi 3 Model A+	2.5A	Limited by PSU, board, and connector ratings only.	350mA
Raspberry Pi 3 Model B+	2.5A	1.2A	500mA
Raspberry Pi 4 Model B	3.0A 1.2A	600mA	
Raspberry Pi Zero W/WH	1.2A	Limited by PSU, board, and connector ratings only.	150mA
Raspberry Pi Zero	1.2A	Limited by PSU, board, and connector ratings only	100mA

To add the battery inside of a case of the Raspberry Pi, the default micro USB (Universal Serial Bus) port is not a good option. A good option is providing power using GPIO. For this, it was used the GPIO of Raspberry Pi of VCC (pin 4) and GND (pin 6). The power specification of all Raspberry Pi models are shown in Table 1[4].

For tests were connected to two batteries of 3.7 V in series, adding voltage to 7.4 V and keep the current. How the batteries have 2,200 mAh, the prototype was able to supply the system for 2.5 hours. It is a low time supply for four hour-lecture (50 minutes). But, adding two more batteries of the same features is enough for almost 5 hours of running.

III. CONCLUSION

This paper is a technical report about the construction of a low-cost prototype to use in lectures by universities based on Raspberry Pi. Due to the simplicity, the prototype runs well and it is easy to implement.

The Raspberry Pi was chosen because of its capability of processing, besides the low cost. The prototype can be installed in classrooms and uses to control the environment, like temperature monitoring and access control of the classrooms. It can be used as a web server, or such as a node in a local network.

At the end, the total costs were R$ 603.00 on average, against R$ 4,369.00 of an ordinary computer in public universities, also on average. This represents 86.19% of cost reduction just for regular lectures, besides other possibilities of implementation using the same equipment.

[4]https://www.raspberrypi.org/documentation/faqs/.

REFERENCES

[1] A. Alvarellos, A. Vázquez, and J. Rabuñal, "Raspberry pimu: Raspberry pi based inertial sensor data processing system," *Proceedings*, vol. 2, no. 18, 2018. [Online]. Available: https://www.mdpi.com/2504-3900/2/18/1159

[2] J. D. O'Leary, O. F. O'Leary, J. F. Cryan, and Y. M. Nolan, "A low-cost touchscreen operant chamber using a raspberry pi$^{\mathrm{TM}}$," *Behavior Research Methods*, vol. 50, no. 6, pp. 2523–2530, Dec 2018. [Online]. Available: https://doi.org/10.3758/s13428-018-1030-y

[3] M. M. Islam, N. Funabiki, M. Kuribayashi, S. K. Debnath, K. I. Munene, K. S. Lwin, R. W. Sudibyo, and M. S. A. Mamun, "Dynamic access-point configuration approach for elastic wireless local-area network system and its implementation using raspberry pi," *International Journal of Networking and Computing*, vol. 8, no. 2, pp. 254–281, 2018.

[4] R. I. Pereira, I. M. Dupont, P. C. Carvalho, and S. C. Jucá, "Iot embedded linux system based on raspberry pi applied to real-time cloud monitoring of a decentralized photovoltaic plant," *Measurement*, vol. 114, pp. 286 – 297, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026322411730605X

[5] V. Vujović and M. Maksimović, "Raspberry pi as a sensor web node for home automation," *Computers & Electrical Engineering*, vol. 44, pp. 153 – 171, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790615000257

[6] N. Miglani and P. Burch, "Educational technology in india: The field and teacher's sensemaking," *Contemporary Education Dialogue*, vol. 16, no. 1, pp. 26–53, 2019. [Online]. Available: https://doi.org/10.1177/0973184918803184

[7] G. Friedland, W. Hürst, and L. Knipping, "The future of multimedia education and educational multimedia," in *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, ser. Emme '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 125–126. [Online]. Available: https://doi.org/10.1145/1290144.1290167

[8] S. Xu, T. Hu, and Q. Zhou, "A case study in jiuwuji minority middle school on rural teachers' educational technology training in terms of a switch of educational product," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 318–320.

[9] M. Qian, B. Zhao, and Y. Gao, "Exploring the training path of design thinking of students in educational technology," in *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*, Aug 2019, pp. 315–319.

[10] J. Hongyan and R. Jianfeng, "A brief analysis on the seminar of professional construction for modern preschool educational technology," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, April 2012, pp. 2262–2264.

[11] A. I. Taganov, A. N. Kolesenkov, D. A. Perepelkin, and D. S. Zhuravlev, "Management of educational risk on the basis of data mining in gis," in *2017 International Conference "Quality Management,Transport and Information Security, Information Technologies" (IT QM IS)*, Sep. 2017, pp. 577–580.

[12] C. Wei and L. Yuan, "Reflection on college informationized teaching model under the background of educational informationization," in *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*, Aug 2019, pp. 81–83.

[13] G. R. Iksanova, "Improvement of university education and educational service consumer interconnection in russia," in *2013 International Conference on Interactive Collaborative Learning (ICL)*, Sep. 2013, pp. 666–668.

[14] S. Tatiana Alexandrovna, "The role of distance educational technologies in the development of educational programmes in a network form: Experience and relevant problems," in *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*, Oct 2018, pp. 1–4.

[15] H. M. Fardoun, A. S. Mashat, and L. Gonzalez, "New subject to improve the educational system: Through a communication channel betwen educational institution-company," in *2013 Federated Conference on Computer Science and Information Systems*, Sep. 2013, pp. 709–712.

[16] T. A. Tabishev, M. V. Alikaeva, and A. L. Betuganova, "Electronic informational and educational environment and organization of the educational process of a modern university (on the materials of the kabardino-balkar state university)," in *2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT QM IS)*, Sep. 2019, pp. 569–572.

[17] Taifur. (2011) Raspberry pi presentation machine with remote control. Tomado de https://www.instructables.com/id/Raspberry-Pi-Presentation-Machine-With-Remote-Cont/.