

An Open Source and *Raspberry Pi* based Score Display System

Gabriele Salvato

A brief introduction

All my three daughters play volley in small teams of my city (Messina – Italy)



The economic support for all the teams' activities is modest and has to be carefully managed in order to maintain the teams alive. The same condition apply to many other small sport teams in our and neighborhood cities.

In all the gymnasiums they play there are different systems for displaying score and (sometimes) other related information. Often the teams have only cardboard displays (like the one in the figure)

that allow to show the score only



Since I had a powerful and inexpensive Raspberry Pi 3



laying around my

desk together with an old 32" Philips LCD TV



, I asked myself if it could be

possible to use them as a display system to show score information, as well as *slide-shows* and *videos* so as to encourage possible sponsors to give financial support to the teams.

I have then envisaged a (low cost) system that could be adapted to small gymnasiums (that will use a single *LCD* screen) as well as to bigger sport facilities in which many *LCD*'s would show, in a coordinated way, the different information.

Since I hate having messy cables laying around, I wanted to create a system that could work wireless so that it can be easily moved without too much trouble.

I ended up in realizing a system in which an Android tablet (the “*game director*”) controls, via WiFi, one or more “*score panels*”, composed by a Raspberry Pi connected to a HDMI capable monitor (or TV). Each *score Panel* is configurable in order to show the score information only or instead show score, slides and videos as dictated by the *game director*.

Each *score panel* can be equipped with a **Raspberry Pi Camera module V2** so as to show (under control of the *game director*) in real time what is happening on the game field, and the camera can



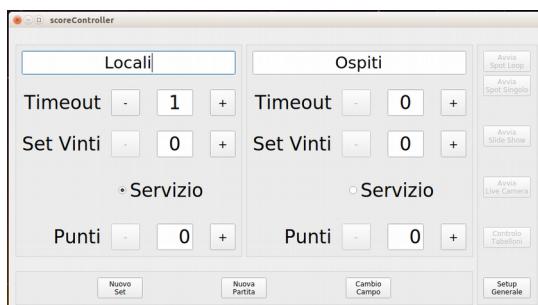
be mounted on a **Pan-Tilt module**

that, in turn, is controlled by the tablet.

Hardware needed

From a **hardware** point of view, at minimum, you need a **computer** (in my case an **Android tablet** equipped with a touch screen and endowed of WiFi capabilities but you can use a laptop if you wish) that will stay on the scorekeeper table and one or more “*score panels*” composed of a **Raspberry Pi** connected by a short **HDMI** cable to a **LCD TV** (or **monitor**).

One (and only one) Raspberry Pi has to be configured in such a way to realize a **Wireless Local Area Network (W-LAN)** acting as a sort of “*access point*”. The tablet plays the role of the *game director* and, together with all other Raspberry Pi, will connect to the **W-LAN** built up by the *access point* in order to send commands and exchange informations with all the *score panels*.



The game director GUI (italian version)



The Volley score panel (italian version)

If Volley is the only sport you are interested in that's all the needed hardware. In the following we will see how to add timing capabilities to the system so to cover other sports.

The Software

In each Raspberry Pi there must be installed a (fresh) copy of the **Raspbian** operating system that can be downloaded from:

<https://www.raspberrypi.org/downloads/>

There are many tutorials that details very well how to download and install the required operating system into a micro SD and I will not go further on this argument apart that I strongly suggest to use a micro SD with a **storage capacity not less than 16GB** and connect your Raspberry to the Internet, trough your router, **via an Ethernet cable**.

Once you have installed the operating system on your Raspberry Pi, let's be sure to have an updated version of the software by opening a terminal and issuing the following command:

```
$ sudo apt update && sudo apt upgrade -y
```

We need to modify some of the default setups of our Raspberry.

First of all, since all the informations will be exchanged via *WiFi*, we need to change, for security reasons, the default password of the user *pi*.

Click on the Raspberry icon  on your desktop and select:

Preferences → Raspberry Pi Configuration.

- From the “System” tab change the password and assign a different “*Hostname*” to each *score panel* (I use *panel-0*, *panel-1* and so on...).
- Check the “*To Desktop*” radio button (to boot directly into the Desktop environment) and the “*Login as user ‘pi’*” checkbox (to be logged automatically without having to enter the password every time).
- Depending on your display (or tv), you may need to change also the “*Resolution*” and “*Overscan*” options.
- Under the tab “*Interfaces*” enable “*SSH*” and, if you intend to use the *Raspberry Camera Module* and the *Pan-Tilt* system, enable the “*Camera*” and “*Remote GPIO*” capabilities.
- Since we will use the *GPU* for showing slide and videos, under the “*Performance*” tab increase the “*GPU Memory*” to 128K.

Close the configuration dialog. You will be asked to reboot: please do it now.

Then you need to clone the repository in which are stored the programs that build-up the *score panel* system. I suggest to clone the repository by using **git** so you can easily update the software as a new releases became available . If you don't have it already installed, open a terminal window and issue the command:

```
$ sudo apt install git
```

Then tell **git** who you are by writing (this is not strictly needed):

```
$ git config -global user.email "your.email@address"  
$ git config -global user.name "Your Name"
```

Obviously replace “**your.email@address**” with your actual email address and “**Your Name**” with your real name.

Now you are ready to clone the software repository. Open a terminal window and write:

```
$ git clone https://github.com/salvato/ScorePanel_executables.git
```

A new folder, **ScorePanel_executables**, will be created that, at present, contains three more folders:

- **Android**
which contains **ScoreController.apk**, the executable for the "game director" part of the system.
- **Raspberry**
which contains two executables: the **panelChooser** executable, responsible of realize the score panels for a few sports, and the **SlideShow** program, the one that take care of showing, on request, the slides. The system make use of the commonly installed program **omxplayer** to show short movies.
- **Ubuntu**
which contains the version of the programs that you can use on Ubuntu (but we will not cover the installation on Ubuntu in this document).

Copy the two programs contained in the **ScorePanel_executables/Raspberry** folder into the *pi* home directory.

The **panelChooser** program depends on some *qt5* libraries that can be already installed and on some other that can be missing. To check which libraries are missing you may issue the following command in a terminal window:

```
$ ldd ~/panelChooser
```

In my Raspbian version (stretch) two of the needed libraries are missing:

```
libQt5WebSockets.so.5 => not found
libQt5SerialPort.so.5 => not found
```

You can install such libraries by issuing the following command:

```
$ sudo apt install libqt5websocket5 libqt5serialport5
```

You can check if everything is working by launching the **panelChooser** program: Open a terminal and issue:

```
$ ~/panelChooser
```

A full screen black window should appears and a message should tell that the program is waiting to be connected to a server. Hit *Esc* in your keyboard and the window should disappear restoring your desktop.

If your Raspberry is not connected to a network the message will instead tell you that the program is waiting for a network connection. Again hit *Esc* in your keyboard to close the black window.

We have to **disable** Screen Blanking to prevent the disappearing of the score during the match. You need to edit a configuration file:

```
$ sudo nano /etc/lightdm/lightdm.conf
```

Search the line looking:

```
#xserver-command=X
```

and change it in:

```
# To disable screen blanking
xserver-command=X -s 0 -dpms
```

If we want to control the *pan* and the *tilt* of the Raspberry camera with two servos, we need to start the *gpio* daemon at boot time. We have to edit an other configuration file:

```
$ sudo nano /etc/rc.local
```

Insert:

```
/usr/bin/pigpiod &
```

just before the line containing:

```
exit(0)
```

If we are using the WiFi for the network connections we have to be sure that the power management does not place the WiFi interface in power save mode but, according to a post of *Dougie Lawson* in the Raspberry forum, in the built-in WiFi on a RPi3 the power management is always off even if the command:

```
$ iw dev wlan0 get power_save
```

will report that it is On.

Now it is time to tell our Raspberry to start automatically the program at boot so you don't need to start it any time you switch on the system. Issue the following command on a terminal window:

```
$ nano ~/.config/lxsession/LXDE-pi/autostart
```

and, since we don't want that a screen-saver can obscure the panel, comment the line

```
@xscreensaver -no-splash
```

changing it in:

```
#@xscreensaver -no-splash
```

Then insert a line to start at every reboot the panel control program:

```
/home/pi/panelChooser &
```

Save and reboot to check that everything has gone well.

To exit the program remember to hit *Esc* on your keyboard.

If you have a WiFi network up and running and don't want to use a dedicated WiFi for your system, you are almost done with your Raspberry, but I strongly suggest that you create your own WiFi network to separate the *score system* from all the other network traffic.

Setting your Raspberry to act as an “Access Point”

To configure your Raspberry as an access point you can follow the instructions at:

<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>

stopping at the paragraph “**ADD ROUTING AND MASQUERADE**” since we don't want to route any message out on internet: all the network traffic should remain inside the local network.

If you plan to have more than one *score panel* in your system, only one have to be configured to act as an access point, all the other must be connected to the network created by that one.

We have done with the *score panel* part of the system. It is time to setup the *game director* part.

Installing the Android App

Connect your Android tablet to your Raspberry and copy the file:

`~/ScorePanel_executables/Android/ScoreController.apk`

into the *SD card* of your tablet. Create also two new directories into the same SD card named respectively: *slides* and *spots*.

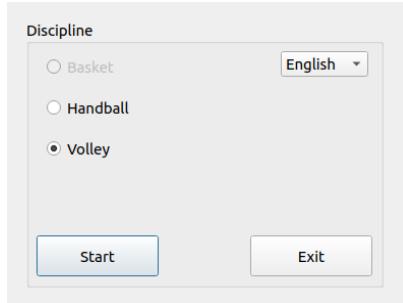
Copy all the slides that you want to show into the *slide* directory (only *jpeg* and *png* formats are currently recognized) and all the movies you want to play (only *mp4* files are recognized at present).

Disconnect your tablet from the Raspberry and open the file manager of your tablet. Navigate to where the **ScoreController.apk** file has been copied and click on its icon to install it.

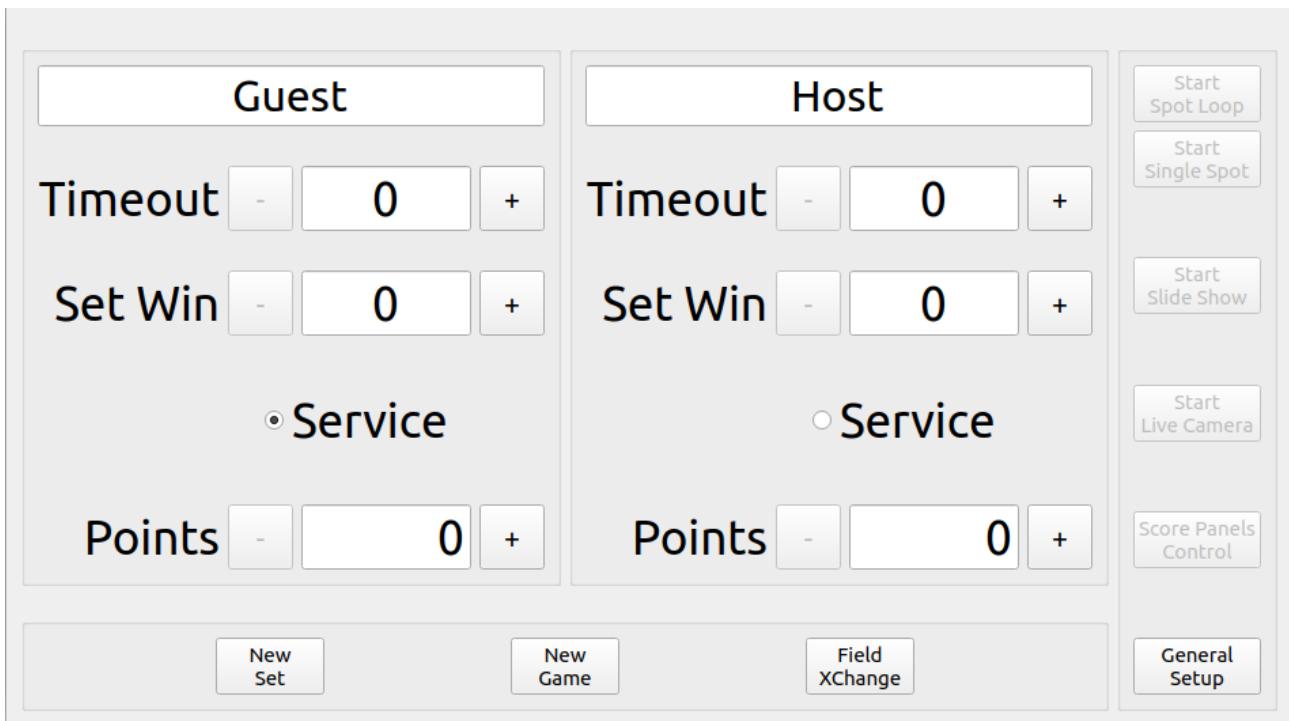
It is possible that the security options on your tablet will prevent to install the program since it comes from an unknown source. You should allow to install also from unknown sources just for the time to install the program and block again once the program has been installed.

A new applet icon should appear into your tablet. Be sure to connect the tablet to the same network of the *score panels* and then click on the program icon: **your new score system should start !**

A dialog like the one depicted should appears allowing to choose the language (at present only Italiano and - a poor translation of – English) and the sport discipline you are interested in. At present only Volley can be run without further hardware. In a follow-up we will consider disciplines that requires timing information.



Select the discipline and click *Start*: the dialog will close and a new window will be shown:



The very first time the *App* will run it will ask to locate the two directories containing the slides and the movies. Please select the right directories and go on.

All the controls in the window should be really intuitive so I'll not explain their meaning. Please feel free to experiment and enjoy your new system.

The Software Sources

As said in the title all the sources are freely available and I hope to receive contributions from the many brave developers around for improving the system.

The programs are written in **c++** using **Qt**. You can clone the following repositories:

<https://github.com/salvato/scoreController.git>
<https://github.com/salvato/ScorePanel.git>