An Approach to Raspberry Pi Synchronization in a Multimedia Projection System for Applications in Presentation of Historical and Cultural Heritage

Nemanja D. Savić, Dušan B. Gajić, Radomir S. Stanković

CIITLab, Dept. of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia
`nemanja.d.savic@gmail.com`, `dusan.b.gajic@gmail.com`, `radomir.stankovic@gmail.com`

Abstract. This paper discusses an implementation of a multimedia projection system with two synchronized video sources, based on the Raspberry Pi single-board computer systems. The system is primarily intended for various projections in presentation of historical and cultural heritage, although it can be used for many other related purposes. The considered multimedia system has two projection surfaces - a horizontal projection plane and a 45° inclined special glass projection plane which creates the hologram-like Pepper's Ghost effect. Each plane displays a separate high-definition video obtained from the corresponding Raspberry Pi. Synchronization of presentations is achieved using custom connection through the general purpose input/output (GPIO) connector of the Raspberry Pi and the *wiringPi* interface library. As valuable aspects of the system, we can point out the following. From the practical point of view, the system has low cost due to the use of Raspberry Pis. Generation of a holographic effect, as well as larger flexibility in creating the contents of the presentations due to the two projection surfaces, are additional attractive features.

Keywords: Raspberry Pi, multimedia projection systems, Pepper's Ghost, GPIO communication and synchronization.

1 Introduction

Various multimedia projection systems are widely used in the presentation of historical and cultural heritage. Besides attracting visitors of related exhibitions, their application is, more importantly, aimed at providing a detailed and thorough insight into presented data and illustration of objects or events. Many of these multimedia systems consist of several different sources of audio and video signals. Therefore, spatial and time synchronization of their projections is an essential task in order to achieve the desired visitor experience and perception of presentations.

In this paper we discuss a solution to the problem of synchronization and communication between two Raspberry Pi computer systems, used in a multimedia projection system developed by ARhiMedia group from the Faculty of Electronic Engineering, University of Niš, Serbia [1]. The projection system considered in this paper is created to allow an improved presentation of cultural and historical heritage through the application of modern information and multimedia technologies. It is registered as a technical solution at the Faculty of Electronic Engineering of the University of Niš, Serbia, as „The System for Multimedia Presentations - An Inclined Projection Plane“[1].

The system consists of two projection surfaces - a 45° inclined special glass plane, which creates a three-dimensional hologram-like Pepper's Ghost effect[1], and a horizontal plane. It also features a monitor display, a video projector, a wireless keyboard, and two Raspberry Pi computer systems for video and audio signal projections. These Raspberry Pi systems are connected through the general purpose input/output (GPIO) connector for the purpose of mutual communication and synchronization. A detailed description of the system can be found in [1].

The system offers flexibility in creating content for multimedia presentations, since it features both the Pepper's Ghost effect and has two different projection surfaces. These attributes, however, require careful synchronization and communication between the two Raspberry Pis used as sources of video signal displayed on each of the surfaces. In this paper, we describe a solution of this synchronization task as implemented in the considered multimedia projection system.

The paper is organized as follows. In Section 2 we describe the details of the application of Raspberry Pi computers in the considered projection system. The proposed solution to the problem of communication and synchronization between the two Raspberry Pis is presented in Section 3. In the closing section, we discuss the main conclusions and offer some directions for future work.

2 Application of Raspberry Pi in the Projection System

The Raspberry Pi single-board low-cost computer systems are based on a single-core ARM central processing unit (CPU). They were primarily developed for education purposes in computer science and engineering [7]. However, since their introduction in 2011 and due to their low price and characteristics such as the possibility to connect to various sensors and cameras, they have found numerous other applications [4, 7]. In 2015, Raspberry Pi 2 with a quad-core ARM Cortex-A7 CPU was released which has even further extended the possibilities for new applications.

In the presented projection system, two Raspberry Pis are used to project two distinct video signals on the horizontal plane and the inclined plane. Source videos are stored on class 10 Secure Digital (SD) cards in the Moving Pictures Experts Group MPEG-4 format. The H.264 compression is used for the videos which are in the FullHD (1920×1080 pixels) resolution and have 25 frames per second. Each source video is uniquely identified by its ID number. Hardware video decoding is used

---

[1] Pepper's Ghost is an optical technique named after John Pepper, a professor at the Royal Polytechnic Institute London who popularized it in 1860s [3]. It is often used in concerts, theatres, museums, and theme parks for creating a hologram-like presentation effect using a glass-plate reflection of a scene or video hidden from observers [2].

through the *omxplayer* tool which is a standard part of the *Raspbian* operating system (OS), built around *Linux* kernel and used on the devices [8].

3 Raspberry Pi Communication and Synchronization

For the purposes of synchronization, the Raspberry Pis are organized in a master-slave system and connected via a custom-made serial connection through the General Purpose Input Output (GPIO) connector as shown in Figure 1. On the system boot, a set of *Bourne-Again Shell* (BASH) scripts [8] is implemented on both the master and the slave device to start the presentations in the way illustrated in Figure 2.

The master device runs two BASH scripts. The first script (*masterScript.sh*) controls the input and selection sent from the wireless keyboard, used to input the video ID to the system, as well as for playing and halting the presentation. The second script (*syncScript.sh*) performs synchronization by sending the allow signal (GO/HALT), the binary video code, and by waiting for the acknowledgement (ACK). The first script starts automatically after Raspberry Pi boots, which is done by adding the *sudo /home/pi/~/masterScript.sh line to the /etc/profile* file of the Raspbian OS.

The slave device runs a single BASH script called *slaveScript.sh*. This script checks for allow and halt signals (GO/HALT) and decodes the received binary code of the video ID. This script also starts automatically after the slave Raspberry Pi boots which requires adding *sudo /home/pi/~/slaveScript.sh* to the file */etc/profile* of the *Raspbian OS*.

Fig. 1. The Raspberry Pi computer systems used in the presented projection system with custom serial GPIO connection for synchronization.

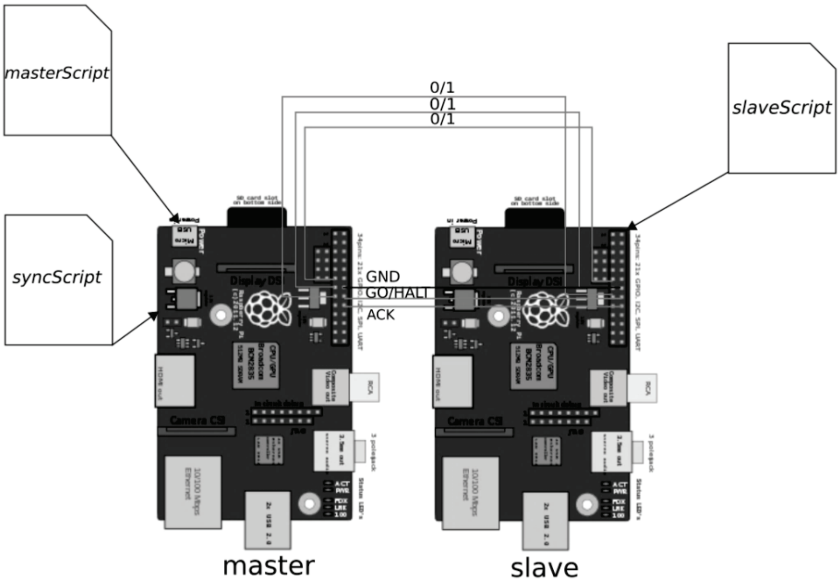

Fig. 2. Master and slave Raspberry Pis with corresponding scripts and signals (GND is ground).

The used Raspberry Pi model B has a GPIO connector with 26 pins which can be used for signal transmission [5, 7]. Figure 3 presents the pin layout of this GPIO connector. Eight are general purpose programmable digital input/output pins which can be used as either the input or the output. Newer Raspberry Pi B+ and Pi 2 models have 14 additional GPIO pins, but the presented solution remains applicable on these systems [5]. The *wiringPi* interface library is needed in order to use the *gpio* command for pin access on Raspberry Pi [6]. Notice that the labeling of GPIO pins in this library differs from the physical labeling [5].

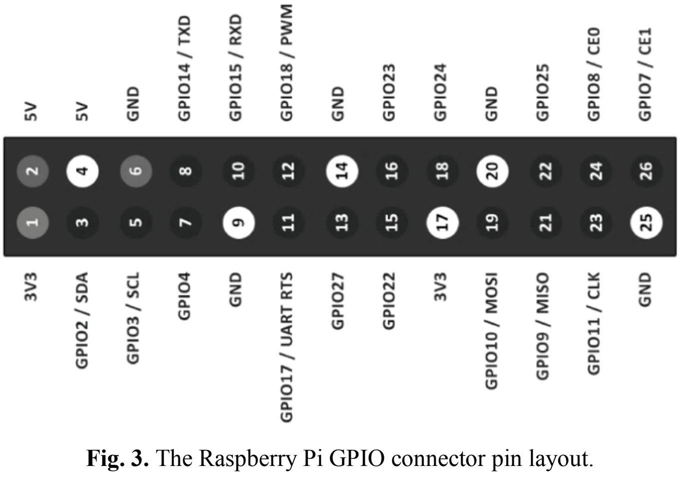

Fig. 3. The Raspberry Pi GPIO connector pin layout.

In the case of the presented projection system, communication between the devices is realized through pins 14, 16, 18, 11, 13, and 15 (where pin 14 is ground, 16 and 18 are GPIO 23 and GPIO 24, and 11, 13, and 15 are GPIO 17, GPIO 27, and GPIO 22, respectively), as shown in Figure 4.

For synchronization purposes (GO/HALT and ACK), we use GPIO pins 23 and 24, while the information about the selected video source is transferred using pins 17, 22, and 27. Since the '000' code word is used for signaling that no video is selected, this allows up to $2^3 - 1 = 8$ different source videos.

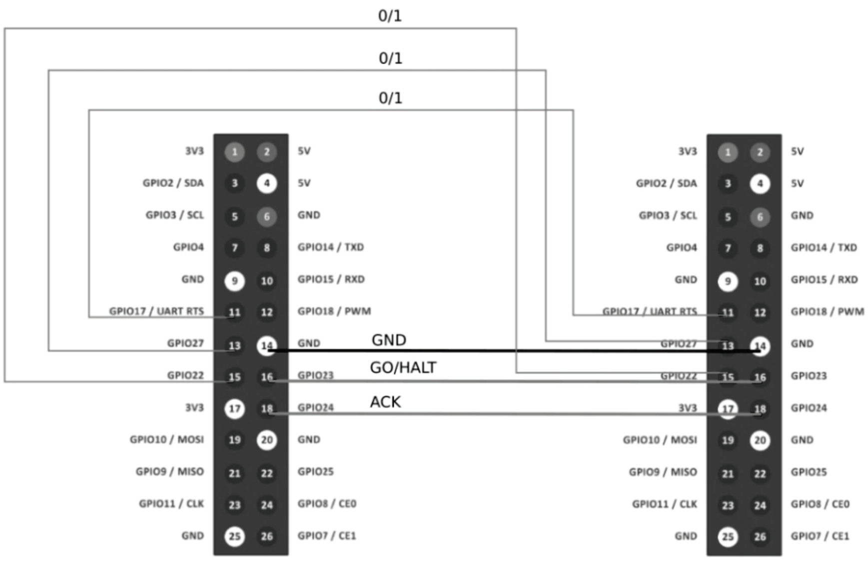

Fig. 4. The layout of the custom GPIO connection used in the presented solution.

The master reads input from the keyboard which represents the ID of the selected video. Depending on the input number, it sends the corresponding binary code to pins GPIO 17, GPIO 27, and GPIO 22, by setting logical one (3.3V) or logical zero (0V). After sending the video ID, master sets GPIO 23 to logical one as allow signal to the slave Raspberry Pi, and then enters into a loop in which it checks the state of the GPIO 24 pin on which the slave Raspberry Pi acknowledges that it started working. The master device leaves this loop only after it detects that the pin 24 is set to logical one after it starts playing the selected video. When the presentation needs to be stopped, master device sets GPIO 23 to logical one as stop signal, and sends code word '000' through pins 17, 27, and 22.

The slave device runs a loop in which it checks the state of the GPIO 23 pin, waiting for the allow signal from the master. It leaves the loop only after it detected logical one (3.3V). After exiting the loop, it reads the binary code of the selected video through pins GPIO 17, GPIO 27, and GPIO 22, and, through setting the GPIO 24 to logical one, reports to the master device that it started the video. Afterwards, slave device continues to check the state of the GPIO 23 pin for the presence of the stop signal.

Master device signals to slave when to start playing the video and then waits for the acknowledgment from the slave before itself starts playing. Both devices first need to reset their GPIO pins (initial state requires all the pins to be set as inputs).

The sequence diagram of communication between the master and slave device is illustrated in Figure 5.

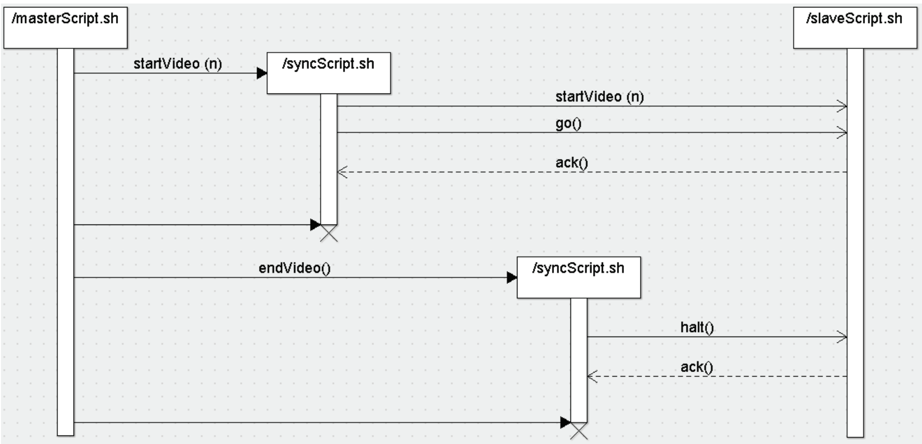

Fig. 5. Sequence diagram of the communication between the master and the slave devices.

4 Conclusions and Future Work

The paper addresses the problem of synchronization of audio and video signals in multimedia projections systems for applications in cultural and historical heritage. In particular, we discuss the synchronization of two signal sources in a low-cost multimedia projection system which can generate the hologram-like Pepper's Ghost effect. The system has low cost primarily because of the use of Raspberry Pis. In addition to producing a holographic-like three-dimensional visual effect, it also offers large flexibility in creating the contents of the presentations due to the presence of two projection surfaces.

The solution to the synchronization task presented in the paper is developed using two Raspberry Pi model B single-board computers. The two Raspberries are connected in a master-slave system and, on the system boot, each of them executes the corresponding purposely-written BASH scripts. They are synchronized using custom serial communication based on the available GPIO connector and the *wiringPi* programming library.

Plans for future development of the system include adding motion sensors or sonars connected to the master Raspberry Pi for starting and suspending the presentation when the visitor approaches and steps away from the system, respectively. Instead of the currently-used wireless keyboard, movement detection devices such as Microsoft

Kinect or Leap Motion can be added to the system for selecting different presentations for playback, controlling various sections and objects within presentations, and choosing the language of presentation.

Acknowledgments

The research reported in this paper is partly supported by the Ministry of Education and Science of the Republic of Serbia, projects III44006 (2011-2015) and ON174026 (2011-2015).

References

1. Stanković, R. S., Tatić D., Gajić D. B., Gajić, N. B., Tatić, D., Ivanović, S., "System for multimedia presentations – An Inclined Projection Plane" (in Serbian), Technical Solution, Report No. 07/05-007/14-001, Faculty of Electronic Engineering, University of Niš, Serbia, January, 2014.
2. Shein, E., "Holographic Projection Systems Provide Eternal Life", Communications of the ACM, vol. 57, no. 7, July 2014, pp. 19-21.
3. Pepper's Ghost – article at Wikipedia, available at: https://en.wikipedia.org/wiki/Pepper%27s_ghost, last accessed on July, 8, 2015.
4. Gajić, D. B., Tatić, D., "A Comparative Analysis of Raspberry Pi and Blueberry TV CAT Computer Systems for Multimedia Applications" (in Serbian), in Proc. of the XIX Conf. YU INFO, CD, pp. 446-450, March, 3 - 6, 2013, Kopaonik, Serbia, 2013.
5. Johnson, M., Raspberry Pi [R2] + Extension Board v2.1 Pinout, available at: https://howto8165.wordpress.com/2014/08/15/raspberry-pi-pinout/, August 2014, last accessed on July 8, 2015.
6. wiringPi - GPIO Interface Library for the Raspberry Pi, available at: http://wiringpi.com/, last accessed on July 8, 2015.
7. Monk, S., Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly, USA, 2014.
8. C. Abing, bash Cookbook: Solutions and Examples for bash Users, O'Reilly, Sebastopol, CA, USA, 2007.