

# A synchronized Scheduler/Reminder Application for Philips 2KTV with Joint SPACE platform

<sup>1</sup>Mohan Kumar J, <sup>2</sup>Leo K Yohan, <sup>3</sup>Sanketh S Shetty, <sup>4</sup>Soumya Mercy Varghese, <sup>5</sup>Sundaresan C, <sup>6</sup>Chaitanya C.V.S

School of Information Sciences, Manipal University.

1 mohuj@gmail.com , 2 leokyohan@gmail.com, 3 chintooshetty@yahoo.com, 4 soumyavarghese89@gmail.com, 5 sundaresan.c@gmail.com, 6 chaitucvs@gmail.com

**Abstract** - JointSPACE is an Open Source project that allows every user/supplier to develop applications for Philips TV displays. JointSPACE is based on the SPACE architecture which was developed by Philips to ease internal development. At some point of time, Philips decided to open its architecture to allow users or anyone to develop code for the TV target. In this paper an application developed is developed based on the JointSPACE platform. The idea is to synchronize the Reminders stored on the mobile phone or any wireless device to the TV. The reason for the development of this application is majority of the people watching the Television tend to forget their schedules while watching Television. Since remote wireless devices like mobile phones and laptops are most common these days, people normally have their schedules stored in these devices. So as a solution for the above mentioned issue, we propose a synchronizer for Scheduler application for the Television that will work with a remote wireless device.

**Keywords** - Philips JointSPACE TV, Directdb, Voodoo, Mobile Applications.

## 1. INTRODUCTION

Television (TV) is an essential entertainment media in common man's life. Presently smartTV started emerging and in the next decade, the user space will be more for it. 25% of the TV shipped during 2011 are smart TV which can get connected to the other devices. This is expected to grow 60% year by year [1]. Even android based TV are into market. These smart TV run a particular software platform for running the applications. One such software platform is JointSPACE from Philips. JointSPACE is an Open Source project that will allow every user/supplier to develop applications for Philips TV displays. JointSPACE is based on the SPACE architecture which was developed by Philips to ease internal development. Similar android mobile stack JointSPACE is also made open for the user to customize the applications or even developers or researchers can develop applications to improve the human life better [2][3]. The system is shown in Figure 1. The idea is to synchronize the reminders

stored in the mobile phone or any other wireless devices to TV and display the message on the TV.



Figure 1. System

## 1.1 JointSPACE

JointSPACE facilitates mainly 2 aspects:

- i. Integration of applications made by suppliers.
- ii. Integration of applications made by customers. JointSPACE addresses this by opening and extending the current TV architecture. [2]

Some of the features of JointSPACE are:

JointSPACE proposes a single platform to develop applications. The platform may be any Linux PC or device capable of running Linux/DirectFB technologies. JointSPACE publishes the essential TV APIs used in the SPACE architecture. JointSPACE provides portable prototyping software that includes and illustrate the essential of the SPACE architecture. [2]

JointSPACE extends the TV architecture to allow:

1. Executing TV applications on a remote system, rendering and being controlled on the TV

2. Executing application on a remote system, controlling the TV APIs remotely [2]

JointSPACE continuously provide new technologies/libraries to ease and improve the development of new applications. JointSPACE extends the TV API to allow controlling more TV functionalities. As JointSPACE is based on DirectFB technologies; following DirectFB packages are used

- DirectFB 1.4
- SaWMan 1.4
- FusionDale 0.8.1.

The JointSPACE exposes 3 core APIs apart from these packages for developers, as shown in Figure 2.

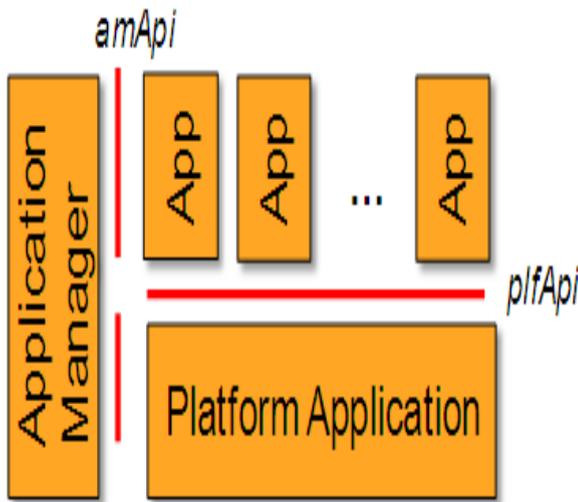


Figure 2. JointSPACE Architecture - 3 Core API

**plfApi:** the Platform control API used by applications to control the Hardware platform features

**papi:** an internal "clean" API toward Hardware suppliers that can be used to glue existing hardware functionalities

**amApi:** the Application manager API, used to communicate between applications.

### 1.2 Connection Management

The connection between the remote device and the television is established by a set of processes. Firstly the application manager api, amapp, starts the communication, after the device discovery. Then the remote device calls the directfbinit function, so that the jsapp master can fork another process jsapp1. Also the process will be added to the application manager. Then the remote device initiates directfbcreate and createwindow functions. Then a window will be created inside the application manager for the particular process. Then the remote device requests requestfocus, through jsapp1 so that the application manager focuses to the amapp. The connection Management is shown in Figure 2.

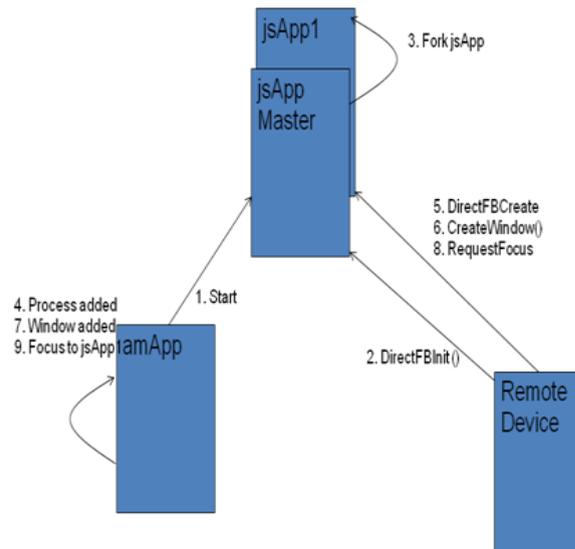


Figure 3. Connection Management

### 1.3 DirectFB – Direct Frame Buffer

A thin library that provides hardware graphics acceleration, input device handling and abstraction, integrated windowing system with support for translucent windows and multiple display layers, not only on top of the Linux Frame buffer Device. It is a complete hardware abstraction layer with software fallbacks for every graphics operation that is not supported by the underlying hardware. DirectFB adds graphical power to embedded systems and sets a new standard for graphics under Linux [6].

### 1.4 JointSPACE Simulator

The JointSPACE simulator allows to experiment with the SPACE architecture on a Linux PC [4][5]. A snapshot of the JointSPACE simulator is shown in Figure 4. It is splitted into various packages organised into sub-directories.

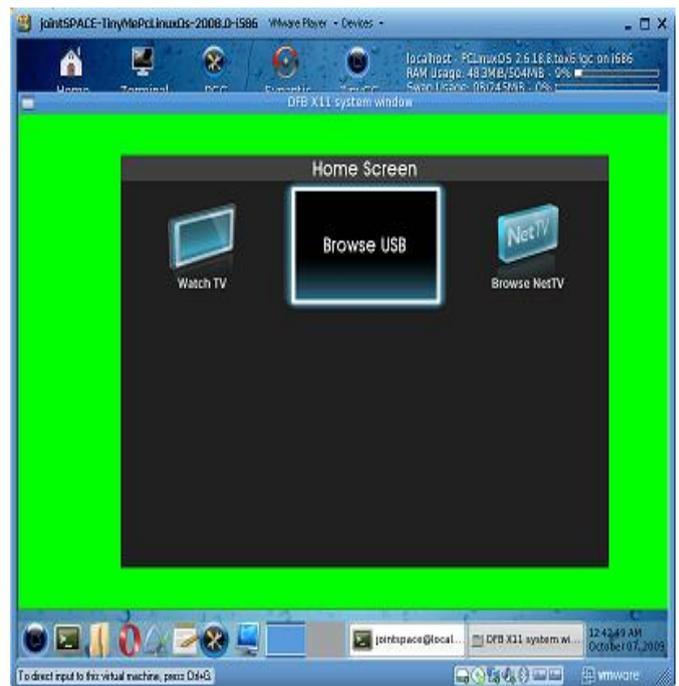


Figure 4. JointSPACE Simulator [4]



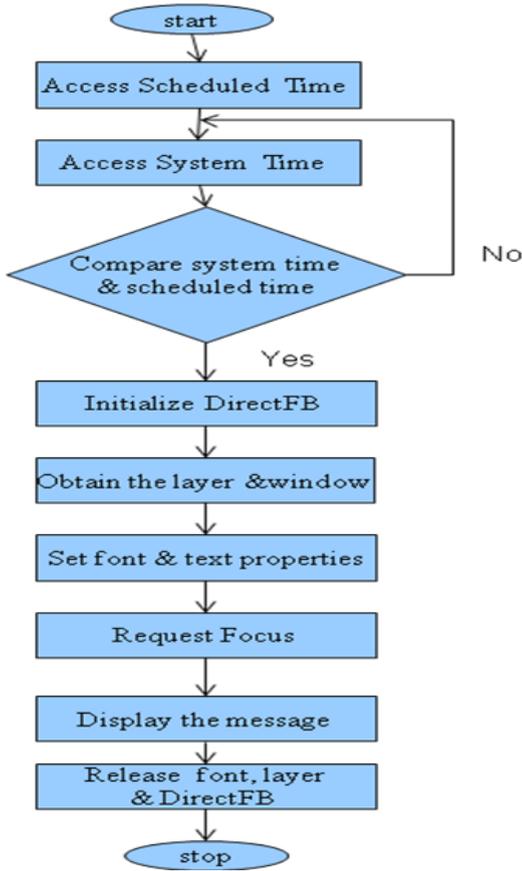


Figure 6. Flowchart of the Implementation

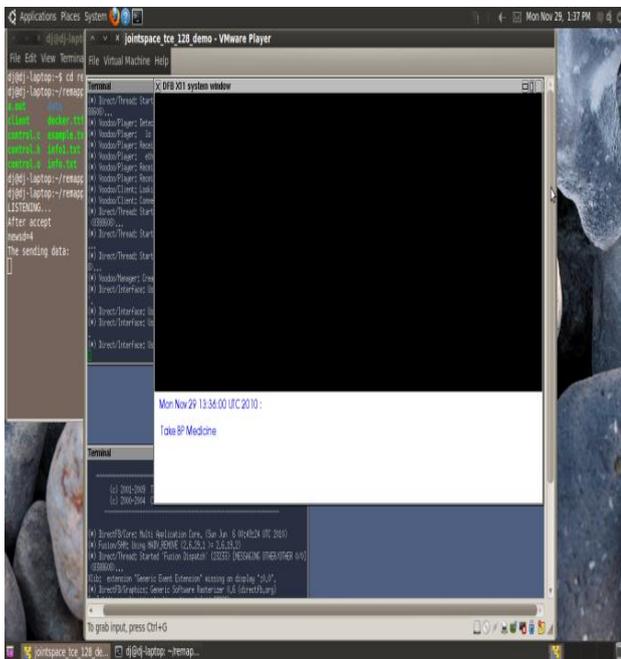


Figure 7. Output of the schedules on the JointSPACE Simulator

### 3. CONCLUSION AND SCOPE

Developed the scheduler application and also developed on the iPhone environment. The application has the capability to store the schedules in the calendar database of iPhone. The future work in the project is to inject the scheduled events into the Television’s default reminder, so that the remote device does not have to be connected with the Television always. Whenever the iPhone is wirelessly connected to the Television all the events in the phone will be injected to the default reminder of the Philips Television. So the application can be very useful in the real-time scenario. Also the future work includes the development of the scheduler application on Android phone and also cross platform development.

### 4. ACKNOWLEDGEMENT

We thank Prof.(Dr) Harishchandra Hebbar, Director School of Information Sciences, Manipal University for supporting us in this project. Especially we have to thank Dr. A Narendranath Udupa, Philips Research, Bangalore and Mr. Manjunatha Maiya, Sr. Project Manager, MU - BoP, Philips Pvt.Ltd for providing us the JointSPACE platform training and donating a Philips 2K smart TV for execution of projects.

### 5. REFERENCES

- i. “The Who, What, When, Where, Why, and How of Connected TV & Advertising”, YuMe White Paper, August 2012.
- ii. [http://www.yume.com/sites/default/files/YuMe\\_Connected\\_TV\\_Whitepaper.pdf](http://www.yume.com/sites/default/files/YuMe_Connected_TV_Whitepaper.pdf)
- iii. <http://foundation.webinos.org/deliverable026target-platform-requirements-and-ipr/26-nettv-fraunhofer/> ( March 2014)
- iv. <http://JointSPACE.sourceforge.net/> ( March 2014)
- v. <http://sourceforge.net/projects/JointSPACE/> (March 2014)
- vi. [http://sourceforge.net/apps/mediawiki/JointSPACE/index.php?title=JointSPACE\\_Simulator](http://sourceforge.net/apps/mediawiki/JointSPACE/index.php?title=JointSPACE_Simulator) (March 2014)
- vii. <http://directfb.org/> (March 2014)
- viii. ” System architecture for virtual world interfacing with TV platform” - Virtual world interfacing with TV platforms Article, [http://wg11.sc29.org/mpeg-v/?page\\_id=298](http://wg11.sc29.org/mpeg-v/?page_id=298) ( March 2014)