JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ELECTRASYNC VISION

Enabling Remote Content Display on TV Screens Using Raspberry Pi as a Kiosk.

Dinesh Deore
Guide, Rizvi College of Engineering (Mumbai , India),
Ansh Kanhaiyalal Tiwari
Student, Rizvi College of Engineering (Mumbai, India),
Suchit Prashant Sawant
Student, Rizvi College of Engineering (Mumbai, India),
Faizan Hussain Abid Hussain Shikalgar
Student, Rizvi College of Engineering (Mumbai, India),
Salman Mo Haider Mo Ali
Student, Rizvi College of Engineering (Mumbai, India),

Abstract:

This paper presents a novel system for remote image display on television screens, using the Raspberry Pi as a cost-effective and versatile kiosk solution. The system uses a Raspberry Pi computer to render images from remote sources onto connected TV screens, enabling dynamic content delivery. A custom software framework is developed, incorporating a lightweight image rendering application and secure communication protocol for receiving image data from remote servers or cloud-based services. The system is scalable, supporting multiple Raspberry Pi units and screens for larger installations and offers flexibility in content management. The system has shown promising results in retail advertising, educational content display, and personal photo galleries. Potential enhancements include integration with IoT devices for interactive displays and the use of AI algorithms for adaptive content delivery. This research contributes to the field by providing a low-cost, efficient, and scalable solution for remote image display, showcasing the Raspberry Pi's potential for commercial digital signage solutions.

Additional Keywords and Phrases: Raspberry Pi, kiosk, Remote Display, TV Display, Image Slideshow.

1 INTRODUCTION :-

The advancement of technology has revolutionized the way we share and display visual content. In today's interconnected world, there is a growing need to remotely showcase images and media on a variety of devices. Whether for business purposes, digital signage, or personal use, the ability to display images on a TV screen in a different location, without being physically connected to the same Wi-Fi network, presents numerous practical applications.

This project introduces a novel solution that allows remote image display on a TV using a Raspberry Pi configured as a kiosk. Traditionally, sharing images on a TV screen required physical presence or intricate network configurations. However, by leveraging the Raspberry Pi's capabilities and the power of web technology, this project offers a versatile and cost-effective approach for achieving remote image display.

The primary objective of this project is to enable users to conveniently select and display images on a TV screen located at a different place, regardless of the network the TV and Raspberry Pi are connected to. This project not only addresses the technical challenges of setting up cross-network communication but also provides a practical solution for a wide range of applications, including digital signage, remote presentations, and sharing images with family and friends. The ability to remotely control and update displayed images from virtually anywhere with an internet connection brings convenience and flexibility to the realm of image sharing and display.

In the ever-evolving landscape of technology, the demand for versatile and accessible solutions for remote content display is on the rise. Traditional methods often require devices to be connected to the same Wi-Fi network, limiting the flexibility of sharing visual content. Our research aims to address this limitation by leveraging the capabilities of a Raspberry Pi, a cost-effective and versatile computing platform.

Literature Survey:

A literature survey on the topic of screen sharing from a laptop to a TV using a Raspberry Pi encompasses exploring various technologies, protocols, and software implementations that facilitate this process. Screen sharing, in this context, refers to the ability to project the display contents of a laptop onto a TV screen, effectively using the TV as an extended or mirrored display. The Raspberry Pi, a versatile and compact single-board computer, can be utilized as an intermediary device to enable this functionality. This survey will outline the key areas of research, methodologies, technologies involved, and potential applications.

Raspberry Pi devices have gained popularity for their affordability, compact size, and flexibility, making them suitable for a wide range of computing projects, including media centers, home automation, and as a tool for learning programming and computer science concepts. Their capability to run full versions of Linux and support for various peripherals and networks makes them an excellent choice for screen sharing projects.

Several technologies and protocols can be used to enable screen sharing from a laptop to a TV through a Raspberry Pi, including:

- VNC (Virtual Network Computing): Allows for remote control of another computer using a network connection. VNC servers and clients are available for almost all operating systems, and the Raspberry Pi can run a VNC server, enabling screen sharing to the Pi-connected display.
- RDP (Remote Desktop Protocol): Similar to VNC, RDP provides a graphical interface to another computer. While more common in Windows environments, RDP clients and servers are available for Linux, including Raspberry Pi.
- DLNA (Digital Living Network Alliance): A set of protocols for sharing digital media between multimedia devices. While primarily used for media streaming, some implementations allow for screen mirroring.

- Miracast: A wireless standard that allows for sending video and sound from one device to another. It requires both the sending and receiving devices to support Miracast.
- HDMI over IP: Technology that allows HDMI signals to be transmitted over IP networks. This can be used for screen sharing when the Raspberry Pi is set up with an HDMI over IP transmitter or receiver setup.

Challenges are Latency like screen sharing and presentations having a delay between the action on the source device and screen. Ensuring the transmitted screen shares the same resolution and aspect ratio as the TV without quality loss requires proper configuration and sufficient network bandwidth. Transmitting sensitive information over networks requires encryption and secure protocols to prevent unauthorized access.

Methodology:

Raspberry Pi :-

The Raspberry Pi is a series of small, single-board computers developed in the UK by the Raspberry Pi Foundation. Initially designed for educational purposes, to teach computer science in schools and in developing countries, it has gained immense popularity among hobbyists, makers, and professionals. Its affordability, versatility, and ease of programming make it suitable for a wide range of applications, from home automation and hobbyist projects to more complex uses like media centers, web servers, and digital signage.

Kiosk :-

Kiosks are designed to offer users specific information or a specific experience, while preventing access to any other activities on the device. They are often found in airports, shops, hospitals, cafes, and museums — any location where people need easy access to information or services like timetables, waiting times, product information, directions, self check-in machines, and so on. Kiosk mode on your Raspberry Pi allows you to boot straight into a full-screen web page or an application without using the desktop environment. It's the foundation for many different projects where you want to display information for a dedicated interaction with a user.

Remote Display:-

Remote Display technology allows for the control and management of digital screens from a remote location. This capability is essential for managing digital signage, where content needs to be updated frequently and may be deployed over a wide area or in multiple locations. Remote display technology enables centralized content management, scheduling, and deployment across various displays, reducing the need for physical intervention and allowing for dynamic content updates.

TV Display:-

A TV display, in the context of digital signage, refers to using a television screen to display digital content publicly. This could be in the form of advertisements, information boards, menus, or any visual content intended for public or targeted audiences. TV displays are chosen for their availability, ease of setup, and high-quality visual output, making them an accessible option for small businesses and educational institutions to large corporations for their digital signage needs.

Image Slideshow:-

An image slideshow is a presentation of a series of images displayed in a sequential or randomized order, often used for advertising, instructional, or entertainment purposes. In digital signage, image slideshows are a common format for content, allowing for the dynamic display of multiple images, messages, or promotions without requiring constant manual updates. This format can be particularly effective in capturing and retaining the audience's attention, conveying information, and enhancing the visual application in graphical methods.Implementation:-

Components:-

- **Raspberry pi**
- **Power Supply**
- **microSD card**
- **Adapter to connect your microsd card with usual computer**
- **Display Cable to connect your raspberry pi to your monitor.**
- **Monitor**

Choose the right Raspberry Pi :-

Kiosk mode works on all Raspberry Pi models. For this tutorial, we'll be using a Raspberry Pi 4 Model B.

Configure your Raspberry pi :-

To begin, **follow the Getting Started documentation to set up your Raspberry Pi.** For your operating system, choose **Raspberry Pi OS (32-bit)** to run headless (without a mouse and keyboard).

During the OS customisation stage, edit settings as follows:

- Enter a **Hostname** of your choice.
- Enter a **Username** and **Password** for authentication purpose.
- Check the box next to **Configure wireless LAN** so that your Pi can automatically connect to Wi-Fi.
- Enter your network **SSID**(name) and **Password** , you can find these in your Wi-Fi settings or on a sticker on your router.
- Check the box next to **Enable SSH** so that we can connect to the Pi without a mouse and keyboard.

Remotely Connect to Raspberry pi :-

SSH allows you to wirelessly connect to your Raspberry Pi, eliminating the need for a keyboard and mouse. It's perfect if your Raspberry Pi is located in a hard-to-reach location, like the back of your television.
To SSH into the Raspberry Pi, you'll use the hostname you set in Imager. If you have issues connecting using this method, you may want to use the Raspberry Pi's IP address instead.For more information about finding your IP address and remote accessing your Raspberry Pi, see the remote access documentation.

Connect via SSH:-

Open a terminal session on *your usual computer*. To access your Raspberry Pi via SSH, run the following command, replacing <username> with the username you chose in Imager:

$ ssh <username>@pi-kiosk.local

$ ssh <username>@pi-kiosk.local
The authenticity of host 'pi-kiosk.local (fd81:b8a1:261d:1:acd4:610c:b069:ac16)' can't be established.
ED25519 key fingerprint is SHA256:s6aWAEe8xrbPmJzhctei7/gEQitO9mj2ilXigelBm04.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/
[fingerprint])? yes
Warning: Permanently added 'pi-kiosk.local' (ED25519) to the list of known hosts.

<username>@pi-kiosk.local's password:
Linux pi-kiosk 6.1.21-v8+ #1642 SMP PREEMPT Mon Apr 3 17:24:16 BST 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 24 09:41:00 2023
<username>@pi-kiosk:~ $

When asked for your password, use the password you created in Raspberry Pi Imager.

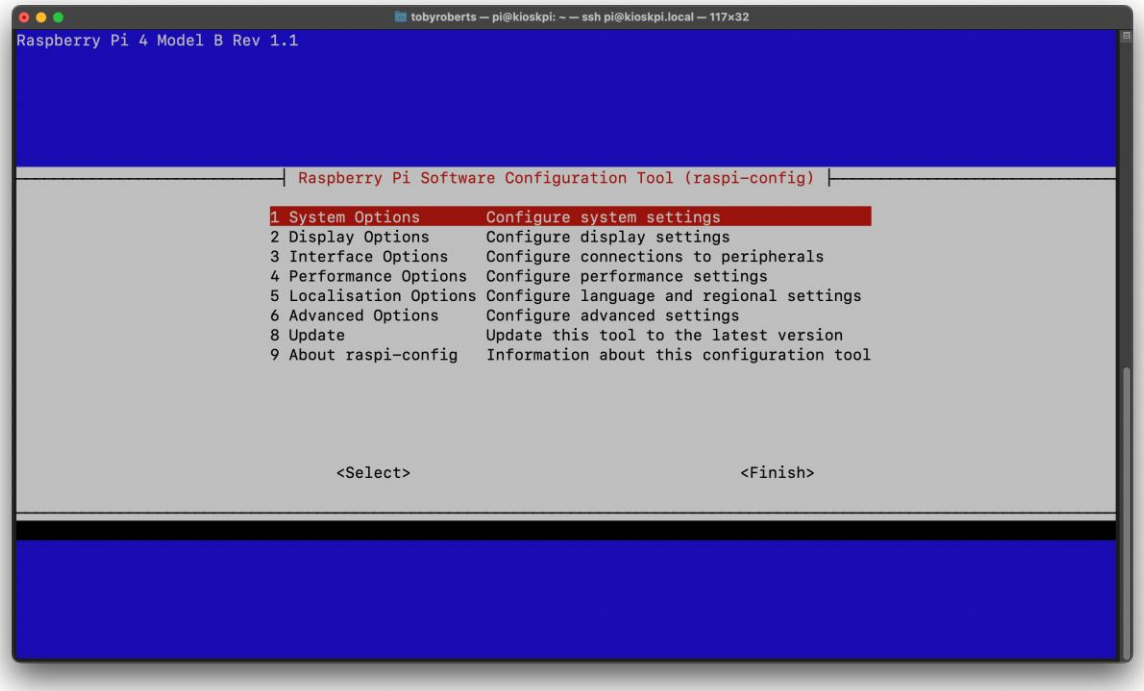
Set Up Kiosk Mode:-

This tutorial requires one additional piece of software, wtype, which simulates keyboard activity. To install it, run the following command:

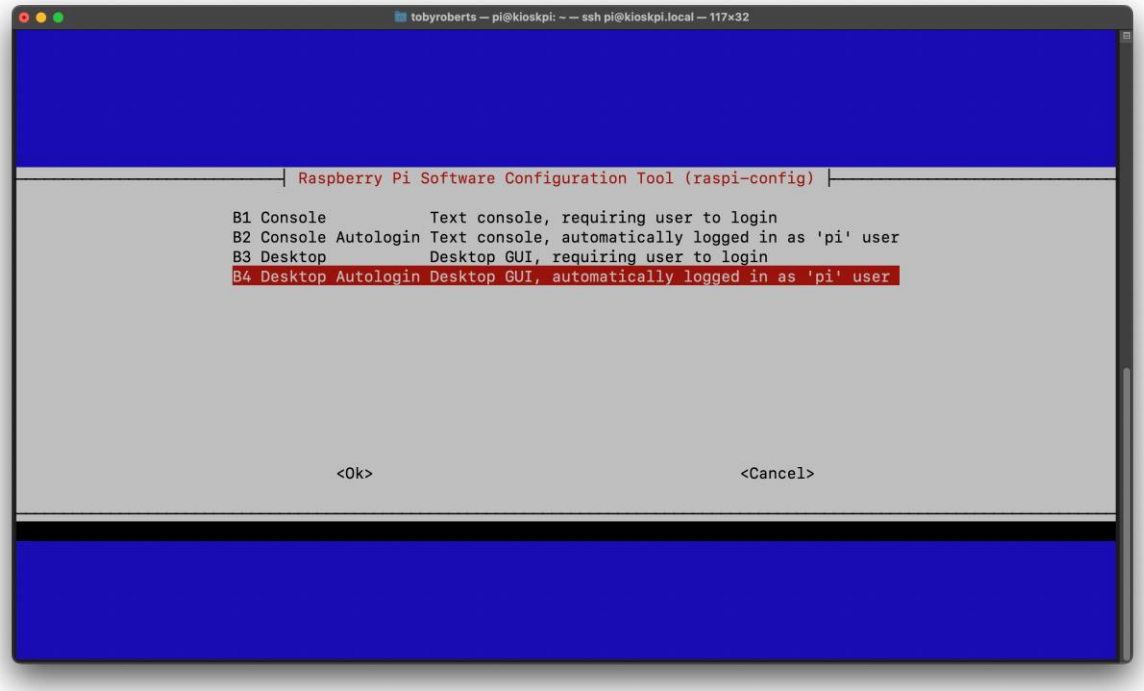
sudo apt install wtype

To ensure your Raspberry Pi boots straight into the desktop environment when it's powered up, use raspi-config:

$ sudo raspi-config



Select System Options > Boot / Auto Login > Desktop Autologin: Desktop GUI, automatically logged in as 'pi' user.



Press **Enter** and **Finish** to save this setting. When prompted, reboot again. Once you have rebooted your Raspberry Pi, you should see the desktop environment on your monitor.

Next, we'll instruct your Raspberry Pi what to present in kiosk mode and how to present it. In this tutorial, we'll display the Raspberry Pi home page and a page displaying the time in London, switching between these two pages every few seconds.

To achieve this, we will edit .config/wayfire.ini. .config/wayfire.ini is a configuration file used to modify the behaviour and appearance of Wayfire, which is used to render the desktop in Raspberry Pi OS.

To open the file for editing in nano, a text editor, run the following command:

$ sudo nano .config/wayfire.ini

Take a look at the section titled [autostart]. At the moment, it reads like this:

[autostart]
panel = wfrespawn wf-panel-pi
background = wfrespawn pcmanfm --desktop --profile LXDE-pi
xdg-autostart = lxsession-xdg-autostart

Let's append a few lines to this section to automatically start kiosk mode whenever your Raspberry Pi powers on:

chromium = chromium-browser https://raspberrypi.com https://time.is/London --kiosk --noerrdialogs --disable-infobars --no-first-run --ozone-platform=wayland --enable-features=OverlayScrollbar --start-maximized
switchtab = bash ~/switchtab.sh
screensaver = false
dpms = false

Add to the file directly below the existing contents of the [autostart] section. When you finish, you should end up with the following:

[autostart]
panel = wfrespawn wf-panel-pi
background = wfrespawn pcmanfm --desktop --profile LXDE-pi

xdg-autostart = lxsession-xdg-autostart
chromium = chromium-browser https://raspberrypi.com https://time.is/London --kiosk --noerrdialogs --disable-infobars --no-first-run --ozone-platform=wayland --enable-features=OverlayScrollbar --start-maximized
switchtab = bash ~/switchtab.sh
screensaver = false
dpms = false

This line opens the Chromium browser in kiosk mode, with two tabsopen: raspberrypi.com and time.is. The extra options alter kiosk mode in the following ways:

--noerrdialogs - suppresses error messages

--disable-infobars - disables notification infobars

--no-first-run - skips the first-run setup experience that typically appears when launching for the first time

--ozone-platform=wayland - uses a Wayland-compatible Ozone platform

--enable-features=OverlayScrollbar - scrollbars appear only when necessary and overlay content instead of using a dedicated scroll gutter

--start-maximized - starts the browser in maximized fullscreen mode

Finally, the line switchtab = bash /home/kiosk/switchtab.sh executes a bash script which will automatically switch between the two tabs every ten seconds.

Press **Ctrl+X**, then **Y**, and finally **Enter** to save the edited file with nano. Next, we'll write that bash script that switches viewing between the two tabs. Usually, the keyboard shortcut **Ctrl+Tab** cycles through the open browser tabs. Our script will use the program we installed, wtype, to simulate and automate keystrokes. To create the script and open it in nano, type:

$ nano ~/switchtab.sh

Add the following to the file:

```
#!/bin/bash



# Find Chromium browser process ID

chromium_pid=$(pgrep chromium | head -1)

# Check if Chromium is running

while

[

[ -z $chromium_pid ]]; do

  echo "Chromium browser is not running yet."
```

```
  sleep 5

  chromium_pid=$(pgrep chromium | head -1)

done

echo "Chromium browser process ID: $chromium_pid"

export XDG_RUNTIME_DIR=/run/user/1000

# Loop to send keyboard events

while true; do

  # Send Ctrl+Tab using `wtype` command

  wtype -M ctrl -P Tab

# Send Ctrl+Tab using `wtype` command

  wtype -m ctrl -p Tab

  sleep 10

done
```

This script first checks that the Chromium browser is running. If not, it waits five seconds before trying again (this gives Chromium enough time to launch before moving on). To toggle between the two tabs, the script uses wtype to simulate Ctrl+Tab every ten seconds**.**

Press **Ctrl+X**, then **Y**, and finally **Enter** to save the new file with nano. Finally, reboot your Raspberry Pi:

$ sudo reboot

Your display should now be show Chrome in kiosk mode, toggling between raspberrypi.com and time.is every ten seconds.

CONCLUSION:-

Using a Raspberry Pi for screen sharing from a laptop to a TV is a versatile and cost-effective approach that leverages the compact size and flexibility of the Raspberry Pi. While challenges such as latency and resolution management exist, ongoing advancements in technology and software development continue to enhance the capabilities and user experience of such setups. This area remains ripe for innovation, offering opportunities for both hobbyist projects and more professional applications.

REFERENCES:-

1.IoT Based Smart Mirror Using Raspberry Pi 4 Ashutosh Narayan Bilange1, Aniket Kadam and H. N. Burande, International Journal Of Advance Scientific Research, Vol 5 ,Iss.4,2020.

2. Design of Smart Mirror Based On Raspberry Pi . B. A. Rani, R. Vinay, C. Darshan, H. S. Shashank andH. N. Bhavana Jain, International Journal of Research in Engineering, Science and Management Volume-3, Issue-5,2020.

3. Smart Mirror using Raspberry Pi Sruthi Gollapalli, Konatham JayaSree, Banavathu Kalyani, V.V.N.V.Phani Kumar, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Vol.9 Iss.6, 2020 .

4. Smart Interactive Mirror Display Varsha Singh, Devi Singh, International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th - 16th Feb 2019 [6].

5. P. P. Patil, “Design and Implementation of Internet of Things Based Smart Mirror Using Raspberry Pi,” vol. 5, no. 2, 2019.

6. Smart Interactive Mirror Display Varsha Singh, Devi Singh, International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th - 16th Feb 2019.

7. P. P. Patil, “Design and Implementation of Internet of Things Based Smart Mirror Using Raspberry Pi,” vol. 5, no. 2, 2019.

8. D’Souza, A. A., Kaul, P., Paul, E., & Dhuri, M. (2019). Ambient Intelligence Using Smart MirrorPersonalized Smart Mirror for Home Use. 2019 IEEE Bombay Section Signature Conference, IBSSC 2019, 2019Januaryr: 5–9. https://doi.org/10.1109/IBSSC47189.2019.8972978 .

9. Garcia, I. C. A., Salmon, E. R. L., Riega, R. V., & Padilla, A. B. (2018). Implementation and customization of a smart mirror through a facial recognition authentication and a personalized news recommendation algorithm. Proceedings - 13th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2017, 2018.

10. M. M. Yusri et al., \"Smart mirror for smart life,\" 2017 6th ICT International Student Project Conference (ICT-ISPC), Skudai, 2017, pp. 1-5.