

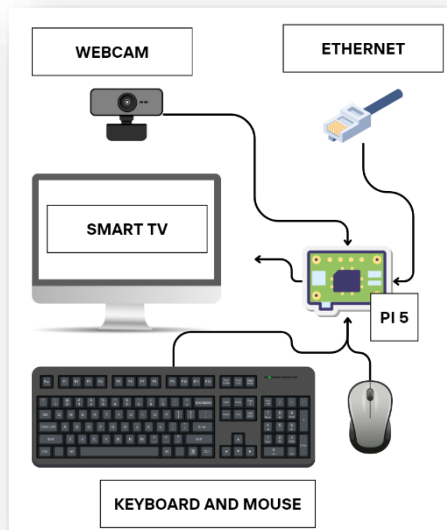
IAU GA 2024 E-Poster Report

By Dominic Vertue, OAD Visiting Fellow

1. Introduction

The E-Poster project was a significant component of the International Astronomical Union (IAU) General Assembly (GA), designed to facilitate a hybrid presentation format. This approach enabled participants to present and engage with posters both online and in-person, ensuring broader accessibility and interaction.

2. Setup



Hardware Configuration:

- **100 x Hisense 40A4K HISENSE 40" FHD SMART TV:** Each screen provided a clear and sizable display for posters, ensuring they were easily visible to both in-person and virtual

participants. This screen was chosen for its value in terms of price, screen size, and resolution.

- **100 x Raspberry Pi 5 (4GB) with Power Supply, Case, Fan, Keyboard, Mouse, and Webcam:** The Raspberry Pi units formed the backbone of the setup, with the latest Pi 5 chosen for its improved performance at the same price as the older Pi 4b. The Pi 5 features a quad-core ARM Cortex-A76 processor running at 2.4 GHz, 4GB or 8GB of LPDDR4X RAM, and dual 4K display output, making it well-suited for handling both e-poster hosting and Zoom breakout rooms simultaneously. Its increased processing power and memory ensure smooth multitasking and seamless management of high-resolution posters alongside video conferencing. Additionally, its Gigabit Ethernet and Wi-Fi 6 support provide stable and fast network connections, crucial for maintaining reliable Zoom sessions.
- **100 x Ethernet Connections:** To maintain a stable and reliable internet connection, each Raspberry Pi was hardwired using Ethernet, preventing connectivity issues that could disrupt sessions.



Software Setup:

- **Raspberry Pi OS:** The operating system selected for each Raspberry Pi was Pi OS, chosen for its compatibility and user-friendly interface. Pi OS comes pre-installed with essential drivers, software, and a familiar user experience, making it an ideal choice. Other systems tested, such as Lineage OS and Linux, did not offer any additional benefits in terms of ease of use. Given these factors, and considering the goal of donating the units, we opted to standardize on the simplest and most accessible option.
- **Chromium Web Browser:** The system was used to run the web-based Zoom application, enabling connections to 100 individual breakout rooms, each linked to a different screen and poster. On Pi OS, Chromium provided the most stable Zoom experience, outperforming Firefox. While Firefox allowed participation, it lacked the ability to display screen shares and pop-out windows simultaneously, which limited its functionality compared to Chromium.
- **Web-based Zoom:** Each screen was set up to host a separate Zoom room, facilitating seamless interaction between presenters and participants. Since Zoom lacks native support for the ARM64 architecture used by the Raspberry Pi, the web-based version offered the most stability. We also tested the Zoom version available through PiApps,

but it did not support screen sharing on the Raspberry Pi units we tested, further solidifying our choice of the web-based approach for reliable performance.

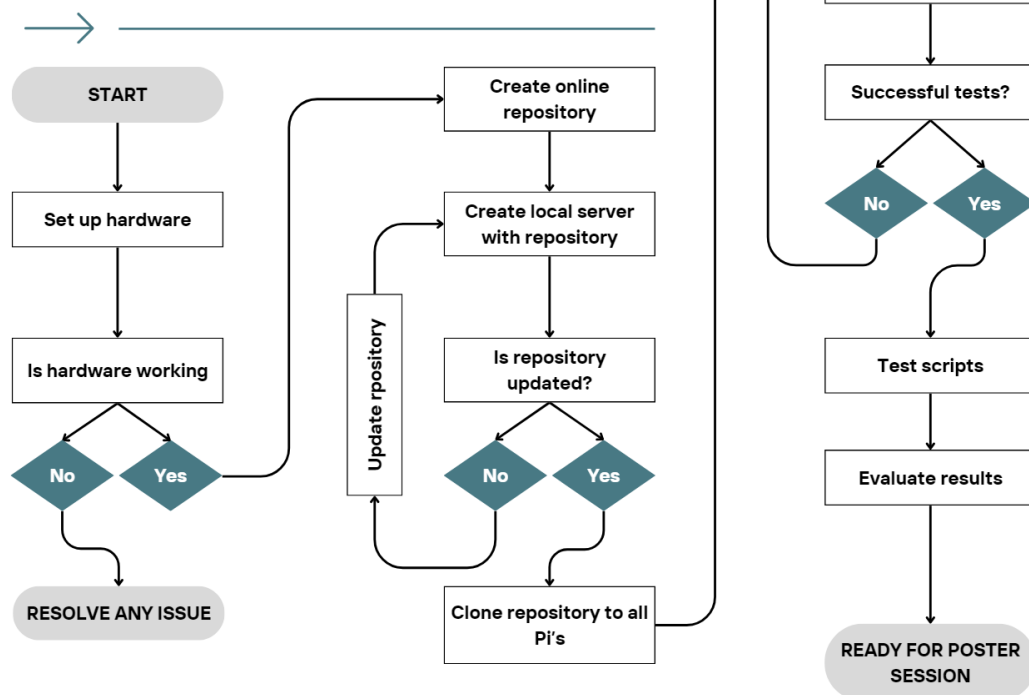
- **Python:** Scripts written in Python were used to automate various tasks, such as opening the correct poster files and managing the Zoom interface.
- **Apache Web Server:** Hosted a local server to manage and distribute files to all Raspberry Pi units.
- **RealVNC:** Provided remote access to the Raspberry Pi units, allowing for monitoring and troubleshooting throughout the event.

Installation Process:

1. **Hardware Setup:** All 100 screens and Raspberry Pi units were assembled, connected, and configured with the necessary peripherals (ethernet connection, keyboard, mouse, and webcam).
2. **Local Server Creation:** A server was set up locally to manage and distribute content to the Raspberry Pi units.
3. **Repository Deployment:** The necessary files, including posters, were copied from GitLab to the local server, then deployed to all 100 Raspberry Pi units.
4. **Zoom Credentials Setup:** Zoom credentials were configured on each unit to ensure they could connect to the correct breakout rooms.
5. **Script Testing:** Scripts were tested to ensure they functioned correctly, automating as much of the process as possible.

Process diagram:

INSTALLATION



3. Running the Sessions

Throughout the IAU GA, the scripts used to manage the E-Poster presentations were continuously optimized and refined. Initially, the process involved manual steps, such as logging into Zoom, loading the poster from the repository, and sharing the screen. The Zoom window was configured to overlay the poster, allowing participants to engage with the presenter effectively.

By the end of the event, the process was streamlined as follows:

- **Zoom Connection:** Ensured that Zoom was running and connected to the correct breakout room.
- **Poster Display:** Ran a script that checked the poster schedule, opened the corresponding PDF on each Raspberry Pi, and displayed it on the screen.
- **Screen Sharing:** The script then initiated screen sharing, with the Zoom window overlaid on the poster, facilitating real-time interaction.

4. Challenges

Technical Issues:

- **Network Interruptions:** Occasional network outages caused Zoom connections to drop, requiring manual reconnection of each Raspberry Pi to its respective breakout room.
- **Automation Limitations:** While efforts were made to develop a fully automated script to manage the entire process, it was not fully completed by the end of the event, leaving some manual interventions necessary.

Logistical Challenges:

- **In-Person Interference:** Managing in-person participants was a challenge, as some inadvertently closed Zoom, minimized posters, or muted audio, disrupting the sessions.

Solutions Implemented:

- **Volunteer Support:** To mitigate issues caused by in-person participants, volunteers were assigned to monitor sessions, assisting as needed.
- **Remote Monitoring:** RealVNC was employed to remotely monitor and reinitialize screens, allowing for quick intervention when issues arose.

5. Lessons Learned

Key Takeaways:

- **Feasibility of Hybrid Poster Sessions:** The project demonstrated that managing up to 100 screens for hybrid poster sessions using Zoom is feasible. Zoom's ability to host 100 breakout rooms enabled smooth interaction between presenters and participants across multiple screens.

- **Importance of a Centralized Repository:** A well-organized repository for storing and managing the posters was essential. It streamlined the loading and display of content on each screen, reducing potential errors and delays.
- **Critical Role of Team and Volunteers:** The success of the project heavily depended on the dedication of the team and volunteers. They were crucial in assisting in-person participants, addressing queries, and managing online engagement, highlighting the indispensable human element in ensuring smooth operation.

Future Recommendations:

- **Consideration of Thin Clients and Servers:** If budget allows, using a centralized server with thin clients instead of individual Raspberry Pi units could simplify setup and management. This would reduce the complexity of handling multiple devices and make updates and troubleshooting easier.
- **Team Training Prior to Implementation:** Conducting comprehensive training sessions for the team before the event would help eliminate many initial challenges. Familiarizing the team with the setup, software, and potential issues in advance would lead to more efficient and trouble-free execution.
- **Use of TV Stands:** Positioning TV screens on stands would enhance the viewing experience by elevating the screens to eye level. This would improve visibility and accessibility, particularly in a busy conference environment, allowing participants to engage more effectively with the posters.

6. Code Used

The code played a crucial role in managing the E-Poster sessions. A local repository hosted the posters, each renamed according to its submission ID. The following process was used:

- **PDF Display Script (IAUScript.py):** A Python script opened the relevant PDF file, resized the window to fit the screen, and initiated screen sharing. This setup made it straightforward to share the screen and manage the Zoom overlay, ensuring a smooth presentation for participants. This script interacts with window management utilities (e.g., `wmctrl`) to retrieve information about open windows on a system. It aims to manipulate windows, potentially to control or organize them in a specific way during an event or session.
- **Distribute scripts to all pi's (create_distribute_scripts.py):** This script generates SSH commands for distributing and executing poster-related Python scripts across multiple Raspberry Pi devices. It processes poster and screen assignment data from a CSV file and creates corresponding SSH scripts for specified dates and times, allowing the remote execution of a poster display script.
- **Assign screens to presenter (Assign only if submitted.py):** This script helps assign screens (up to 100 per session) based on submissions and if user has been tagged as paid or not. It processes input data from Excel files, checks conditions, and assigns screen to qualifying entries.
- **Create a program based on assigned screens (Program.py):** This script takes the results from the Assign if submitted script and creates an excel program showing day, time, presenter, and which screen/zoom room the poster was allocated too.

- **Summarize screen utilization (Summarize_max_screens.py):** This script processes an Excel file to determine the maximum number of screens used in both morning and evening sessions across different dates. It generates a pivot table that organizes the screen numbers and start times for morning and evening sessions, saving the summarized data in a new Excel file.

7. Conclusion

The E-Poster project at the International Astronomical Union (IAU) General Assembly was a groundbreaking initiative that enhanced the hybrid conference experience by integrating 100 screens powered by Raspberry Pi units. This allowed for a dynamic and interactive poster session accessible to all attendees, whether present in person or online. The project demonstrated the feasibility of managing large-scale hybrid sessions using Zoom, emphasized the importance of a centralized repository for content management, and highlighted the critical role of a well-coordinated team and volunteers. Despite challenges such as network interruptions and the need for continuous monitoring, the event was a success, providing valuable insights for future hybrid conferences.

Key recommendations for improvement include investing in thin clients and centralized servers to streamline the setup, as well as providing comprehensive team training to address initial challenges. Additionally, using TV stands would improve the visibility and accessibility of the screens during events.

Beyond its immediate success, the E-Poster project has long-term benefits. The decision to purchase, rather than rent, the screens and Raspberry Pi units enabled them to be donated to local schools after the event, extending the impact of the IAU GA. This initiative supports the IAU's broader goal of promoting education and engagement in astronomy, as these tools will be repurposed for ongoing outreach and educational activities in the community, creating lasting resources for future generations.

*Access to Code Used

To access all the code used during the implementation refer to: <https://github.com/Acidian/IAU-GA-Scripts/tree/main>