This is a preprint of the article peer reviewed and published in Acta Astronautica Volume 138, September 2017, Pages 43-52
at https://doi.org/10.1016/j.actaastro.2017.05.023 .
Please cite the published version.

SPACE EDUCATION AND OUTREACH SYMPOSIUM (E1)
Lift-Off - Secondary Space Education (2)

ASTRO PI: RUNNING YOUR CODE ABOARD THE INTERNATIONAL SPACE STATION

Mr. David Honess[a]*, Mr. Oliver Quinlan[b]

[a] *Raspberry Pi Foundation, 30 Station Road, Cambridge, Cambridgeshire, United Kingdom CB1 2JH, dave@raspberrypi.org*
[b] *Raspberry Pi Foundation, 30 Station Road, Cambridge, Cambridgeshire, United Kingdom CB1 2JH, oliver@raspberrypi.org*
* Corresponding Author

Abstract

A team of leading United Kingdom (UK) space companies, in collaboration with the Raspberry Pi Foundation (RPF), created the Astro Pi project (www.astro-pi.org) in order to exploit the outreach possibilities offered by the flight of British European Space Agency (ESA) Astronaut Tim Peake to the International Space Station (ISS) between December 2015 and June 2016. At the core of the project is the Astro Pi payload, which was developed to be taken to the ISS and into classrooms. It consists of a Raspberry Pi B+ computer, an add-on board called the Sense Hardware Attached on Top (HAT), a camera, a joystick, and six push buttons. The Sense HAT board was designed to host a range of sensors and input/output devices to be used for experiments aboard the ISS and in the classroom. The payload was qualified for spaceflight in 2015, and a competition was run to promote engagement between schools and the UK space industry that was supported by a range of learning resources. The payload was launched into space on the Orbital Sciences OA-4 Cygnus cargo freighter in December 2015, and the winning experiments were run on the ISS by Tim Peake in early 2016. These generated a range of data for analysis on the ground. Between February and March 2016, a second competition was run. In both competitions, the prize was to have students' code uploaded and run on the Astro Pi payload by Tim Peake. This paper is divided into four sections. The first provides details of the Cygnus launch and flight operations aboard the ISS. The second reviews the results of the first competition, and how these were distributed and analysed by the schools. The third covers the second competition including its results, impact, and engagement. The fourth details ongoing outreach, events, and the establishment of a long-term legacy for Astro Pi. This paper continues the work in the 2015 paper ASTRO PI: LAUNCH YOUR CODE INTO SPACE (IAC-15-E1.2.1); readers may consult the original paper for context first.

Keywords: Education, STEM, Computing, Programming, Coding, Peake

Acronyms/Abbreviations

- AC: Alternating Current
- ARISS: Amateur Radio on the International Space Station
- AV: Antivirus
- BIOTESC: Biotechnology Space Support Center
- CAD: Computer-Aided Design
- CC-BY-SA: Creative Commons Attribution Share-Alike
- CCD: Charge-Coupled Device
- CHX: Cabin Heat Exchanger
- CPU: Central Processing Unit
- CSV: Comma-Separated Values
- EO: Earth Observation
- ESA: European Space Agency
- ESERO: European Space Education Resource Office
- EST: Experiment Sequence Test
- FAQ: Frequently Asked Question
- GCC/G++: GNU Compiler Collection
- HAT: Hardware Attached on Top
- HTML: HyperText Markup Language
- HTTP: HyperText Transfer Protocol
- HTTPS: HyperText Transfer Protocol Secure
- IDE: Integrated Development Environment
- ISS: International Space Station
- JPEG: Joint Photographic Experts Group
- JSL: Joint Station LAN (see LAN)
- KUIPS: Ku-Band Internet Protocol Service
- LAN: Local Area Network
- LED: Light-Emitting Diode
- MB: Megabyte

- MCP: Master Control Program
- MP3: Moving Picture Experts Group Audio Layer 3
- MPCC: MultiPurpose Computer and Communications
- NDVI: Normalized Difference Vegetation Index
- NASA: National Aeronautics and Space Administration
- NORAD: North American Aerospace Defense Command
- OS: Operating System
- OMD: Orchestral Manoeuvres in the Dark
- PDF: Portable Document Format
- RPF: Raspberry Pi Foundation
- SCP: Secure Copy
- SD: Secure Digital
- SEU: Single Event Upset
- SSH: Secure Shell
- STEM: Science, Technology, Engineering and Mathematics
- STL: Stereolithography
- TCP/IP: Transmission Control Protocol / Internet Protocol
- TMA: Transport Modified Anthropometric
- UAE: United Arab Emirates
- UK: United Kingdom
- UOP: Utility Outlet Power
- US: United States
- USB: Universal Serial Bus
- UTF: Unicode Transformation Format
- V: Volts
- VPN: Virtual Private Network

1 Introduction

Astro Pi is an educational programme intended to engage children and young people with programming and physical computing, using the context of space exploration. It is based around coding challenges, where young people create programs on Raspberry Pi computers or compatible software. The winning programs are then run on the ISS, on Raspberry Pi hardware which is functionally identical to that used by the entrants, with the exception of some additional space reinforcement.

The first round of the programme took place in 2015 and involved challenges for primary school and secondary school children.

The primary challenge was aimed at students aged 11 and under. Two winners were chosen, who worked with the Raspberry Pi Foundation to code, test, and prepare their software for space flight.

The secondary challenge was aimed at students aged 11 to 18 and had two phases. Phase one was identical to the primary competition and ran in parallel, with entries judged in the same way. The winners were not chosen at this stage. Instead, teams with promising ideas were rewarded with hardware kits to be used during the second phase. The kit included a Raspberry Pi computer and a Sense HAT add-on board, which incorporated environmental sensors and an LED display. In phase two they were required to program their experiments on this equipment and submit the completed software. These programs were then judged by representatives from the Raspberry Pi Foundation and experts in the space industry from our partner organisations, and five winners were selected.

All seven winning experiments were tested on the Astro Pi flight hardware and launched to the ISS, where they were run on the Astro Pi payload over the course of several weeks, under the supervision of Tim Peake. The data collected was downloaded to ground, made available for the students to analyse, and subsequently released to the public on the Astro Pi website.

1.1 Winning Experiments

Crew detector
Age 11 and under. Takes photographs of the crew, using the humidity sensor to detect their presence.

SpaceCRAFT
Age 11 and under. Logs a variety of sensor measurements of the ISS interior to a CSV file, with the aim of using the computer game Minecraft on Earth as a data visualisation tool.

Flags
Age 11-14. Tracks the location of the ISS and works out what country it is above, then shows its flag on the LED matrix along with a short phrase in the local language.

Watchdog
Age 11-14. Continually measures the temperature, pressure, and humidity. Displays the data in a cycling, split-screen display. Raises visual alarms if these deviate from acceptable ranges.

Trees
Age 14-16. Takes near-infrared pictures of the ground to post-process back on Earth, to measure plant health across wide areas of land.

Reaction Games
Age 14-16. A suite of puzzles, memory games, and logic-based tests that record response times, with the aim of investigating how crew reaction time changes over the course of a long-term space flight.

Radiation
Age 16-18. Radiation detector using the Raspberry Pi Camera Module. Uses image recognition software to count specks of light caused by radiation, and produces a measurement of the radioactivity occurring.

Data Logger (long-term environmental monitoring)
Not a student experiment. Written by the Raspberry Pi Foundation, the program continually takes sensor measurements, recording the averages for each sensor every ten seconds into a CSV file for download to ground.

2 Launch and flight operations

2.1 Preflight preparations

The Astro Pi physical payload consists of two Raspberry Pi B+ computers with sensor peripherals, enclosed in aerospace grade aluminium cases weighing over half a kilo each. In order to gain permission to put a payload onto a launcher, a flight safety certificate must be obtained. ESA have a three-stage review process where the payload developer presents their documentation and receives guidance on the next stage from a panel of safety experts.

The code was examined by Raspberry Pi staff and modified to fix bugs, improve performance, and make it work with the control software. Care was taken to preserve as much of the original student code as possible. Lengthy burn-in tests were conducted to check for long-term issues like memory leaks and lack of disk space. The software was not submitted to the ESA software control board, as it was intended to run offline at all times.

Payload developers are also required to write an operations manual detailing all on-orbit and ground-based procedures required for the mission. This document is then used by ESA personnel to develop and test the procedures that the crew will follow on-orbit. An EST was performed in August 2015, resulting in several changes to control software and a full repeat of the EST tests. In October 2015 Raspberry Pi staff consulted ESA astronaut Tim Peake, resulting in further refinement.

The qualification process was done with the safety requirements of the Soyuz TMA launch vehicle in mind, with the intention that the payload would fly with Tim Peake on 15 December. However, due to storage restrictions, the Astro Pi payload was moved to Orbital Sciences Cygnus OA-4, departing earlier than the Soyuz. The Cygnus was launched on United Launch Alliance's Atlas V rocket from pad 41 at Cape Canaveral on 6 December 2015, arriving at the ISS on 9 December.

2.2 Deployment scenarios on-orbit

The payload was designed to be powered in two ways on board the ISS. This was accomplished using a UOP AC inverter, which supplies 125V US mains, and a domestic wall-socket phone charger. It could also be powered from the USB port of a crew support laptop, but this option was not used.

The Astro Pi cases have a NASA-standard attachment for mounting onto the ISS multi-use brackets. The corner bolt enclosures also have holes for tethering to provide further deployment flexibility.

The two Astro Pi computers were deployed in different modules of the ISS: one in the European Columbus module and one in the US Node 2 module, mounted on the Nadir hatch window for Earth observation. The Node 2 unit was originally planned to be deployed in the Cupola and powered from one of the laptops there; however, this contravened the NASA Cupola usage agreement, therefore the Nadir hatch window was selected as an alternative. Tim Peake deployed the first Astro Pi in the Columbus module on 2 February 2016, and the second in Node 2 on 16 February.

2.3 Execution of student experiments

The 2015 Astro Pi competition produced seven winning Python programs. Six of these were allocated one full week of run-time on-orbit to gather data; one program was a suite of memory and reaction time games, which only required short run-times on multiple occasions. Additionally, one experiment for scientific data logging, written by the Raspberry Pi Foundation, was allocated three weeks of run-time.

The payload was supplied with bootable SD cards containing the experiments, along with a full Raspbian OS which boots the Astro Pi and begins processing without crew interaction. The objective was to allow these programs to reach their allocated run-time and return any output files generated to ground for the students to analyse.

LAN qualification for the payload was not attempted in order to meet the deadlines of Tim Peake's launch schedule. It was decided the objectives would be accomplished by three offline Astro Pi deployments lasting three weeks. Each deployment would execute two week-long student experiments consecutively, followed by the scientific data-logging program for another full week (or until convenient for the crew to stop it).

Table 1 below illustrates this deployment plan. Any output files would be manually transferred to ground after each deployment.

Location	Duration	Astro Pi unit	Experiments
Columbus	3 weeks	Visible spectrum camera	Crew Detector Space-CRAFT Data Logger
Columbus	3 weeks	Visible spectrum camera	Watchdog Radiation Data Logger
Node 2 Nadir hatch window	3 weeks	Infra-red spectrum camera	Flags Trees Data Logger

Table 1: Astro Pi deployment plan

The reaction time games were planned into the deployment process. After completing the setup for each deployment, Tim Peake would play one or two reaction games, thus creating three separate snapshots of his reaction times for the students to analyse on ground.

The MCP custom Python script, written by the Raspberry Pi Foundation, was used for task scheduling. The MCP provides a navigable menu on the LED matrix, allowing the crew to manually select either an experiment or a task list which manages the automatic processing of experiments. The task list continually records the total run-time for each experiment, and proceeds to the next when the target total has been reached. This was designed to be resistant to multiple failure modes, including SEUs that reboot the computer. After such an event, the MCP resumes processing where it left off.

The MCP also monitors disk space and ceases all processing if space runs low on the bootable SD cards. It then displays a warning message to the crew, who respond by performing the procedures to complete the deployment and transfer the data to ground. Only the Trees experiment risked filling disk space with continuous near-infrared pictures of the Earth.

2.4 Transfer of data to ground

Without LAN certification, a manual procedure for transferring files was necessary. The process requires approximately 20 to 30 minutes of crew time, depending on the nature of the files being transferred, and was unavoidable with no network access.

The downlink procedure was to manually shut down the Astro Pi, remove its bootable SD card, insert the card into a Windows laptop and copy the complete Transfer folder onto the laptop. The output file format varied depending on the students' choices, but the most common types were UTF-8 encoded text files (trace logs and CSV data) or JPEGs taken from the Camera Module.

Text data could be downlinked to ground and released. Image data required NASA and ESA approval before release. There was a crew privacy concern for the Crew Detector experiment, which took pictures in the Columbus module where medical experiments are carried out.

When the data was received, it was checked, authorised, and then emailed as a zip file to Raspberry Pi staff for evaluation, with permission to release. Staff were then required to confirm that the data was intact and as expected before it was removed from the ESA/NASA servers, then the files were forwarded to the relevant students to analyse.

2.5 Crew time used

2.5.1 Prime

Prime activities are unavoidable, and include cleaning, maintenance, accommodating visiting vehicles, and dealing with emergencies. Educational activities are rarely classified this way, therefore only unpacking the cargo transfer bag containing the payload was classified as prime.

2.5.2 Reserve

A reserve activity should always occur, provided there are no higher-priority activities or emergencies that must be dealt with first. There are also priority levels within the reserve classification; educational activities are the lowest. All crew activities required for this mission were classified as reserve. These included deployment and stowage for the two Astro Pi computers, three manual file copy activities, and playing the reaction time games. It was never precisely known when these activities would occur, but the Raspberry Pi Foundation received a daily operations report whenever an Astro Pi-related activity occurred. These reports detail the activities completed, whether any problems were encountered, and what remedial actions were taken.

2.5.3 Private

Outside their five-day week the crew may take on more tasks at their own discretion. Private activities are not reported, unless it is a reserve activity completed during private time. This is one area where Astro Pi benefitted from the support of a highly motivated and engaged astronaut. From direct email communication with Tim Peake, the Raspberry Pi Foundation learned that he used private time to interact with some experiments, and to fix an issue in one program.

2.6 Problems encountered in flight

2.6.1 Experiments that ran suboptimally

The Flags experiment relied upon stored NORAD two-line element telemetry, which was outdated when the program ran on-orbit. Subsequently, the location calculated did not match the location of the ISS and incorrect flags were displayed. Tim Peake manually updated the telemetry in his spare time, and afterwards confirmed it was showing the correct flag when he looked out of an Earth-facing window.

The Watchdog experiment tackles a design issue with the Sense HAT, where thermal transfer from the Raspberry Pi CPU causes its temperature sensors to always read a few degrees above ambient. This was addressed by using the nominal difference of CPU temperature above background ambient, as a means to calibrate the Sense HAT sensors. The calibration code was developed on a Raspberry Pi 2, whereas the Astro Pi flight hardware contains a Raspberry Pi B+, which produces less heat from its CPU, so the calculated ambient temperature was always out by several degrees. It is possible to post-process the data to correct this.

The Trees experiment required a blue filter to exclude all red light from the camera, so that only infrared light is recorded in the red channel of the image files. However, no blue lens filter was available, meaning that the data acquired could not be post-processed for NDVI analysis as intended by the students. A visible light index was used instead, impacting on analysis quality.

The Radiation experiment measured ionising radiation using the CCD sensor of the Raspberry Pi Camera Module. However, the radiation could not penetrate the payload's aluminium case, causing the experiment to detect only a very low level of ionising events. The experiment was verified on ground using a Camera Module and a neutron beam at the Rutherford Appleton Laboratory in the UK. Future experiments will explore alternatives to the Camera Module.

2.6.2 Experiments not completed

The Reaction Games experiment was only run on one occasion due to high crew workloads. Having only one sample of Tim Peake's reaction time was not enough data to draw conclusions about how his reaction time had changed throughout the mission.

3 Scientific Results

This section examines the results of the 2015 competition experiments obtained by Tim Peake during ISS expeditions 46 and 47. These results are shared online for anyone to download and analyse.

3.1 Review of raw experiment results

3.1.1 Crew Detector

Files: https://github.com/astro-pi/SweatyAstronautCode/tree/master/iss downloads

Eleven pictures were taken over the course of seven days. Nine of these show Tim Peake, one shows Russian Cosmonaut Mikhail Kornienko, and some were empty. The results consist of the images and three custom log files.

3.1.2 SpaceCRAFT

Files: https://github.com/astro-pi/SpaceCRAFT/tree/master/iss downloads

Over 60,000 rows of sensor measurements were collected in a 26 MB CSV file.

3.1.3 Flags

This experiment was display-only and did not produce any results.

3.1.4 Watchdog

Files: https://github.com/astro-pi/watchdog/tree/master/iss downloads

Over 17,000 rows of measurements were recorded in two CSV files. There was a minor issue with the ambient temperature calibration code, as stated in 2.6.1.

3.1.5 Trees

Files: https://github.com/astro-pi/enviro-pi/tree/master/iss downloads

A total of 4301 Earth observation images were recorded, each perfectly in focus. However, a blue lens filter was not available, as stated in 2.6.1, meaning that an alternative method of analysis was used (see 3.3.5).

3.1.6 Reaction Games

Files: https://github.com/astro-pi/reaction-games/tree/master/iss downloads

Due to high workload, Tim only played two of the games on one occasion. Each game produced a short log file recording Tim's performance times.

3.1.7 Radiation

Files: https://github.com/astro-pi/radiation/tree/master/iss downloads

140 discrete, seemingly random ionising radiation events were detected by the sensor during the week-long experiment. The thickness of the flight case kept out much of the radiation.

3.1.8 Data Logger (long-term environmental monitoring)

Files: https://github.com/raspberrypilearning/astro-pi-flight-data-analysis/tree/master/data

The data was gathered on three occasions in two different ISS locations, and produced around 80 MB of CSV data which was downloaded to ground and made available online, with an accompanying educational resource [1]. The intention is for the CSV files to be analysed by students to look for evidence of various types of activity, such as crew presence, CHX dry-outs, oxygen repressurisation events, ISS reboosts and flying through the South Atlantic Anomaly.

3.2 Method of distribution to schools

The files returned from the experiments were initially only shared with the students that wrote the programs, to allow them to be the first to publish the results. The intention was to cultivate their desire to see the project through to its conclusion, as well as their understanding of the scientific process in general.

ESA personnel emailed the files to the Raspberry Pi Foundation; these were forwarded directly to the individual students and their teachers.

3.3 Student interpretations

3.3.1 Crew Detector

Student analysis proved that the theory of the humidity sensor being able to detect the crew's presence was correct. Log files show that each spike in humidity successfully resulted in a picture being taken. One picture was also accompanied by a button press to confirm crew presence.

The students wrote up their results and published them online [2]. In early March of 2016 they delivered a 30-minute presentation at the 4th Raspberry Pi Birthday event, where they explained their results to an audience of students, teachers, and parents.

3.3.2 SpaceCRAFT

The data was published on social media, and an educational resource [3] was developed to show how to visualise the data in Minecraft. As the raw data is similar to that captured by the Data Logger experiment, this data could also be analysed in the same way using this educational resource.

3.3.3 Flags

This experiment did not produce any files that could be analysed as its purpose was to display information for the crew. However, Tim Peake confirmed that the correct flags were being displayed.

3.3.4 Watchdog

The logs show that the temperature alarm state was continuously showing, due to the calibration problems stated in 2.6.1. An analysis has not yet been published.

3.3.5 Trees

Due to the issues stated in 2.6.1, the students sought an alternative chlorophyll-based plant indexing system.

This involved independent research, consulting another Principia education programme, EO Detective [4]. A visible light-based index was suggested which, while inferior to NDVI, will allow some level of analysis. The students are yet to publish their findings.

3.3.6 Reaction Games

Having only one sample of Tim Peake's reaction time was not enough data to draw any conclusions about how his reaction time had changed throughout the mission.

3.3.7 Radiation

After initial conclusions that the experiment was not successful due to a lack of ionising radiation penetrating the camera sensor, the log file was examined and an unusual event was discovered. The software uses a calibration routine to compensate for radiation damage to the CCD sensor and Bayer filter. For the first three days, the calibration code subtracted pixels but this inexplicably ceased. The students theorised that the cause may be pre-existing damage to the camera, light leaking in from the lens cover, or subsequent radiation damage. They have yet to publish a full analysis.

3.3.8 Data Logger (long-term environmental monitoring)

The Flight Data Analysis data has shown several oxygen repressurisation events, one CHX dry-out, and two ISS reboosts; these were verified by Raspberry Pi staff.

3.4 Publication of data sets online

After the four-month grace period for the students to publish their results, the data from each experiment, along with available student interpretations, was made publicly available online. It is hoped that the availability of this data may encourage further analysis and collaboration between the students and the public. The data's availability was announced on the Raspberry Pi blog [5] and explained on the Astro Pi website [6]. The data is hosted on GitHub [7].

3.5 Outcomes for future missions

Several conclusions may be drawn from the results of these experiments:

- Young people can create functioning computer programs that run on the ISS and generate useful results.
- Some technical constraints exist, which can be accounted for in future competitions.
- The project produced high levels of engagement in computer programming among students.
- We have a reliable computer for spaceflight. The MCP log files showed no unexpected reboots from radiation during ISS increments 46 and 47, while the payload was online. This indicates that the payload is resistant to SEUs.
- Analysis and publication of the results should be supported in future competitions, for both existing participants and the wider public.
- Experiments that depend on crew time must be carefully considered, with appropriate caveats communicated effectively.

4 The second coding challenge

4.1 Rationale

A second coding challenge was devised to capitalise on the interest generated by the launch and the implementation of the first challenge. Two concurrent competitions were run, one aimed at students with a moderate level of programming experience, and one for those with little or no experience. Two challenges were designed for the competition: creating an MP3 player for the Astro Pi units, and programming music using Sonic Pi software for Tim Peake to listen to using this player.

4.2 Sonic Pi music competition

Sonic Pi is a domain-specific language based on Ruby, designed to allow the creation of music through programming in a real-time, live environment. It was created by Sam Aaron at the University of Cambridge Computer Laboratory, supported by Broadcom and the Raspberry Pi Foundation. The Raspberry Pi Foundation provide a range of related learning materials, including a scheme of work for teachers [8], a MagPi Essentials book [9], and a regular feature in The MagPi magazine [10]. Sonic Pi is a powerful tool which can be used for creating complex programs, but it presents a shallow learning curve for those less familiar with programming.

The music competition was designed to be a point of entry to Astro Pi requiring less previous experience of programming than the first coding challenge, while still giving participants scope to think creatively and learn new programming skills. The brief was to create an original piece of music using Sonic Pi for Tim Peake to listen to on the ISS. Pieces had to be between 120 and 360 seconds long, a single file, and no more than 200 lines long. These constraints were necessary to control both the time it would take to judge and to listen to the pieces. Sonic Pi has the capacity to import additional audio files to manipulate as part of a piece, and to interface with other software synthesisers. For this contest, participants were not allowed to use this, and had to use only the built-in functions of Sonic Pi 2.8.0. Entrants had to agree to licence their entry as Creative Commons CC BY-SA so it could be shared on the Astro Pi website.

The Sonic Pi music competition ran between 2 February and 21 March 2016. It was open to children in the UK between the ages of 11 and 18. The judging criteria included code quality and the creativity and originality of the music produced.

4.3 MP3 player competition

The MP3 player competition was devised for children and young people with more experience of programming, and was therefore both more challenging and more directly focused on software engineering. Entrants were asked to program an MP3 player for the Astro Pi units that Tim Peake could use to listen to new music. It was required that this program be written in the Python language (either version 2.7 or 3.2) which is supported on the Raspberry Pi, including the Astro Pi units. Specific constraints included the use of the Sense HAT's LED display as an interface, and the use of keyboard event detection. Criteria for judging included the functionality of the software, appropriate use of hardware features, and code readability and quality.

4.4 Teaching Resources

For the Sonic Pi music contest, participants were directed to the 'Getting started with Sonic Pi' resource on the Raspberry Pi website [11], which provides step-by-step tutorials on writing music through code. More advanced participants were directed to Sam Aaron's book on Sonic Pi [12]. For teachers, the existing Sonic Pi scheme of work on the Raspberry Pi Foundation website was also available.

For the MP3 player contest, it was necessary to ensure that entrants used software libraries available to the offline Astro Pi units on the ISS. To support participants in working around this limitation, we provided two examples of MP3 players using the built-in libraries Pygame and Omxplayer [13]. These suggested a way to implement the core function of the MP3 player, while leaving details of design and further functionality to the participants.

The original Astro Pi contest resources for the Sense HAT were also available [14]. These provided support for accessing the sensors in the Sense HAT, using the display to give feedback to the astronauts, and making use of the buttons for user input to control a program.

4.5 Judging process

Entries were shortlisted by software experts, aerospace professionals, and Raspberry Pi Foundation staff, and passed to judges for the final decision. The judges selected one winner for each of the school Key Stages 2 (ages 8-11), 3 (ages 12-14), 4 (ages 14-16), and 5 (ages 16-18). The judges for the competition were as follows:

Sonic Pi Music:

- Andy McCluskey, OMD
- Paul Humphreys, OMD
- Ilan Eshkeri, British/French film composer
- Sam Aaron, author of Sonic Pi
- Nimal Navarathinam, Systems Engineer at Surrey Satellite Technology, UK
- David Honess, Astro Pi Programme Manager at the Raspberry Pi Foundation

MP3 player:

- Libby Jackson, Astronaut Flight Education Programme Manager at the UK Space Agency
- Ben Dornan, Causeway Associate in Computer Science at the Royal Institution, UK
- Slawomir Zdybski, Education Team at the European Space Agency, Holland
- Pat Norris, Partner and Space Specialist at CGI, UK
- Fabienne Wyss, ISS Operations Engineer at BIOTESC, Switzerland
- David Honess, Astro Pi Programme Manager at the Raspberry Pi Foundation, UK

4.6 Entries received

The Sonic Pi Music contest received 27 entries. The MP3 player contest received 13 entries; more entries were from the older age groups. We believe a number of factors contributed to the lower number of entries submitted:

- Time constraints: the short nature of the competitions was necessary to comply with Tim Peake's schedule, but would have limited entry numbers.
- Submission deadline: the deadline of the competition occurred in the UK school holidays.
- Degree of challenge: while the MP3 competition was intended to be more taxing, it became clear that the length of submission required for the Sonic Pi competition (120 seconds minimum) represented a higher degree of challenge than anticipated.

4.7 Impact and engagement

762 children were involved in submitting completed entries, and the website associated with the contests has had 26,968 unique views. We have delivered outreach activities at events with a total audience of 216,000, with significant numbers of young people having their first experience of programming at those events. Media attention associated with the contests has contributed to our broader objectives to raise the profile of STEM education.

These coding challenges also provided a useful pilot for devising a contest with astronaut involvement while that astronaut's mission was underway. The initial coding challenge was planned while the Principia mission was at the preparation stage, but this challenge demonstrated that education projects can be devised during missions as well as before. This is an important precedent for future aerospace education in general, and specifically allows for future development of the Astro Pi project to continue to derive educational benefits from the units that remain in space.

5 Ongoing Outreach and Legacy

5.1 Outreach events

The Raspberry Pi Foundation undertakes a large number of outreach events. In the course of the programme, Astro Pi has been made an important feature of these events, with increasing interest from the public as the Principia mission took place.

The Sense HAT hardware provides an accessible entry point to physical computing, where event attendees can use either visual or text-based programming languages to interact with physical hardware, monitor the environmental sensors, and provide feedback on the LED display. Space exploration is a compelling context to understand how even the simple programming learned at these introductory workshops can be used to solve problems.

5.2 Resources

The Astro Pi educational resources remain available on our website, and the recently developed Sense HAT emulator ensures they are accessible with or without the physical hardware. Resources include tutorials on analysing the data collected on the ISS and collecting environmental data on Earth, interacting with Minecraft using the LED matrix, and creating simple games using the Sense HAT display and inputs [15].

5.3 Build your own Astro Pi

A side effect of the Astro Pi mission was that the Raspberry Pi Foundation was inundated with requests for replica flight cases or CAD design files to allow third-party manufacture.

It was decided that allowing educational manufacture of the flight case would provide a fun way to engage students in the programme. Having a replica of the flight computer also allows students to prototype and test the physical interactions of the crew with their experiment. Making it affordable was also a priority, and 3D printing technology provided the solution, though some changes needed to be made to the CAD files to allow for the exigencies of the printing process.

The files were released as part of a free educational resource [16], along with a set of downloadable STL files for each printed part. The educational resource acts as the guide to printing, assembly, and testing.

5.4 Joint Station LAN operations

In preparation for future Astro Pi missions, work was carried out between April and June 2016 to qualify and test the Astro Pi payload for use on the ISS JSL.

This was intended to reduce the crew time requirement for future users. The largest allocation of crew time during the Peake mission was the manual file copy procedure, which was repeated several times. With the JSL connection, ground control teams can remotely access the Astro Pi computers to perform maintenance, upload and execute student software, and download experimental results without crew involvement. Deployment and stowage activities remain unchanged, but the total crew time requirement has been reduced by over 70%.

The qualification and test process was successfully completed in two months. This was achieved by developing a security-hardened version of the Raspbian OS, in consultation with a cybersecurity contractor employed by ESA. As Raspbian OS is a derivative of Debian Linux, this process was straightforward with all industry-standard software tools available. The work principally involved configuration and testing. The following software tools were used to make the OS compliant with ISS security requirements:

- ClamAV: antivirus and antimalware suite of tools.
- Samba: Microsoft Windows interoperability suite used to retrieve up-to-date ClamAV antivirus definition files from a NASA server on the JSL.
- Iptables: software firewall with rules to permit necessary protocols only.
- SSH and SCP, port 22 forwarded to ground: remote access to Linux command prompt and file transfer.
- Apache 2 HTTP and HTTPS, ports 8080 and 443 forwarded to ground: simple HTML status page for ground-to-space connectivity testing.

As JSL security policies exclude code compilation tools like GCC and G++, a waiver was required for the Python interpreters installed on the flight OS.

Remote access is achieved through the use of a ground laptop dedicated to Astro Pi, kept in a secure location. The laptop has VPN access to the MPCC ground node at ESA Col-CC in Munich. The MPCC ground node is a server that can access the NASA KUIPS service. This uses a network of satellites and can provide continuous TCP/IP connectivity to the JSL wherever the ISS is in its orbit.

The MPCC ground node will only provide IP access to the Astro Pi payload for the specific VPN login used on the ground laptop, and only at specific times. All software uploads, student or otherwise, are subject to the same testing and software control board evaluation as before. Once approval has been granted, access is scheduled on the MPCC ground node and ESA personnel plan their remote access activity to coincide with this. The following ESA document numbers can be consulted for further information on the security requirements for the JSL:

- ESA-ISS-COL-SEC-TN-0001 (ESA ISS List of Unsecured Protocols)
- ESA-ISS-COL-SEC-TN-0002 (MPCC Supported IP Protocols)
- ESA-ISS-COL-SEC-RS-0002 (Security Requirements for LAN Connected Payloads)

Future Astro Pi participants will not be permitted to use the JSL for any purpose. There may be circumstances in the future where use of the JSL by student software could be permitted for scientific research.

5.5 Sense HAT Emulator

To increase engagement with Astro Pi and remove lack of access to hardware as a barrier to participation, the Raspberry Pi Foundation worked with Trinket to create a web-based Sense HAT emulator that can be used from any internet-connected computer. The emulator presents a virtual Sense HAT on screen, with a window where users can develop Python code and test its functionality as if using a Raspberry Pi with the Sense HAT hardware. The sensors are emulated to provide input, and both these and the buttons can be manipulated by the user in real time to test their program. Output can also be given via the emulated LED display.

To accommodate offline users and models of Raspberry Pi with lower performance, a desktop application version of the emulator was developed by a separate contractor. Written in Python for cross-platform support, it can be run on any variant of Linux, though the intention is for it to be run on Raspbian. This emulator differs from the web version in that it will allow students to integrate other Raspberry Pi features, such as the Camera Module, with their programming. The user writes code in the Python IDE as opposed to the integrated code window of the web version. The code development experience is therefore closer to the way in which a physical Sense HAT is used.

The emulators can be accessed here:

- Web version: https://trinket.io/sense-hat
- Desktop version: https://sense-emu.readthedocs.io/

Code from either emulator can be deployed to a Raspberry Pi with a physical Sense HAT, and translates without modification.

5.6 Future developments

5.6.1 Principia conferences in the UK

In November 2016, the UK Space Agency will host two conferences to celebrate the educational work linked to Tim Peake's Principia mission. These events are an opportunity for students to show their projects to a panel of leading space experts, including Peake himself.

The Principia mission was linked to many educational activities and the conferences will be attended by students selected from all these activities. Raspberry Pi staff will also be attending both conferences to run workshops, and support students giving presentations by allowing them to borrow real Astro Pi flight units.

5.6.2 ISS Expedition 50/51 (Pesquet)

French ESA astronaut Thomas Pesquet is to fly to the ISS in November 2016 for expedition 50 and 51. The Raspberry Pi Foundation has been working closely with ESA Education to design a scaled-up Astro Pi competition, open to all ESA member states, as part of his mission.

The competition is the first attempt at developing a repeatable activity on a European scale similar to projects like CanSats, Mission X, and Spheres Robotics. It will target students aged 10-14 years, and will pose a single challenge which can be solved by writing Python software for the Astro Pi computer. The challenge will be designed to cultivate a wide range of solutions with no single perfect answer. Entries will be divided into two age groups and judged at national level, so that each member state supplies a maximum of two experiments to be uploaded and run on orbit, with the results returned to the students for analysis.

In support of this competition, ESA plan to fund 400 Astro Pi kits that will be distributed across the current ESERO offices. The kits will be provided on a temporary basis to selected schools, and will be supported by a set of cross-curricular learning resources and demonstration videos covering each Astro Pi feature. The kits will also include the parts required to build a 3D printed replica of the Astro Pi flight computer.

5.6.3 ISS Expedition 52/53 (Nespoli)

Italian ESA astronaut Paolo Nespoli is due to fly to the ISS in May of 2017 and current planning shows him having an Astro Pi activity as part of his mission. The precise form that this will take will be informed by the outcomes of the Pesquet mission.

The Astro Pi payload will be on board the ISS until 2022, leaving approximately four years of potential educational use after the end of that mission. Use of the payload is dependent on an ESA astronaut being on board.

6 Conclusion

During the last two years the Raspberry Pi Foundation has run a series of competitions to engage young people with computing and STEM subjects. This involved forging collaborative working relationships with the UK Space Agency, the European Space Agency, and several international aerospace companies through the Astro Pi programmes aboard the International Space Station. This has allowed children and young people to have an unprecedented opportunity to create something that will operate in space.

The 2015 competition for the Principia mission was largely successful, with some useful data produced. Some lessons were learned in the areas of hardware giveaways, crew time requirements, avoiding in-flight experiment problems, and incentivising follow-up research when the results were returned to ground. Some physical limitations in the payload were also identified, informing how it would be used in future missions. On a smaller scale, the second competition in 2016 demonstrated what is possible at short notice during the flight of an astronaut.

The Astro Pi programme has begun to plan orbit activities during four upcoming ISS increments (50 to 53), involving two further ESA astronauts. These activities will be optimised by JSL operations, and their associated Europe-wide ground-based outreach will be informed by lessons learned to date. Issues with access to hardware, which have previously been a limiting factor for competitions, will be addressed by free-to-access software emulators. These will play a significant part in future educational outreach.

Astro Pi will continue to deliver on the educational goals of the Raspberry Pi Foundation, the UK Space Agency, the ESA, and the aerospace partners involved. Having successfully demonstrated the possibility of running code written by students on the ISS, the programme will continue to engage as many children and young people as possible with this experience of computing and STEM.

Acknowledgments

The Raspberry Pi Foundation wishes to thank colleagues at UK Space Agency, European Space Agency, BIOTESC Switzerland, TLOGOS Italy, Surrey

Satellite Technology, Airbus Defence and Space, CGI Group, QinetiQ Space, UK Space Trade Association, ESERO UK, Knowledge Transfer Network for Space, and Nesta.

Finally, much is owed to British ESA astronaut Tim Peake, who has been so supportive of the project from the start.

References

[1] Raspberry Pi Learning Resources: Astro Pi Flight Data Analysis, https://www.raspberrypi.org/learning/astro-pi-flight-data-analysis/, (accessed 01.09.16).

[2] R. Hayler, Astro Pi data from the International Space Station, 7 March 2016, http://richardhayler.blogspot.co.uk/2016/03/astro-pi-data-from-international-space.html, (accessed 02.09.16).

[3] Raspberry Pi Learning Resources: Exploring space with Minecraft, https://www.raspberrypi.org/learning/exploring-space-with-minecraft/, (accessed 02.09.16).

[4] Principia Mission: Earth Observation Detective, https://principia.org.uk/activity/eodetective/, (accessed 02.09.16).

[5] D. Honess, Astro Pi: Mission Update 9 – Science Results, https://www.raspberrypi.org/blog/astro-pi-mission-update-9-science-results/, (accessed 02.09.16).

[6] Astro Pi: Science Results, https://astro-pi.org/competition/science-results/, (accessed 02.09.16).

[7] GitHub: Astro Pi organisation, https://github.com/astro-pi/, (accessed 02.09.16).

[8] Raspberry Pi Learning Resources: Sonic Pi Lessons, https://www.raspberrypi.org/learning/sonic-pi-lessons/, (accessed 01.09.16).

[9] S. Aaron, Code Music with Sonic Pi, first ed., Raspberry Pi Trading (Ltd), Cambridge, 2016. https://www.raspberrypi.org/magpi/issues/essentials-sonic-pi-v1/, (accessed 01.09.16).

[10] The MagPi Magazine, https://www.raspberrypi.org/magpi/, (accessed 01.09.16)

[11] Raspberry Pi Learning Resources: Getting Started with Sonic Pi. https://www.raspberrypi.org/learning/getting-started-with-sonic-pi/, (accessed 01.09.16).

[12] S. Aaron, Code Music with Sonic Pi, first ed., Raspberry Pi Trading (Ltd), Cambridge, 2016. https://www.raspberrypi.org/magpi/issues/essentials-sonic-pi-v1/, (accessed 01.09.16).

[13] GitHub: Astro Pi organization, Examples of MP3 playback known to work on the Astro Pi Raspbian image, https://github.com/astro-pi/flight-mp3-examples, (accessed 01.09.16).

[14] Raspberry Pi Learning Resources, https://www.raspberrypi.org/resources/, (accessed 01.09.16).

[15] Raspberry Pi Learning Resources, https://www.raspberrypi.org/resources/, (accessed 01.09.16).

[16] Raspberry Pi Learning Resources: 3D Printed Astro Pi Flight Case, https://www.raspberrypi.org/learning/3d-printed-astro-pi-flight-case/, (accessed 01.09.16).