National Aeronautics and Space Administration

Raspberry Pis for Space Guideline

Steven M. Guertin
Radiation Effects Group
Jet Propulsion Laboratory, California Institute of Technology

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

National Aeronautics and Space Administration

Raspberry Pis for Space Guideline

NASA Electronic Parts and Packaging (NEPP) Program
Office of Safety and Mission Assurance

Steven M. Guertin
Radiation Effects Group
Jet Propulsion Laboratory, California Institute of Technology

NASA WBS: 724297.40.49
JPL Project Number: 107590
Task Number: 40.31.55.03.05

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

http://nepp.nasa.gov

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration (80NM0018D0004). This work was sponsored by the National Aeronautics and Space Administration Electronic Parts and Packaging (NEPP) Program.

TABLE OF CONTENTS

1. Raspberry Pis for Space Guideline

This is a guideline to assist those interested in deploying a Raspberry Pi in a spacecraft build [1-4] with a primary emphasis on the radiation reliability issues involved. This guideline is intended to cover most of the critical issues while also trying to balance the high likelihood that such a user is limited in options to avoid the issues raised. This guideline strives to be consistent with the level of risk that such a user is likely to accept while also considering that the level of risk may vary wildly from one user to the next – so it is presented in the context of the high-end of a minimal system-level approach, consistent with the NASA Board Level Proton Testing Book of Knowledge [5]. Readers can interpret the presented materials in light of their own need to operate a lower level of cost or reliability than presented here.

This guideline is intended only to provide support and background understanding. It does not focus on solving any of the many Raspberry Pi implementation issues. In some cases, alternatives are suggested. In some cases, critical issues related to ensuring reliability will be discussed with the expectation that they will likely involve recommendations that cannot be used.

The final point to make in what is presented here is that the goal is not to show a user how difficult it will be to use their device in a space mission. In practice, the majority of reliability issues that take a Raspberry Pi outside of the normal flow for reliable spacecraft will yield anomaly or system failure rates that are below the limit for the user while being very high compared to a normal risk matrix for a typical NASA mission. That is, users will probably find that Raspberry Pis, provided they make it to space and function at all, will have failure rates well below other components of a spacecraft built in a similar approach. Further, some of the considerations discussed here are for ensuring lower than 1% chance of failure of the mission due to the device – which means many users with much higher acceptable risk will consider some issues relatively minor.

In the event users are considering using Raspberry Pis in a situation similar to a normal classification of a NASA mission, this guideline is not recommended. Users are highly recommended to contact the author, the NEPP program, or similar experts available to them through their program's contacts (see also Section 11.1).

2. Overview

A significant number of users have operated Raspberry Pi computers in space [2-4][6]. Some of these users have used the Raspberry Pi as a way of performing outreach and pulling in the public hobby and education interest areas, such as Astro Pi [6]. Others have simply included them because they had extra SWaP (size, weight, and power) space in their budget and did not see them as a major risk if they ceased to operate properly. The overwhelming majority of Raspberry Pis indicated for flight come from low Earth orbit (LEO) missions. LEO is among the most benign environments in which a spacecraft can operate, from the radiation point of view. Because of this last statement, the typical argument of heritage is of very minimal use for indicating the likelihood of a Raspberry Pi providing sufficient reliability for a space mission. However, in the modern launch environment, the overall cost of some types of payloads may be as low as $10,000, and the risk may be worth keeping the price down.

Raspberry Pi computers can (depending on the system communication architecture) be added to a design for as low as $5 (ignoring software development costs and difficulties in actually bolting it in). Thus, the use of Raspberry Pis in some flight systems may be seen as warranting some risk, if they can give another computing device that can offload work from a critical flight system.

This guideline seeks to minimize the added risk of adding a Raspberry Pi to a flight system. While they can add processing capability very cheaply, it is also possible that they can be implemented unnecessarily, or in such a way that they are unlikely to work much of the time (if at all).

A. Goals and Scope of This Guideline

The principal goals of this guideline are the following:

1. List the types of applications that are typically run on Raspberry Pis, indicating alternatives that may fit the need.
2. Explore the types of problems that users are likely to have and the key hardware implementation considerations.
3. Provide estimated error rates and possible permanent failure probability due to radiation effects for general Raspberry Pi computers if included in a space mission.
4. Review the principal sources of other failures that tend to be unique in space mission, as a set of items that users may be interested in researching further.

B. Radiation Effects as a Critical Issue

Computers for space environments must be able to withstand space-specific challenges. This section discusses general issues, though the focus of this guideline is radiation effects. Some of these challenges may constitute residual risk that a user should consider in addition to the discussion here. Raspberry Pi computers will likely do well against the majority of challenges because they tend to have very good overall reliability.

The key radiation issues considered in this guideline are single-event effects (SEE) and total ionizing dose (TID). SEE encompass single-event functionality interrupt (SEFI), single event latchup (SEL), and single-event upset (SEU). It should be noted that there are other SEEs, including damaging events, that we do not single out in this guideline. It is only an overview of the most critical issues. SEL is a damaging event, but in later sections the focus on damaging events is agnostic of the particular mechanism. The only reason SEL is pointed out is because it specifically results in high current and represents a class of SEEs that require system-level protection. SEFIs, on the other hand, refer to just about any SEE that produces an observable behavior that is problematic and not easily categorized. TID refers to general degradation of electronics through the accumulation of energy through ionization over time.

C. Space Radiation Failures of Consumer-Type Electronics

“You don’t hear of people saying their Raspberry Pi’s failed” might be a typical response if you ask or try to get information on space failures of Raspberry Pis. However, aside from the Astro Pi program, it is not clear that a space failure of a mission with a Raspberry Pi would ever be reported. And if it were, it would be difficult to show the Raspberry Pi as the root cause.

There is no good resource for reviewing failures of consumer electronics in space. Diagnosing failures of equipment that cannot be recovered is very difficult. And in the cases where a diagnosis was made, it is almost never shared in a way that lends to easy review of examples.

There are, nevertheless, many examples of consumer electronics that repeatedly have errors, or simply fail, due to space radiation. The worst examples have been units that cannot reliably operate for more than a few hours, when the original plan was that they would “usually” work for a day or so as a worst-case estimate.

Relevant to this discussion are NAND flash memories and SD cards. Note that [7] has reported on SD cards that fail with protons with a cross section so high that their expected failure rate is above 1 event every 3 months in LEO (where 1e10/cm^2 protons equates to roughly 5 years of exposure).

D. Flight Heritage

Flight heritage refers to the argument that a system or component is likely to work for a mission because another one was used on another mission and it seems to have worked. In some cases the implementation of the article may have been done with monitoring, so that things like TID-induced increases in current can be monitored. Those situations can improve the heritage argument for a unit being considered for flight. Unfortunately, for Raspberry Pis, the heritage argument has significant limitations. See Section 10 for more information.

3. What Is Raspberry Pi?

Raspberry Pi computers are single board computers that have been designed with the Internet of Things being among the original intended targets. In practice, however, Raspberry Pi computers are entirely capable of running a desktop Linux distribution. And they can run custom hardware over WiFi. In fact, there are many things people do with Raspberry Pis. The most powerful Raspberry Pis provide quad-core ARM Cortex A72 processors.

A comparison of Raspberry Pi models is provided in Table 1, below.

Table 1: Comparison of Raspberry Pi models. Note that OTG is the abbreviation for USB "on the go".

Year	Pi Model	Processor	Core	# Cores	GPU	RAM	USB	Ethernet	WiFi	Typical Power (5V)
2020	Compute Module 4	BCM2711	Cortex A72	4	Yes	1-8 GB	2.0[*]	GigE	Yes	5 W
2019	4B	BCM2711	Cortex A72	4	Yes	1-8 GB	3.0 & 2.0	GigE	Yes	6.25 W
2018	3B+	BCM2837	Cortex A53	4	Yes	1 GB	2.0	GigE	Yes	6 W
2017	Zero W	BCM2835	ARM1176JZF-S	1	Yes	512 MB	OTG		Yes	5 W
2015	Zero	BCM2835	ARM1176JZF-S	1	Yes	512 MB	OTG			800 mW
2021	Pico	RP2040	ARM M0+	2		264 KB[+]	1.1			500 mW
2018	3A+	BCM2837	Cortex A53	4	Yes		2.0		Yes	6 W
2016	3	BCM2837	Cortex A53	4	Yes		2.0	10/100	Yes	7 W
2015	2	BCM2836	Cortex A72	4	Yes	1 GB	2.0	10/100		4 W
2012	B	BCM2835	ARM1176JZF-S	1	Yes	256 MB	2.0	10/100		3.5 W

[*] - The Raspberry Pi Compute Module 4 is an interposer that has to be connected to a board that breaks out the physical peripheral functions.

[+] - The Raspberry Pi Pico’s RAM is provided on-chip, inside the processor.

The original approach was to provide a platform where users could use C++ or Python, in a Linux or similar operating environment, to provide access to simple hardware peripherals. Over time, the models have become either more capable from a computing perspective (such as the 4B), or more focused on providing a lower foot print hardware interface, such as with the Pico and Zero.

4. What Are Raspberry Pis Used For?

There are many different applications that a Raspberry Pi can be used for. The primary thrust of Raspberry Pi users tends to be some sort of localized Linux operation usually with the ability to run a high level programming language or scripts. Because of the nature of some Raspberry Pis, a unit could literally be inserted into something and operate as an Internet of Things (IoT) device providing remote hardware access for a relatively trivial piece of hardware via secure shell. Or they could perform autonomous operation of a remote-controlled car. Below is a table of typical applications that a Raspberry Pi might be used for.

Table 2: Typical Raspberry Pi Use Cases

Application/ Description	Enet/Wifi	Quad Core	GPIO	Camera	USB	SATA	Other
Sensor Monitor Log and control remote sensors	Yes	-	Yes	-	?	-	?
Remote Operations/Actuator Translate network to hardware to operate devices	Yes	-	Yes	-	?	-	?
Desktop PC Use Raspberry Pi as desktop PC	Yes	Yes	-	-	Yes	Yes	AV Connections
File Server/Storage File system/repository	Yes	?	-	-	Yes	Yes	-
RC/Robot Control Remote operation of car/robot	WiFi Only	?	Yes	Yes	?	-	May use AI/Automation
Data/Image Analysis Run classification or AI offline	Yes	Yes	-	-	?	?	-
Camera Use as camera/record & transmit images; can be used for stop motion or timelapse	Yes	?	-	Yes	Yes	Yes	-
System/Network Monitor Use Raspberry Pi to monitor system/network and issue maintenance commands	Yes	-	?	-	?	-	-

5. Standard Issues for Space Computers

Below are some standard issues with bolting a Raspberry Pi into a spacecraft. Some of these are not obvious how to handle with typical Raspberry Pis because they require specifications and testing that the manufacturers just are not focused on. It is possible, however, that most Raspberry Pis will meet many of these challenges through “good manufacturing”.

A. Hardware Considerations

Flight hardware usually is designed for challenges of transit to, and operation in space. Raspberry Pis are special in that it is not expected that users will modify the printed circuit boards (PCBs), and it is not expected that users will design their own alternate hardware. That leaves for consideration only how the Raspberry Pi is held in the hardware and how connections are made to it. The following hardware considerations may be critical for missions that wish to deploy a Raspberry Pi.

1) Thermal

Raspberry Pi thermal control is handled in terrestrial environments through convection. They depend on heating surrounding air and the natural processes by which the hot air escapes and is replaced by colder air. Since this doesn’t happen in space, Raspberry Pis must be conduction cooled. If no conduction path is provided on purpose, then the components on a given Raspberry Pi will heat the base circuit board, which will then heat whatever mounting system is used to hold the board. This may include several contact points before heat is connected to any radiators on the unit. For some Raspberry Pis, like the Zero, this may not be a significant problem. However, it is recommended to make a strong thermal connection between essentially all of the ICs on the Raspberry Pi, or perform a thermal analysis to identify the primary heat producers.

2) Power

This guideline assumes that a user is going to use a Raspberry Pi single board computer (SBC). It also assumes that power is going to be delivered to that board through a standard USB power port. There are a significant number of options for how to provide power, but it must be fused in some way. If the Raspberry Pi gets a certain class of destructive events, it can draw high current. Thus, the system should be monitoring for this and attempt to shut it down. In practice, such an event usually comes with the unit exhibiting symptoms of a SEFI, but this is not always the case, so users should be ready.

If fusing of the power is available, it has the added benefit of enabling power cycling of the unit and possibly powering the unit down to conserve power.

3) Communications

Ideally, any piece of hardware that is going to be included as a black box should be isolated. It is recommended that all communications connections to Raspberry Pi units include an electrical gap. Fiber optics can provide this. Even if this can be achieved, the system should also be prepared for the possibility that the Raspberry Pi becomes a source of garbage communications. A good solution here would be to cut power if it is suspected that the Raspberry Pi has begun creating noise on a communications channel.

4) Vibration

This is included for completeness but is not considered to be a serious risk for a Raspberry Pi. During launch, spacecraft are exposed to significant vibration (including shock) and g-loading. The majority of modern electronics does not have significant amounts of material that might be damaged during this environmental stress. However, if this is a serious concern, it is recommended to check against industry standards for recommendations to minimize risk.

5) Charging

Isolated and non-conductive surfaces in spacecraft collect charge during flight. It is highly recommended to avoid using any of the standard plastic cases available for Raspberry Pis. Instead, a metal or otherwise ESD (electrostatic discharge) dissipative mechanical connection to the spacecraft is recommended. Typically spacecraft include a conformal coating that provides ESD dissipation on surfaces.

B. Likely Failures or Errors and How to Avoid Them

Raspberry Pis in space will (eventually) have a handful of SEEs. Some SEEs, and long-term TID (likely more than a few years in LEO) may eventually render a Raspberry Pi non-functional. We indicate the error modes or system failures and how to avoid them.

6) Permanent Failures

It is strongly suggested to be able to disconnect a Raspberry Pi from the flight power systems. TID will eventually render a Raspberry Pi non-functional, and given the commercial nature of the power devices, this may happen within only a few months to years (depending on the exact components).

An obvious additional requirement for handling permanent failures is the ability to detect a failed Raspberry Pi and indicate to the flight hardware to disconnect it.

This may be considered a simple part of the recovery scheme for some of the other issues indicated below, because if the system can disconnect the Raspberry Pi, presumably it can reconnect it. This gives a straightforward way to do a temporary disconnection in the event that a Raspberry Pi is in a SEFI state. Unfortunately, some simple-to-fix permanent failures, such as an SD card that needs to be reprogrammed, are likely to be outside of the capabilities of the flight systems.

7) SEFIs

SEFIs are actually the most likely events to be observed on a Raspberry Pi. The reason is because the software systems typically are not robust to errors and will require to be reset. The primary issue with a SEFI is how the flight systems detect a SEFI. Once detected, the most reliable way to handle a SEFI is to simply turn the unit off for an extended period of time, then turn it back on.

6. Raspberry Pi Radiation Effects

There is a small amount of existing radiation data. Some additional data was taken by the NEPP program in support of this guideline. Unfortunately, none of it is very comprehensive. Raspberry Pis are essentially full systems, and typical radiation test efforts are focused on components and analysis of system level responses to component behavior. Full understanding of the unit's radiation response is outside the scope of the data that are available.

It is, therefore, necessary to consider Raspberry Pis as units and try to understand the limitations of the existing radiation data. Then try to extract general information which will be used in Section 7 to highlight likely issues in various environments.

C. TID Performance

Several groups have explored the TID performance of various Raspberry Pis. The NEPP program explored the Raspberry Pi B and showed that it worked (up to functional benchmarks and log in) up to 40-50 krad[Si] [9]. Decena, as part of work towards the Opal CubeSat by Utah State University showed functionality of Raspberry Pi Zero up to 100 krad[Si] [10]. Similar data was collected by G. Toumbus [11] on a Raspberry Pi compute module 3.

For this guideline, the NEPP program looked at the TID performance of the Raspberry Pi Zero and Raspberry Pi 3B+. The testing required a full boot to nominal operation and monitoring of operational characteristics during characterization tests. As a result, both units showed no degradation of any kind at 10 krad[Si]. At 30 krad[Si], all units showed some degradation, though the 3B+s performed better. Some of the Raspberry Pi Zeros actually hanged or failed to perform some of the tests.

These tests all have a general difficulty in common. That is that there is no good way to ensure that a user application will actually work. The reason this occurs is because the test conditions do not obviously cover all possible use scenarios. However, they do all indicate that there is no obvious issue with any Raspberry Pi up to 10 krad[Si] (some of the tests seem to indicate much higher levels, but appear to have limitations of their approach or are not calling it a failure when a unit has its first reduction in performance). This information is used in the radiation environments section.

A further problem with TID performance is that it generally cannot be guaranteed that just because one test resulted in a given TID level, another test of another Raspberry Pi will give the same results. The issue is that for TID results to carry from one unit to another, a strict parts program is typically required where the wafer lot of all of the components of the test unit has to match that of the flight unit. It is understood that will probably not happen for a Raspberry Pi user, but it is a serious limitation of the predicted TID performance of a Raspberry Pi system.

D. SEE Performance

There is essentially no current SEE performance for Raspberry Pis available. A test was conducted as background work for this guideline, but the linear energy transfer coverage was very minimal. This means that the test likely missed what are classified as rare events from space particles with a flux of less than 100/cm^2-year. The key result for SEFIs was that Raspberry Pi Zero's appear to have a saturated cross section of about 3e-4. A plot of the results is shown in Figure 1, below. This information is used in the radiation environments section.

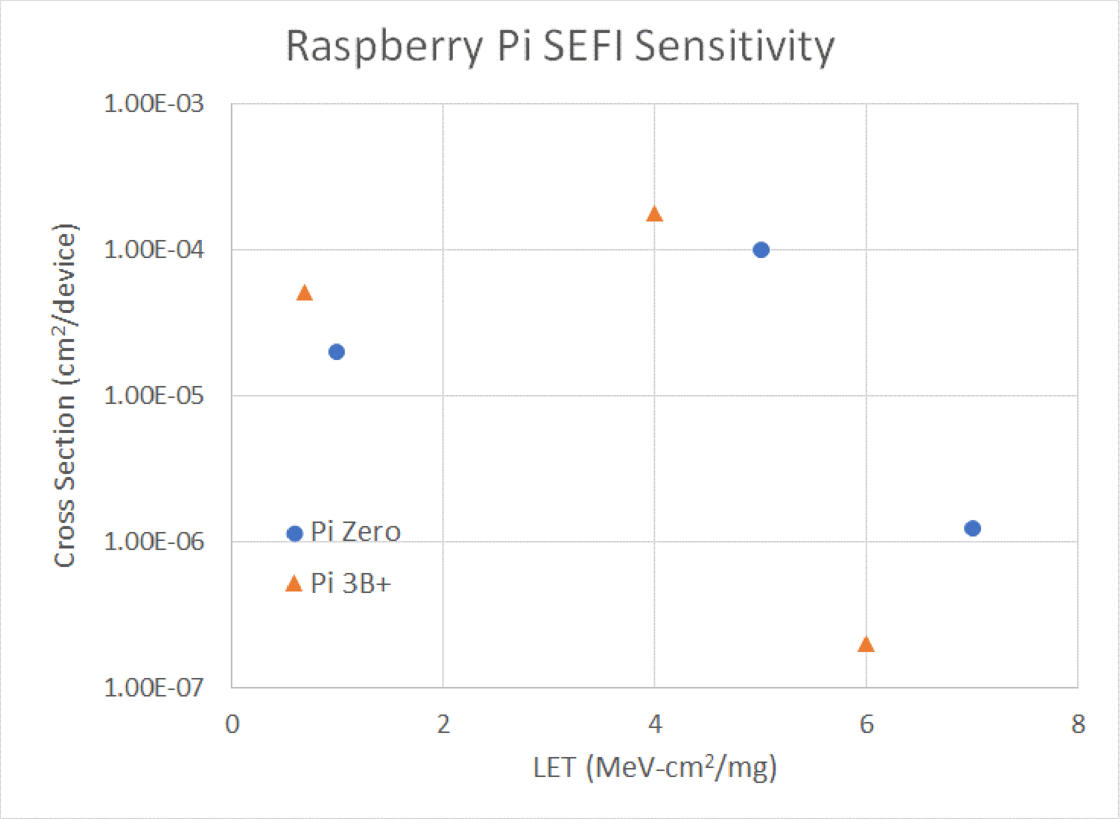

Figure 1: SEFI sensitivity of Raspberry Pis - note that above LET 5 the ions are ranging out and the cross section reduces. The onset LET appears to be below 0.6, and saturation is roughly 3e-4cm^2.

7. Radiation Environments Considered Here

The most common environments of interest for users of Raspberry Pi fall into two categories, both driven my mission cost. The basic argument we use for this consideration is that a mission that is going to fly a Raspberry Pi is likely dependent on very low cost to launch. In some cases, large deep space missions may be providing very low-cost access (either now or in the near future) to places like Mars orbit. But the known short-term options cover the majority of any deep space orbits. The three categories we will consider are: Low Earth Orbit (LEO), Geostationary Earth Orbit (GEO) – which is discussed with interplanetary space, and Lunar orbit.

A. LEO Orbit

This is where essentially all Raspberry Pis that have flown up to 2021 have been. There are three general categories of interest – low inclination, high inclination, and polar.

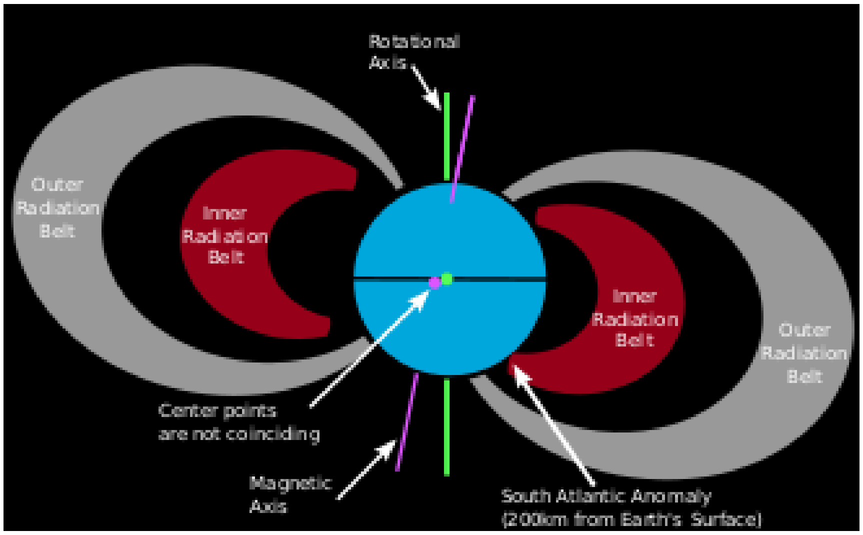

Figure 2: The magentic field of the Earth is not aligned with the rotational axis. This gives rise to the South Atlantic Anomaly, where the inner proton belt drops into the LEO altitude range.

TID levels in LEO are dominated by flares (for polar missions) and how they pump up the trapped particle belts. Also, LEO TID is very sensitive to exactly what altitude the orbit is in – while LEO is supposed to be "under the belts", in practice LEO orbits skim the belts. The belts are indicated in Figure 2. As a result, if the orbit is 1000km, TID will be much higher than the 400km of the ISS.

8) Low Inclination

In LEO orbit, typical quoted TID levels are about 1-2krad[Si]/year (behind 100 mils of Aluminum), with a possible increase due to flares (very minimal in low inclination). Thus a Raspberry Pi is likely to survive here to at least 5 years without issues (though it will likely deorbit before then).

For SEE, low inclination is essentially driven by the South Atlantic Anomaly. Given the SEFI cross sections quoted above, Raspberry Pis can expect SEFIs with an upper bound rate of about 1/100 days in low inclination LEO. In practice the rates will probably be about 3x below this rate.

9) High Inclination

High inclination typically refers to the ISS, which orbits at an inclination of 51.6°. A lot of CubeSats that launch with rockets to the ISS end up in this orbit. At this inclination, a significant amount of time is spent both in the trapped radiation belts and in unprotected space.

In practice, because of various tradeoffs, the TID and SEE levels in this orbit are almost the same as low inclination. So, the results above hold – at least 5 years of TID survivability, and SEFIs on the order of once every 100 days. Note, however, that a fair amount of Raspberry Pi on-orbit knowledge comes from the Astro Pi program. These units are not only inside a ruggedized Al enclosure, but they are also in the astronaut enclosure of the ISS. This is key because the ISS itself adds shielding to bring the natural radiation down from ~1-2krad[Si]/year down to about 30rad[Si]/year – to protect the astronauts. As a result, the radiation environment inside the ISS should be considered very different than that outside.

High-inclination orbits also are more likely to be affected by solar flares. As a rough rule of thumb, a solar flare may be as much TID as a full year of TID accumulation. It also is likely to overwhelm electronics giving a few months of exposure in a few minutes or hours. It is expected that most electronics will have problems during solar flares.

SEE rates in high-inclination orbits for Raspberry Pis are expected to remain in the 1/100 days range. In practice the rates will probably be about 3x below this rate.

10) Polar Orbits

Rather than review polar orbits in detail here. We will instead indicate that above about 40° the natural space environment has very little geomagnetic shielding (though it does still get protection from solar events), but it also has very little trapped particles. As with high-inclination, the exposure tends to average out except that the high LET component increases significantly at higher inclinations. This means that the chances for a damaging SEE increase. Unfortunately, we do not have a good estimate of the rate of damaging SEE for Raspberry Pis. If we did, we would have to warn that data from one Raspberry Pi likely does not apply to another – even the same model.

Thus, the predicted TID performance will be at least 5 years. Each flare will likely add a year of TID exposure per year.

SEE rates in polar orbits for Raspberry Pis are expected to remain in the 1/100 days range. In practice the rates will probably be about 3x below this rate. Damaging SEEs may be much more common but we do not have a means to estimate them. This could mean that only a few months of exposure might be sufficient to damage a Raspberry Pi.

B. GEO Orbit/Interplantary Space

Generally speaking, the main contributor to SEE risk in this orbit is galactic cosmic rays (GCR). Without the Earth's trapped particle belts, the SEE rate from GCR for the SEFI sensitivities reported in Section 6.2 give rates about 5x higher than in LEO. That is, the predicted rate is closer to 1/20 days for a SEFI as the high estimate, and the best guess is about 3x lower, or about 1/60 days.

In GEO orbit spacecraft will still be in the outer trapped particle belts of the Earth, and this is not so good. About 10 krad[Si]/year is the likely dose rate, so Raspberry Pis in GEO orbit may have trouble with TID after the first year.

In interplanetary space, TID is very minimal unless the spacecraft is hit by a flare. Typically TID is below 500 rad[Si]/year, so a Cubesat can survive 20 years in interplanetary space.

C. Lunar Orbit

Lunar orbit is actually a special case of the information presented below. We present it here because it appears likely that Cubesats and other low-reliability/classification spacecraft may be likely to head to Lunar orbit in the coming decade.

The moon has no radiation sources of its own that would impact spacecraft. The principal possible source would be a magnetic field with trapped particles. Instead, the moon is actually more benign than interplanetary space (but much more dangerous than non-polar LEO). The primary cause is because the moon blocks radiation from the portion of the sky that it fills (i.e. the solid angle subtended from the spacecraft's point of view).

Thus both TID and SEE levels range from the full interplanetary amounts given above, to about ½ those values when in close orbit of the moon. The Lunar Reconnaissance Orbiter (LRO) orbits the moon at about 50 km, and sees these levels. Thus for low altitude lunar orbiters, TID can reach 40 years (provided no flares), and SEE rates will be about 1/40-1/120 days for SEFIs.

D. Solar Flares

Some of the sections above pointed to solar events. This section provides the same information but in a place that is easy to find. Solar events are particularly bad for electronics because they can deliver months of exposure in only a few minutes or hours of time. As a result, if a spacecraft is in the line of a flare, the typical response is to go into a safe mode of some type and "wait it out". Because of the event rates in the normal environment for a Raspberry Pi, there is a high likelihood it will have a SEFI during a flare. Typically, some sort of watchdog will observe the system having problems and either shut it down for ~48 hours, or try to reboot it right away – repeat the process a few times – then shut it down for ~48 hours.

Solar flares are significantly mitigated by the Earth's geomagnetic shielding. Some of this extends near the poles, but high inclination systems are still at pretty high risk. Flares tend to energize the trapped particle belts. And they will impact polar orbit spacecraft as well. Typically, Earth orbiters take about a year of exposure from a single flare. In Lunar orbit, or interplanetary space, solar flares can deposit as much as 5 krad[Si], which is easily 10 times the normal exposure for a year. Typical flares can vary considerably.

E. Other Orbits

We did not consider Mid-altitude Earth Orbit (MEO), which sits directly in the Earth's trapped particle belts and easily collects more than 20 krad[Si] per year. We also left out orbits of other planets, because in general they are either a subset of GEO, or their radiation risk is so bad that an program sending a spacecraft there must use a viable parts/radiation approach that would preclude a Raspberry Pi.

Cruise to other bodies in the solar system fall into the interplanetary designation and are generally less stressful than GEO. In practice, GEO has essentially all of the interplanetary radiation risk with added TID risk due to possible extension of the upper trapper particle belts. Without the belt component, GEO and interplanetary space are very similar (up to the level of detail considered in this guideline).

Orbit of other bodies without trapped particle belts (i.e. without a magnetic field) can be considered interplanetary space except that the body obstructs radiation from the solid angle of the sky that it blocks. If orbit is very close to the body, then a maximum of ½ of the incoming radiation is blocked. This specifically applies to the moon and Mars orbit.

8. The Flash Memory on Raspberry Pi

Raspberry Pis use Flash memory in the micro SD form factor. In other places in this guideline we simply provide the most general guidance possible regarding these. However, in this section we will briefly explain why we cannot provide more specific guidance on the use of SD cards for space missions.

The Flash memory technology used in SD cards comes from NAND Flash (ref), and can come in sizes from 1 GB up to 1024 GB. This is a tremendous range over which to provide general radiation recommendations, or even address applicability. Typically, users find it difficult to obtain Flash memory cards for Raspberry Pi applications that are less than 16 GB. And it is not obvious why anyone would need (or trust) 1024 GB SD cards. Keep in mind that some generations of SD cards are intended for camera use, where the likelihood they will ever have data deleted is relatively low, so they may be wholly inadequate for anything like a computer application.

Because the Raspberry Pi uses the SD card in order to boot, errors in the SD card are critical. Luckily, SD cards have a fairly high expected bit error rate even on the surface of the Earth, and the potential increase from SEE in space is relatively minor. But there are other radiation issues with Flash memory that cannot really be addressed in this guideline.

From a system architecture point of view, such a memory system for flight would have some sort of backup with an automatic fallback if the code image on the Flash memory were found to be corrupt.

The TID risk is considered the greatest for Raspberry Pi NAND flash. And the high likelihood that the user will be unwilling to refresh the boot image, combined with the potential for outright failure, are likely to create the most significant limitation to use.

But there is good news. There are a huge number of available SD card sizes and manufacturers. In practice, the older “single level cell”, or SLC NAND technology tends to perform best for radiation effects. But for many devices, it is unclear if general erase and write due to other cell technologies will fail first, or if the device’s charge pumps may fail first.

As indicated earlier in this section, however, many SD cards may not have anywhere near the erase/write cycles desired by the user. They may (depending on the manufacturer) implement wear leveling technology intended to minimize problems when overwriting data. I may actually be very difficult to prove or interrogate the wear leveling technology in a way to understand if it will impact TID performance.

9. What Will a Failure Look Like?

There are a number of likely failures that may beset a Raspberry Pi in a space application. Some failures will be resettable, a small number will appear as glitches that somehow do not impact anything, some will be permanent, resulting in the device being unable to boot, potentially for a large number of reasons.

Below is a list of device or system failures and how a user might identify them.

1. SD Card Failure – this is most likely caused by a charge pump failure due to TID (though it may also be caused by SEL or other permanent transistor failure in the SD card). This may immediately result in never being able to read boot or other data from the SD card. It may also manifest as a system that can partially or completely boot but has general failures any time the Raspberry Pi tries to write to the SD card.
2. High Current – during testing for this guideline it was observed that some Raspberry Pis may increase in their system-level current consumption by 2-3x between 10 and 30 krad[Si]. The user's actual TID sensitivity to high current may be significantly different. When high current draw starts due to TID, it typically shows failure in one of two ways. First, it may draw too much power and sag the power system resulting I the unit no longer functioning correctly (possibly giving lots of errors in driver initialization, if the Raspberry Pi shows any indication of life at all). Second, it may cause a thermal excursion with potentially catastrophic impacts across the rest of the spacecraft. As a potential side-benefit, if the Raspberry Pi overheats the system, it may inadvertently anneal its TID damage and begin working nominally – though it should be considered essentially at end of life because additional TID will continue to cause more problems.
3. Bit Errors (SBUs) (Detected) During Code Execution – typically these source from registers, caches, scratch SRAMs, file buffers, and DRAMs (DDR4, etc.). The system behavior depends heavily one where the bit errors originate, and what type of error protection the various components and interfaces have. Typically, the most likely SBE sources are actually detected, with some being corrected and others simply elevated to the operating system which almost always gives up and crashes the system. Because of the risk of terrestrial SBEs, any space user is likely to have a very large number of detected SBEs before seeing undetected errors. Unfortunately, because of the high relative error rates, undetected SBEs in space are also likely to happen.
4. Undetected SBEs During Code Execution – these are the same types of things as above, but under the scenario where the event is not detected (or corrected). The typical sources are the same as above, though many sources, like Caches, are almost always detected unless the situation is highly atypical. The primary issue with undetected errors is how they will manifest. They could result in sending a message to perform a physical action ("firing pyros" is the example that is typically used to describe a worst-case incorrect physical action. Anything like this can happen, but it depends greatly on the code. And there is a primary saving grace here – the user should be experiencing a high number of these during operation if one might suddenly show up while doing something critical. An example of typical SBE impacting an executing code is that a loop control variable may be modified so that a fill algorithm goes past the end of its memory space. The majority of these will result in SEFIs (see below).
5. SEFIs are the grab-bag of SEEs – they have to (should) come with no other clear indication or the problem. In practice, they often get assigned to events that are very likely due to a bit error in a register. The fundamental observable is that the system does not know what to do and takes a drastic action. It should be pointed out that a lot of SEFIs will be identified by the processor because something strange and unexpected is happening. In practice, if a SEFI occurs on the processor, it is important to note that user code spends a lot of time calling kernel routines, so if Linux reports a Kernel panic, it doesn't immediately indicate a kernel or user issue. However, if Linux detects an error in a Kernel routine, it is possible it will kill the thread and hope, or kill it and halt the system.

10. Key Restrictions of Similarity Arguments

A. Raw Similarity Requirements

It is reasonable to point out that similar devices have flown on successful missions in the past. This argument is likely to be used as a justification for using Raspberry Pis in future space applications. Unfortunately, this argument only has limited application for Raspberry Pis. We list the key limitations on applicability of heritage arguments here.

The issue here is absolute knowledge versus hopeful evidence. If all details required to establish carrying information from one use scenario to another are verified, it is possible to come very close to showing that previous heritage

1. In order to ensure TID performance from one unit to another, it is necessary to match part numbers and lot/date codes as well as verify with the manufacturer that all parts for a given lot are manufactured in the same fabrication facility using the same machines. Unfortunately, this is not likely to happen for Raspberry Pi users. Instead, the closest they can get is to match all part numbers and ensure fabrication of the units at the same facility.
2. SEE performance also requires matching part numbers, especially on power or reference-type analog devices. Another source of problems that have been observed in the past are crystal oscillators – they should be exactly the same.

B. Where Similarity Fails

There is one class of events where similarity is almost impossible to establish based on heritage, and that is statistically rare events. The most obvious category of these is permanent failure. At best, if there are N missions flying a board that might permanently fail, then N failures might be observed (by contrast, for bit errors in non-critical memory, many events may be observed on a single mission, yielding much higher statistical significance). Worse, permanent failures are unlikely to be properly attributed to the component or mechanism that actually experienced the failure. For example, if 9 one-year missions are conducted using component X, and none of them observed permanent failure, the best-case scenario for the predicted rate is the upper bound 2-sigma rate of 3.7/9 missions, putting the worst case upper bound rate for the next 1-year mission at about 0.41/year. Even if this were somehow acceptable, it is actually likely wrong and underestimates the actual rate. Even if the mission durations are the same, it is not obvious that the usage cases sensitize all of the devices the same.

For example, SEL is worse in devices that run hotter, so if the new application runs hotter than the previous ones, 0.41/year may be an underestimate. Also, some missions may not bias the device at 100% duty cycle. Without details, heritage arguments are simply a guess, except for one key observation – many modern electronics have done very well in LEO missions, typically due to the very weak radiation exposure. That is, many systems have done well simply because the stress is minimal. In these cases, flight heritage argues more towards good overall design, workmanship, surviving launch, reliability, and ease of use.

C. Loose Similarity

The biggest problem with hard similarity requirements is that if you don't meet them, there is still a high likelihood that one unit will have a similar radiation response to another. For example, two LPDDR4 devices may be from two different manufacturers but have SBU and SEFI rates that are within a factor of

10 of each other. Given that the testing of the Raspberry Pi Zero and 3B+ for this guideline showed that the DDR devices give errors at only a few % of the other parts of the units, this means that DDR devices are not likely to be critical on Raspberry Pis.

That being said, power devices violate this principal regularly. Power devices are often used interchangeably – especially field effect transistors (FETs). FETs that are used near their maximum gate rating will fail much more regularly in space than those that stay below a safe operating area band, typically no more than 50% of the maximum gate voltage. A system could replace a "good" FET that had a rate of 0.01/year in LEO with another FET with a rate of 6/year in LEO with almost no indication that anything different is on the board.

11. Recommendations

At the time of writing of this guideline, the goal was to provide an overview primarily driven by expertise and best practices, rather than actual radiation data. The reasons for this are threefold: (1) the overall scope is intended to provide general understanding and not be an exhaustive study of a very large number of existing Raspberry Pi systems; (2) it is unknown the extent to which a potential user would be willing try to link their flight unit to data taken on a handful of sample Raspberry Pis; and (3) in some situations it was considered overly expensive to explore radiation effects of specific types to specific portions of the Raspberry Pi. Examples of the final point indicated include: package on package processor/memory arrangements that are very expensive to test with high LETs; flash memory SD cards can have significant radiation-related problems that could drive an experiment but be irrelevant to most users. Recommended follow-on efforts to this work include:

A. For Missions Following NASA or NASA-Like Parts Programs

If a user is considering using a Raspberry Pi in a flagship, Class 1, or any other NASA-categorized mission, using a Raspberry Pi is not necessarily a non-starter. However, it can be safely said that if the goal is to avoid buying or designing a $250k radiation hardened computer for flight, it might be less expensive to buy the radiation hardened option.

It is possible that there is no good solution for a very specific application, and the risk picked up application is worth keeping costs down. It is recommended that the actual risks and potential options be discussed with a NASA radiation effects expert. There have been very specific situations in recent years where this type of solution might be acceptable for various systems, especially if they are not flight-critical.

B. Do Some Radiation Tests

The most failure risk to a flight system is when it may become entirely non-functional. The primary risk for this type of failure is likely to come from TID. There is little evidence that a Raspberry Pi unit will encounter catastrophic SEE risk, although it is very hard to prove this without a lot more testing than was performed in the available literature (refs), and the SEE testing performed for this guideline.

The critical issue with TID is that tests performed on a set of units may not relate at all to another set of units. It is understood in the radiation effects community that lot acceptance testing is required for components bought in batches because TID performance can change significantly when no other performance parameters change (ref – recommendation to do LDC testing). Lot acceptance testing is (most likely) impossible. It may be possible to engage the Raspberry Pi foundation to enlist support to do a run of Raspberry Pis with user-selected components. But it is outside the scope of this guideline to provide support or recommend resources in this regard.

Although it will be less than ideal, the recommended test campaign is the following. The user should identify the typical environment, duration, and approach to handling flares (typically either assume one every few years, or ignore them completely). This sets a target TID level (usually at a few krad[Si]/year augmented by 3-5 krad per flare). The user then buys a set of Raspberry Pis that include the intended flight unit(s). From this set, the user constructs a lot acceptance test batch. The test units are then irradiated in a Co-60 TID lab, while operating, with periodic checks for boot and performance, known as characterization. Ideally the lab test TID will be 2- to 3- times the target mission TID.

For the TID testing performed for this guideline, both Raspberry Pi 0 and Raspberry Pi 3 B+ worked with no apparent impact to any operating parameters until after 10 krad exposure. This is consistent with some other tests [refs – NEPP], so it is likely that if the user application requires 10 krad or less, that Raspberry Pis will likely function sufficiently.

C. SD Card

It is highly recommended that SD cards be tested for TID along with the target Raspberry Pi units. A set of three or more options, covering multiple sizes, manufacturers, and if possible, feature sizes, should be considered.

12. Alternatives to Raspberry Pi

There are many different types of users of Raspberry Pis. As indicated in Section 4, there are also many applications. Because of this, the alternatives for space applications of Raspberry Pis run a wide range. Alternatives also depend greatly on the level of available funding for the alternative. The simple truth is that for ~$50, one can purchase, setup, and run simple hardware with a Raspberry Pi Zero using Raspbian with Python. It is hard to beat this price point for Python/high level language and OS control of hardware. It is even harder if the goal is to increase reliability. In coming up with a rough list of alternatives, some items are obvious but may not be known to some users. Other items simply do not have good alternatives. The table below is provided to give users a starting point for thinking about alternatives to Raspberry Pi applications.

Table 3: Generic "radiation tolerant" alternatives to Raspberry Pi use cases by price range

Application/ Description	$100-300 Option	$1000-$3000	$10k-$30k
Sensor Monitor Log and control remote sensors	Gang sensors into other hardware (possibly)	RH Microcontroller	Custom Space Computer
Remote Operations/Actuator Translate network to hardware to operate devices	Connect to other (RH) hardware (possibly)	RH Microcontroller	Custom Space Computer
Desktop PC Use Raspberry Pi as desktop PC	None	None	None
File Server/Storage File system/repository	None	None	Custom Space Computer
RC/Robot Control Remote operation of car/robot	None	None	Custom Space Computer
Data/Image Analysis Run classification or AI offline	None	None	None
Camera Use as camera/record & transmit images; can be used for stop motion or timelapse	None	None	Unknown
System/Network Monitor Use Raspberry Pi to monitor system/network and issue maintenance commands	None	RH Microcontroller	Custom Space Computer

Note: If using Raspberry Pi as a "desktop" computer, there is no viable space computer alternative in the $10-30k range that can provide similar processing performance. The indication in other applications expects that the Raspberry Pi's computing capabilities are not heavily utilized. Further, it is unclear if a "desktop" computer configuration can be supported in a spacecraft design.

Above $100k, a user can probably get a custom piece of space grade computing hardware for most potential Raspberry Pi applications. In the mid-range, $10k-$30k, things get a little dicey because there are some options but they get expensive fast and may not have the computing power or other performance metrics required. However, above $10k, the list provided in the Military and Aerospace Electronics website article on radiation hardened electronics gives a pretty good list of options [12]. The principal players for "low end" space computers are things like Cubesat Space Computer [13], and AiTech's SP0-S [14] as just a couple of a fairly long list of examples. The issue is the tradeoff between cost and performance, where the upper bound of performance is unfortunately rather low. In terms of raw performance, the Raspberry Pi 4, with its quad core A72 cluster, easily outperforms pretty much any space grade computer for less than $250k, provided the issue of space survivability can be ignored.

In terms of rad hard microcontrollers (RH Microcontroller), there are a huge number of options. Many of these would have been considered computer processors not that long ago. There are a number of Sparc-, ARM-, and even RISC-V-based space microcontrollers available from many different companies such as Microchip, Vorago, CAES, TI, etc.. It is understood that the convenience of the Raspberry Pi development environment is not duplicated in these RH microcontrollers, and it is not obvious that the price will stay in the range specified because some of these devices carry very expensive software requirements. So, in order to go this direction, some research will be required to ensure that a possible solution will actually meet the working price range.

13. Acknowledgement

The material in this guideline is partially based discussions and input from a handful of people who gave useful insight into the common issues faced by people implementing inexpensive solutions like Raspberry Pis. Those people include Edward Wyrwas, Michael Lowry, Richard Alena, Glen Steele, Trevor Turchan, Eli Kawam, and Heather Quinn. To the extent that the presented material is accurate and insightful, it is due to them and others who have provided much valuable feedback during the development of this guideline. I apologize to any contributors left off this list.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Aeronautics and Space Administration Electronic Parts and Packaging (NEPP) Program. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. The cost information contained in this document is of a budgetary and planning nature and is intended for informational purposes only. It does not constitute a commitment on the part of JPL and/or Caltech.

14. References

[1] Raspberry Pi website: https://www.raspberrypi.org/ (accessed September 2021)

[2] NASA PiSat Press Release "Celebrate Pi Day with NASA Goddard and Pi-Sat": https://www.nasa.gov/feature/goddard/2016/celebrate-pi-day-with-nasa-goddard-and-pi-sat (accessed September 2021)

[3] Surrey Satellite Technology Ltd. Press Release "SSTL releases spectacular Raspberry Pi camera image and video of the Earth", Surrey, Sep. 5, 2019.

[4] A. Zucherman, "Cislunar Explorers Mission Update" Cubesat Workshop, 2020

[5] S.M. Guertin, "Board Level Proton Testing Book of Knowledge for NASA Electronic Parts and Packaging Program", JPL Publication 17-7, Nov. 2017.

[6] Astro Pi: https://astro-pi.org/ (accessed Sep. 2021)

[7] Kimura, Et Al "Breakdown Phenomena in SD Cards Exposed to Proton Irradiation", Trans. JSASS Aerospace Tech. Japan Vol. 12, pp. 31-35, 2014 – puts rates at about 1 in 80 days for SD cards

[9] D.P. Violette "Arduino/Raspberry Pi: Hobbyist Hardware and Radiation Total Dose Degradation", EEE Parts for Small Missions, NASA GSFC, Sep. 2014.

[10] Mojica Decena, Jonh; Wood, Brian; Martineau, Ryan J.; Tayor, Michael; and Dennison, JR, "Radiation Damage Threshold of Satellite COTS Components: Raspberry Pi Zero for OPAL CubeSat" (2018). Utah State University Student Research Symposium 2018. Posters. Paper 84. https://digitalcommons.usu.edu/mp_post/84

[11] G. Toumbas, "Raspberry Pi Radiation Experiment" University of Surrey, May 2018

[12] J. Keller "The evolving world of radiation-hardened electronics", https://www.militaryaerospace.com/computers/article/16707204/the-evolving-world-of-radiationhardened-electronics, June 2018 (accessed Sep. 2021).

[13] Cubesat Space Processor Datasheet, Space Micro, https://www.spacemicro.com/products/digital-systems/CSP%20CUBESAT%20SPACE%20PROCESSOR.pdf (accessed Sep. 2021)

[14] SP0-S Space Computer by AiTech, https://aitechsystems.com/product/sp0-s-rad-tolerant-3u-compactpci-sbc/ (accessed Sep. 2021)