

Callisto-Lx: Callisto on Raspberry Pi

Whitham D. Reeve

1. Introduction

Callisto-Lx is a Raspberry Pi computer that uses the Linux operating system and application software for automatic and manual control of the Callisto instrument. The Callisto instrument forms the basis for the e-Callisto solar radio spectrometer network described at {[e-Callisto](#)} and is a software-controlled, frequency-agile radiometer operating in the VHF and UHF bands. The International Space Weather Initiative {[ISWI](#)} recognizes e-Callisto as an *Instrument Project* that is used worldwide. Data produced by Callistos around the world (and other solar radio spectrometers) are hosted by and freely available from Fachhochschule Nordwestschweiz (FHNW) in Switzerland.

Note: Internet links in braces { } and references in brackets [] are provided in **section 7**.

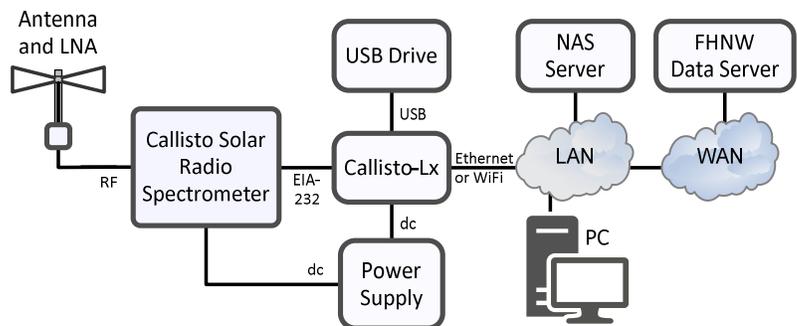
The original Callisto software developed by Christian Monstein at ETH-Zurich in Switzerland is freely available. It is Windows-based and provides a full set of features, reliable control of the instrument and data archiving. Callisto-Lx uses the application software originally developed for Unix by Juha Aatrokoski at Metsähovi Radio Observatory in Finland. It is freely available at {[Unix](#)} and is licensed under the General Public License (GPL) version 3. I added scripts to the Raspberry Pi operating system that control archiving the Callisto data files and sending them to the e-Callisto data server at FHNW. A table compares the Windows and Linux versions of the Callisto software systems (table 1).

Table 1 ~ Comparison of Windows and Linux versions of Callisto software

Attribute	Callisto	Callisto-Lx	Remarks
Platform	Personal Computer	Raspberry Pi	
Operating system	Windows XP and >	Debian Linux	C-Lx uses Raspbian Jessie Lite
Graphical user interface (GUI)	√		
Command line interface	√	√	
Automatic/Manual control	√	√	
FITS data	√	√	
Log file	√	√ †	† C-Lx uses daemon.log file
Spectral overview (OVS)	Scheduler/Manual	Scheduler/Manual	
Real-time light curve $y(t)$ plot	√		
Real-time spectrum $y(f)$ plot	√		
Real-time spectrum $y(f,t)$ plot	√		
Callisto configuration file	√	√	
Frequency configuration file	√	√	
Scheduler configuration file	√	√	
Raw data mode	√	√	
Calibrated data mode	√		
Software tools	√		Freq. Genie, WWW Genie
Automatic sunrise/sunset		√ ‡	‡ C-Lx requires helper program
NAS data storage	FTP	FTP	Requires FTP client software
Upload to FHNW server	FTP	FTP	Requires FTP client software

All Callisto-Lx connections to other devices in a working system are through standard interfaces and use standard protocols (figure 1). The Callisto-Lx usually is managed from a Secure Shell (SSH) terminal emulator program running on a PC connected via a local area network (LAN) but it also can be managed via a wide area network (WAN) such as the internet. The Callisto instrument is automatically controlled by Callisto-Lx according a predetermined schedule, typically from sunrise to sunset. Spectral power data for a predetermined list of frequencies is collected in the standard astronomical data format as Flexible Image Transport System (FITS) files. Each file typically contains 15 minutes of data in a 200 x 3600 pixel array produced from 4 samples per second for each of 200 frequencies. The data files are copied to the FHNW data server and optionally to a local Network Attached Storage (NAS) server using the File Transfer Protocol (FTP) and then archived to the on-board memory card and optionally to external storage, thus providing automatic data backup.

Figure 1 ~ Basic Callisto-Lx configuration. Callisto-Lx controls the instrument and collects data at regular intervals throughout the scheduled observing period. The data are copied to a worldwide web server (FHNW) and optionally archived to other devices such as a NAS server and USB hard drive or flash drive.



Callisto-Lx does not support real-time display of lightcurves or spectrum plots (as does the Windows-based software); however, near real-time *quicklook* spectrum plots are provided by the FHNW server at {[CallistoData](#)}. Alternately, or in addition, Callisto-Lx can be setup to copy the data to Callisto-Pi {[ReeveCPi](#)}, which produces Portable Network Graphics (*.png) spectrum images from each FITS file within a few minutes. These images can then be viewed on a local PC or forwarded to an internet server for display on the worldwide web.

This project was developed over a period of about 15 months starting in December 2015. The first version of the callisto application software had a fatal bug in that the software would go into an infinite loop on startup when used with the RPi and other ARM platforms (ARM stands for Advanced RISC Machines and is a type of reduced instruction set computer used in the RPi). Considerable time was spent troubleshooting but a fix was not available until January 2017.

As with all my Raspberry Pi projects, I had considerable assistance with the software development portion; see section 8, **Acknowledgements**. I have prepared a complete step-by-step guide for installing the Callisto-Lx hardware and software at {[CLXGuide](#)}. I also can provide an RPi platform with the necessary hardware peripherals and a micro-SD memory card with the Callisto-Lx software image already installed; see section 9, **Order Information**. The remainder of this paper briefly describes the hardware and software and provide some operational details.

2. Hardware

The hardware used in Callisto-Lx consists of three basic components: The Raspberry Pi platform; EIA-232 serial interface circuit compatible with the RPi CMOS logic levels; and a power supply. A USB-serial converter also may

work in place of the EIA-232 level converter, but I made no attempt to implement or test it. An optional hardware real-time clock (RTC) with battery backup also may be equipped and is recommended.

RPi platform: There are many versions of the Raspberry Pi {[RPI](#)}, and I tested the RPi 1 model B+, RPi 2 model B and RPi 3 model B. The RPi model 1 B also may work but I did not test it. In this document I refer to all models simply as *RPi*. I built two prototypes, one with the RTC and one without. The only difference is how the system clock is set as described below. The serial interface and RTC can be shop-built or commercial Raspberry Pi Hats may be used.

The RPi may be connected to a local area network either through a wired Ethernet interface or wireless WiFi interface. The RPi 3 model B has built-in WiFi capability but it will not work if the RPi3 is installed in a metal enclosure. The other models require a wireless USB dongle for WiFi access. The computing hardware system less enclosure is quite compact and costs less than 60 USD (figure 2).

Figure 2 ~ Callisto-Lx hardware includes the Raspberry Pi platform (bottom printed circuit board), a battery backed real-time clock PCB (middle) and a serial interface level converter PCB (top), stacked to form a compact unit. The two peripheral circuit boards are mounted on 11 mm standoffs. Both boards have unneeded prototyping space. The USB and Ethernet interface connectors on the main board are behind at upper-right. A 3-pin right-angle header in the foreground is used in place of the DB-9M connector supplied with the Hat (compare to next figure).



Serial interface: The Callisto-Lx serial interface uses signals available on the RPi general-purpose input/output (GPIO) connector. The GPIO is based on 3.3 V CMOS logic and is not directly compatible with the EIA-232 interface required by the Callisto instrument. Therefore, it is necessary to use a voltage level converter transceiver. The interface described here is the *Serial Pi Plus* available at {[Serial](#)} (figure 3), which is a commercial Raspberry Pi Hat based on the MAX3232 transceiver integrated circuit. Cost is about 9 USD. One advantage of this particular product is the signals are brought out to both a DB-9M connector and separate header pads.

Figure 3 ~ *Serial Pi Plus* Hat used in Callisto-Lx includes a MAX3232 transceiver, a few charge pump and decoupling capacitors, a DB-9M connector and a 40-pin GPIO connector. All surface mount devices are preinstalled by the factory. The DB-9M connector is shown soldered to the PCB in this image but is removed to reduce space when the RPi assembly is installed in an enclosure. Image source: AB Electronics.

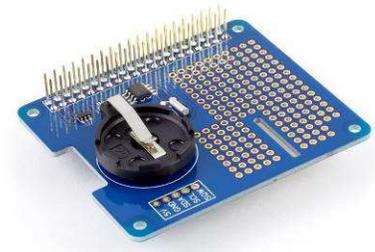


Power Supply: In its basic configuration, the Callisto-Lx requires a 5 Vdc, 1 A (5 W) power supply but more current may be needed if the platform is used with a display. For example, a touchscreen display could be used that derives its power from the same bus as the RPi, in which case a 5 V power supply for the system should be rated at least 2 A (10 W). I installed Callisto-Lx in an enclosure. For better integration with the Callisto instrument, which requires nominal 12 Vdc for power, I used a 12 V to 5 V, 10 W step-down dc-dc converter and installed it in the enclosure with the RPi. This allows both the instrument and Callisto-Lx to be powered from the station 12 Vdc power supply.

Real-Time Clock: The Callisto instrument does not require internet (WAN) access to operate; however, data are time-stamped by the controlling computer and the basic RPi platform requires internet access to keep time. As originally supplied the RPi is not equipped with a battery backed hardware real-time clock (RTC). It keeps time by counting its on-board oscillator and implementing a software time-of-day clock (*system clock*). The RPi uses the Network Time Protocol (NTP) to set and regularly update the system clock.

By default, the RPi accesses NTP time servers on the worldwide web, thus requiring internet access. It may be reconfigured to access a local NTP server such as the GpsNtp-Pi [{ReeveGNPi}](#) through a LAN connection. However, if the LAN and WAN connections both fail or are not available throughout the observing period, the time-of-day system clock cannot be set or updated. Therefore, it is recommended to use a hardware real-time clock and associated battery. Although the hardware clock can be manually set to the correct time, it does drift. The drift is compensated by NTP when the RPi has access to an NTP server. Commercial RTC Hats are available; I used the *RTC Pi Plus* available at [{RTC}](#) (figure 4), which is based on the Maxim DS1307 RTC integrated circuit. Cost is about 9 USD + CR2032 battery. Although the DS1307 is capable of being fine-tuned to improve its time accuracy, the Linux drivers presently available do not support that feature.

Figure 4 ~ *RTC Pi Plus* Hat used in Callisto-Lx includes a DS1307 integrated circuit, an I²C bus interface IC, a few passive components, battery holder and a 40-pin GPIO connector. All surface mount devices are preinstalled by the factory. The battery is supplied by the user. Image source: AB Electronics.



Enclosure: I installed the RPi board with the two Hats in an extruded aluminum enclosure (figure 5) along with a small printed circuit board that contains the step-down dc-dc converter (figure 6). Power control and indication are installed on the back panel and the EIA-232 interface connector is installed on the front panel (figure 7). The external appearance is identical to Callisto-Pi and GpsNtp-Pi except for the EIA-232 interface connector.

Figure 5 ~ Anodized extruded aluminum enclosure houses the RPi assembly and a converter power supply. The enclosure shown here with matching rubber bumpers on the end-plates is Box Enclosures BEX-series, and its dimensions are 108 W x 160 L x 50 H mm. The Callisto-Lx footprint is almost identical to the Callisto instrument.



Figure 6 ~ Internal converter power supply uses the CPS-1-M1 printed circuit board (PCB). The power supply uses a 10 W step-down dc-dc converter (rectangular black block at upper-right) along with input and output filter components, polarity guard diode and overcurrent and overvoltage protection. The input and output are connectorized so the power supply can be easily installed and removed. The cable is terminated in a micro-USB connector, which is used by the RPi for power input. The Callisto-Pi and GpsNtp-Pi use the same power supply.

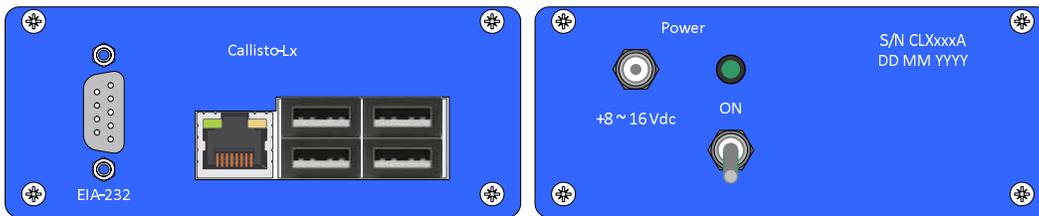
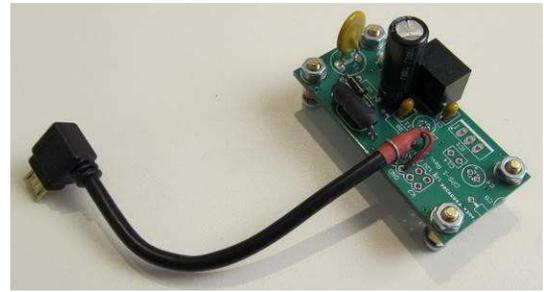


Figure 7 ~ Callisto-Lx front and back panels. Left: The front panel is cutout for flush-mounting the RPi USB and Ethernet interfaces and also includes a DB-9M connector for the serial interface, which is wired to the RPi through a wire-mount socket housing. Right: The power input jack, on-off switch and power indicating LED are on the back panel. The rear panel also is wired to the internal power supply through a wire-mount socket housing.

Display: The simplest way to operate Callisto-Lx is in the so-called “headless” mode in which no directly connected keyboard, mouse or display is needed. Alternately, an HDMI monitor or TV with an HDMI port may be connected to the RPi HDMI port and a keyboard and mouse may be connected through a wireless or wired connection to a USB port on the RPi. Another alternative uses a touchscreen, which provides a graphical user interface (GUI) display and the equivalent of a keyboard and mouse all-in-one. I tested Callisto-Lx with a small touchscreen display {[Display](#)} but found the “headless” configuration far more convenient. Because my Callisto-Lx configurations use an enclosure, the RPi HDMI connector is not available for external connection to a display anyway.

3. Software

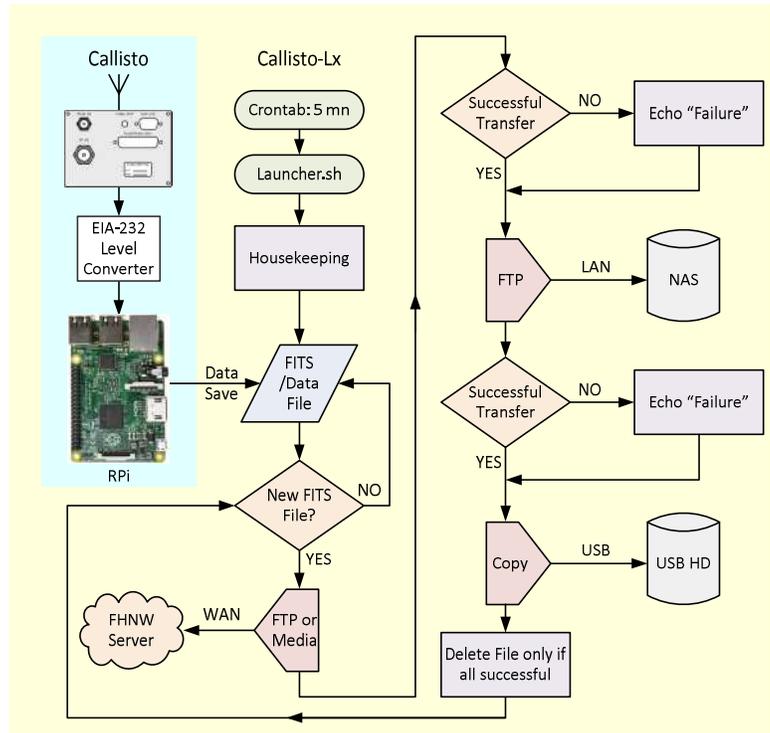
The main software components in Callisto-Lx are the Linux operating system and the Callisto software application program. A number of software tools and applications also are installed to support testing and operating the hardware and software. As data files are produced by the Callisto software application, a launcher shell (figure 8) runs in the background to manage these files including sending them to the e-Callisto data server and archiving to a Network Attached Storage (NAS) server and a USB hard drive or USB flash drive.

Generally, all software installations are configured from a Windows PC using PuTTY or Tera Term terminal emulator as the primary command line interface. The command line interface provides a secure shell (SSH) connection to the RPi through the LAN. The PC also uses WinSCP as the RPi file explorer (WinSCP stands for Windows Secure Copy Protocol). The advantage of WinSCP is that directory structure is shown in a more obvious way than is possible with a command line interface. Also, it allows files to be conveniently dragged and dropped between the Windows PC and the RPi.

Operating system: The latest RPi operating system at the time of writing is the Raspbian *Jessie* implementation of Debian Linux [{Raspbian}](#). I installed the *Jessie Lite* version because it uses much less space on the memory card than the full version with no loss of features relevant to Callisto-Lx. The operating system software is easily downloaded and installed on the micro-SD memory card used in the RPi. Before the application software and tools are installed, the RPi operating system is brought up-to-date. Depending on the amount of updating required, this may require several hours (one of my systems ran overnight to upgrade).

Figure 8 ~ Launcher shell flow chart. After a data file is produced by the Callisto application software, the launcher shell manages its disposition, which includes copying to FTP servers and to an optional local USB drive.

The serial interface needs to be configured in software. If installed, the hardware real-time clock uses the RPi's built-in I²C bus and requires installation of additional software drivers and configuration for operation. A separate terminal emulator program called *Minicom* is installed on the RPi for testing the serial interface connection with the Callisto instrument. After the interfaces are setup and tested, the application software is installed and configured. Software installation and configuration of the peripherals requires at most an hour and involves following a step-by-step guide, which includes tests at each major step [{CLXGuide}](#).



Callisto application: The application software is provided as *.tar.gz file packages that are first extracted and then compiled on the RPi to build the executable binary files. The result is a set of directories and the main callisto program. The program requires the function library for input/output of FITS formatted data files, which is installed using the Advanced Packaging Tool (APT), a normal Linux installation process. I found the installation to be relatively easy and trouble-free, requiring only a few minutes. The callisto application is installed as a *service*, which can be started, stopped and restarted as needed.

4. Callisto Configuration Files

Callisto-Lx uses configuration files that are very similar to the Windows versions: callisto.cfg, frqxxxx.cfg and scheduler.cfg. For the most part, the configuration rules described in the Windows version of the Callisto Software Setup Guide ([{CallistoSSG}](#)) apply to the Linux versions, but the Linux versions have some limitations, which are described in [{CLXGuide}](#).

The callisto.cfg establishes the basic operational parameters for the Callisto instrument including serial interface port, station coordinates, frequency configuration filename, data file paths, spectrometer gain and observation interval. The main differences between the Windows and Linux callisto.cfg files are the serial port nomenclature and an extra parameter in the Linux version that specifies a TCP port number of a command server built into the callisto application. The command server allows remote control from anywhere that can access the Callisto-Lx.

The frequency configuration file frqxxxxx.cfg contains a list of frequencies and the sweep interval for the instrument. The Linux version of the software can observe up to 400 channels in the range 45 to 870 MHz with up to 62.5 kHz resolution (a total of 13 200 channels are available). The Callisto instrument supports a sweep rate of 800 channels per second (1000 channels per second if the instrument is externally clocked) but is typically setup to repetitively sweep through 200 channels in 0.25 seconds. Unlike the Windows version, Callisto-Lx does not support integrating (averaging) of specific frequencies in the frequency configuration file – any such entry in the configuration file is simply ignored. The basic integration time for the Callisto is 1 ms.

The scheduler configuration file scheduler.cfg lists the observation start and stop times and also sets the time for an optional spectral overview (OVS). The OVS usually is taken a couple hours before sunrise and is a tabulation of relative spectral power in each channel throughout the Callisto's entire frequency range (all 13 200 channels). The OVS is useful for RFI surveys and for preparing a frequency configuration file with frequencies having the least spectral power (interference). If a scheduler file does not exist, the callisto software reverts to a manual mode in which the instrument must be started and stopped by the user.

Figure 9 ~ Callisto-Lx (top) and Callisto instrument (bottom) connected through an EIA-232 cable. In this image the Callisto-Lx is equipped with a USB flash drive for data archiving. A USB hard drive also may be used. An Ethernet cable is not yet connected but will be in an operating system. A power splitter cable on the back allows both Callisto-Lx and the instrument to be powered by one connection to the station 12 Vdc power system.



5. Daily Operation

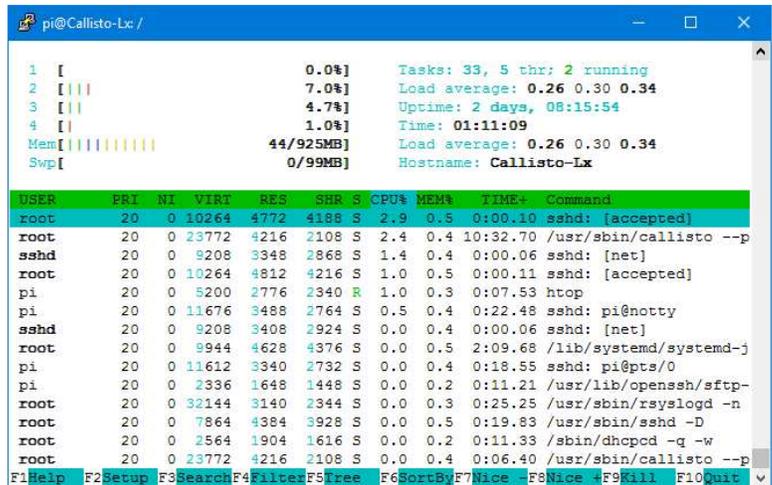
After the Callisto-Lx has been completed and tested, the system is ready for operation (figure 9). For a new setup, it is first necessary to load the default frequency configuration into the instrument EEPROM through the command line interface. The Callisto application is then run as a service under Linux, which allows the

instrument to be manually or automatically started and stopped. Manual operation is through the command server previously mentioned. Automatic operation is through the scheduler configuration file, which is used if present.

The software used in Callisto-Lx is capable of automatically generating a new scheduler file each day based on the station coordinates, but I have not yet tested this feature. Such a feature would be handy for users at latitudes where the daylight hours vary considerably throughout the year. In my case, I manually update the scheduler file as needed every month or two months depending on the time of year.

One of the handy software tools that I installed on Callisto-Lx that is unrelated to its daily operation is *htop*, a resource viewer similar in operation to the familiar Windows *Task Manager*. The program name comes from the initials of its developers. It shows that Callisto-Lx is a fairly light-duty application in terms of CPU and memory usage (figure 10).

Figure 10 ~ Linux resource viewer *htop* shows usage of various processes. This screenshot shows the light load that Callisto-Lx imposes on the CPU is only a few percent and memory usage is only 44 MB out of approximately 1 GB.



6. Conclusions

Callisto-Lx is based on the Raspberry Pi and Linux operating system and is used to automatically and manually control the Callisto instrument. The hardware costs about 60 USD and the software is free. As with the Windows version of the Callisto software, once setup the Callisto-Lx operates with little or no user intervention. It has proven to be predictable, reliable and easy to use.

7. References and Web Links

- {CallistoData} <http://soleil.i4ds.ch/solarradio/callistoQuicklooks/>
- {CallistoSSG} <http://www.reeve.com/Documents/CALLISTO/CALLISTOSoftwareSetup.pdf>
- {CLXGuide} http://www.reeve.com/Documents/Articles%20Papers/Reeve_Callisto-Lx_Setup.pdf
- {Display} <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>
- {e-Callisto} <http://www.e-callisto.org/>
- {ISWI} <http://www.iswi-secretariat.org/>
- {Raspbian} <https://www.raspberrypi.org/downloads/raspbian/>
- {ReeveCPi} http://www.reeve.com/Documents/CALLISTO/Reeve_Callisto-Pi_Setup.pdf
- {ReeveGNPi} http://www.reeve.com/Documents/Articles%20Papers/Reeve_GpsNtp-Pi_Setup.pdf

{RPI} <http://www.raspberrypi.org/>
{RTC} <https://www.abelectronics.co.uk/p/52/RTC-Pi-Plus>
{Serial} <https://www.abelectronics.co.uk/p/51/Serial-Pi-Plus>
{Unix} http://www.metsahovi.fi/callisto/e-Callisto_for_Unix/

8. Acknowledgements

I am indebted to Juha Aatrokoski for his assistance with operational details of the Callisto Unix software and to Whitham Reeve II for his assistance implementing the application software on Linux and developing the launcher shell that manages the Callisto data files.

9. Order Information

Ordering: A Callisto-Lx memory card and Raspberry Pi platform with peripheral boards may be ordered by sending an email inquiry to the address right; be sure to put something meaningful in the email Subject line or else the email will be automatically deleted by the server.

OrderInfo@reeve.com

Document information

Author: Whitham D. Reeve

Copyright: © 2017 W. Reeve

Revision: 0.0 (Original draft started, 6 Mar 2017)

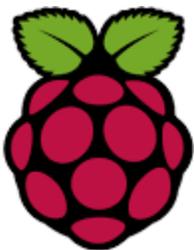
0.1 (Completed 1st draft including all images, 13 Mar 2017)

0.2 (Minor edits and corrected max channels, 21 Mar 2017)

0.3 (Minor edits prior to distribution, 01 Apr 2017)

Word count: 3606

File size: 13892608



Logos are property of their respective owners