

# PLUS 1 MINI USER GUIDE

## INDEX

- 1 Introduction
- 2 Installation & Quick Start
- 3 The Cartridge Slots
  - 3.1 Cartridge basics
  - 3.2 Compatibility
  - 3.3 Sideways RAM
  - 3.4 Managing Sideways Banks
- 4 SD Cards & MMFS
  - 4.1 Choosing and Preparing an SD Card
  - 4.2 MMFS Basics
  - 4.3 Managing Disk Images
  - 4.4 Loading & Saving programs and data
- 5 The TUBE Interface & PiTube
  - 5.1 Preparing the PiTube Software
  - 5.2 Installing the Pi
  - 5.3 Using the PiTube
- 6 The Joystick Port
  - 6.1 Hardware
  - 6.2 Joysticks and Games
  - 6.3 Software Development
- 7 Compatibility Switches & Commands
- 8 Technical Information
  - 8.1 I/O Registers
  - 8.2 Pinouts
- 9 Acknowledgements

# 1. INTRODUCTION

The Plus 1 Mini is a multi-function expansion unit for the Electron, with the following capabilities:

- Connects directly to the Electron, no extra hardware required
- Load and Save from a standard SD card
- Up to 128KB of sideways RAM
- Two Plus 1 compatible cartridge slots
- PiTube co-processor interface
- Commodore/Atari type 9-pin joystick interface
- 3D printed enclosure

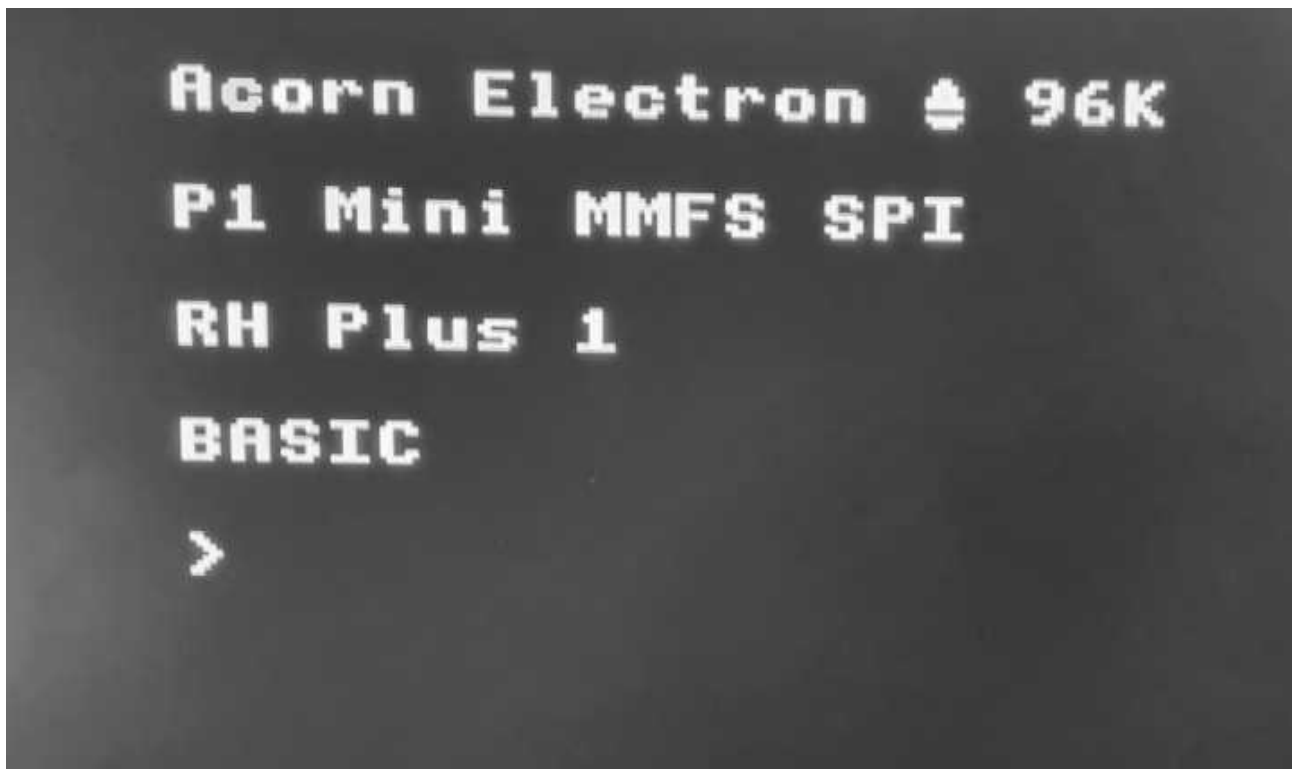
The purpose of this guide is to demonstrate how to safely install the Plus 1 Mini and make full use of all its features.

## 2. INSTALLATION & QUICK START

Installing the Plus 1 Mini takes only a few steps

- Make sure the Electron is powered off.
- Clean the Electron's expansion connector using a pencil eraser followed by an application of isopropyl alcohol or contact cleaner.
- Connect the P1M, it will only fit one way. Make sure the large mounting holes on the P1M line up with the holes on the underside of the Electron.
- Use the two included 6x30mm screws to secure the P1M and Electron together. **Do not over-tighten the screws as this may damage the Plus 1 Mini's case.**

Once these steps have been completed switch on the Electron. You should see a startup-screen like this:



```
Acorn Electron 96K
P1 Mini MMFS SPI
RH Plus 1
BASIC
>
```

If you do not see this screen switch off the Electron, remove the P1M and clean the edge connector again.

## 3. THE CARTRIDGE SLOTS

On top of the Plus 1 Mini are two cartridge slots which are physically and electrically compatible with the slots on Acorn's original Plus 1 interface.

Each slot can accept either a 'short' or 'long' cartridge, although the rear slot will be limited to a short cartridge if a Raspberry Pi co-processor is fitted.

The Electron must be switched off before inserting or removing a cartridge or damage will occur.

*Please note: the Plus 1 Mini cartridge sockets are around 2mm wider than those on the original Plus 1. This is not normally a problem, but some uncased cartridges or those with narrower edge connectors can seat improperly in the socket. Removing and re-inserting the cartridge will usually cure any alignment problem.*

### 3.1 CARTRIDGE BASICS

Acorn designed the Plus 1 cartridge system as an easy way to expand the Electron with new hardware.

Each slot is capable of supporting up to 32KB of memory, split over two 16KB banks. This memory can be ROM, RAM, non-volatile storage, etc, or a mix of these.

Any memory on a cartridge inserted into the frontmost slot will show up as banks 2/3 in the Electron's sideways memory map. The rear slot occupies banks 0/1.

By default the Plus 1 Mini works like the original Plus 1, reserving banks 0/1/2/3 for cartridges. However, it is possible to temporarily disable the cartridge slots in order to use those banks for 64KB of extra sideways memory. Please see the compatibility section for details.

Additionally, cartridges can optionally contain 16KB of memory that maps into sideways bank 13. Only cartridge can make use of this facility at a time, so inserting two cartridges which use bank 13 will cause a conflict and is not recommended.

As well as memory the cartridges have access to an input/output system using memory locations in the &FCxx and &FDxx areas (known as FC page and FD page) that enables the Electron to communicate with I/O hardware contained in cartridges.

### 3.2 COMPATIBILITY

Acorn's cartridge system is, by necessity for low-cost 1980s era hardware, quite simplistic. There is no mechanism for avoiding resource conflicts between the two cartridges, or between a cartridge and the host Electron and expansion unit.

In practice this means attempting to use two cartridges utilising the same addresses in FC or FD pages for I/O will cause both cartridges to misbehave. This issue is rare but no work-around exists other than not using those particular cartridges at the same time!

Resource conflicts can also occur between a cartridge and the Plus 1 Mini's internal hardware. For example, the SD card interface uses addresses &FC80 and &FC81. A cartridge using those addresses will cause a conflict and unpredictable behaviour.

The Plus 1 Mini provides options for dealing with such situations. These are:

- \* Disable SD card
- \* Disable cartridge slots
- \* Disable Tube
- \* Safe mode

On the rear of the Plus 1 Mini unit are four DIP switches which can be used to select between 'active' or 'software control' for each of these options. See section 7 for more details.

The Plus 1 Mini has been carefully designed to minimise conflicts, but use of these options may occasionally be required.

### 3.3 SIDEWAYS MEMORY

To fully understand the Plus 1 Mini's features it is useful to have a good grasp of how the Electron's 'sideways memory' system functions.

The 6502 microprocessor used by the Electron is capable of addressing up to 64KB at one time. On the Electron the lower half of this address space (addresses &0000 - &7FFF) is occupied by the computer's standard 32KB of RAM, which is used to store user programs and screen data.

The upper 16KB contains the operating system (MOS in Acorn nomenclature), overlaid by I/O space at &FCxx and &FDxx, and the ULA chip registers at &FExx.

That leaves the area between these two (&8000 - &BFFF), which is where the interesting things happen. This area is known as *sideways* space.

At boot up the sideways space contains the 16KB of ROM used to store the Electron's BBC BASIC language interpreter. But Acorn provided a mechanism for switching out the BASIC ROM and replacing it with something else.

Note that this switching is transparent to the user, so for example the Plus 1 Mini's SD card can be used by BASIC programs. The MOS will intelligently switch between the BASIC and SD card ROMs as needed.

Acorn's MOS supports up to sixteen 'sideways banks', each of which can be paged into the sideways space. In theory this would permit 16 x 16KB sideways banks, or 256KB of memory, to be used. However, banks 10 and 11 contain the BASIC ROM and pages 8 and 9 are used by the Electron hardware for, of all things, interfacing with the keyboard.

That leaves 12 banks (192KB) available for other uses. Acorn's Plus 1 reserves banks 0-3 for memory contained in cartridges and the Plus 1 Mini duplicates this behaviour by default.

Also, the Plus 1 contains a ROM chip which appears in bank 12. This is used to control the Plus 1's printer port function. The other banks (4-7, 13-15) are left unused.

Because it contains so much more hardware the Plus 1 Mini makes fuller use of sideways slots. The map looks like this:

- 15 - (ROM) Plus 1 utilities
- 14 - (ROM) SD card support (MMFS)
- 13 - (EMPTY) Reserved for cartridge ROMs
- 12 - (ROM) Joystick and compatibility commands
- 10, 11 - BASIC
- 08, 09 - Keyboard



- 07 - (RAM) used by MMFS
- 04 – 06 - (RAM) 64KB sideways RAM
- 02, 03 - Front cartridge slot
- 00, 01 - Rear cartridge slot

This map is carefully chosen to contain all the necessary ROMs and 64KB of sideways RAM, while still enabling the cartridge slots to function.

However, an alternate map is available which disables the cartridge slots and uses banks 0 – 3 for an extra 64KB of sideways ROM. See the DIP switch section for more details.

### 3.4 MANAGING SIDEWAYS BANKS

The Plus 1 Mini provides various commands for managing sideways memory banks. These commands are provided by the ‘RH Plus 1’ ROM which occupies bank 15.

#### **\*SRLOAD filename bank**

Loads a ROM image from disk or SD card into the specified sideways bank, eg, \*SRLOAD “testrom” 7 will load the file testrom into bank 7.

#### **\*SRSAVE filename bank**

Saves the contents of a sideways bank to disk or card.

#### **\*SRWIPE bank**

Erases the contents of the specified bank.

#### **\*INSERT bank**

Causes the operating system to scan the specified bank for a valid sideways ROM image and enables the ROM. Some ROMs may require a reboot to function.

**\*UNPLUG slot**

Disables a ROM in the specified sideways bank. A reboot may be required to fully disable the ROM.

**\*ROMS**

Lists the contents of all sideways banks. RAM banks will be shown by an inverted 'R'.

## 4. SD CARDS AND MMFS

The Plus 1 Mini provides the capability to use an SD memory card to store programs and data. To do this it uses a ROM known as MMFS, which enables the vast majority of Electron software to work with a memory card by emulating a floppy drive system.

One SD card can contain over 500 ‘disk images’ which when accessed appear to the Electron as a floppy disk. MMFS provides additional commands to manage disk images.

The following sections describe how to use SD cards and disk images with MMFS. It is an overview rather than an exhaustive guide as MMFS has its own documentation at:

<https://github.com/hoglet67/MMFS/wiki>

### 4.1 CHOOSING AND PREPARING AN SD CARD

MMFS requires an SD of no more than 8GB capacity, formatted in with the FAT32 filesystem and which supports the legacy SPI protocol.

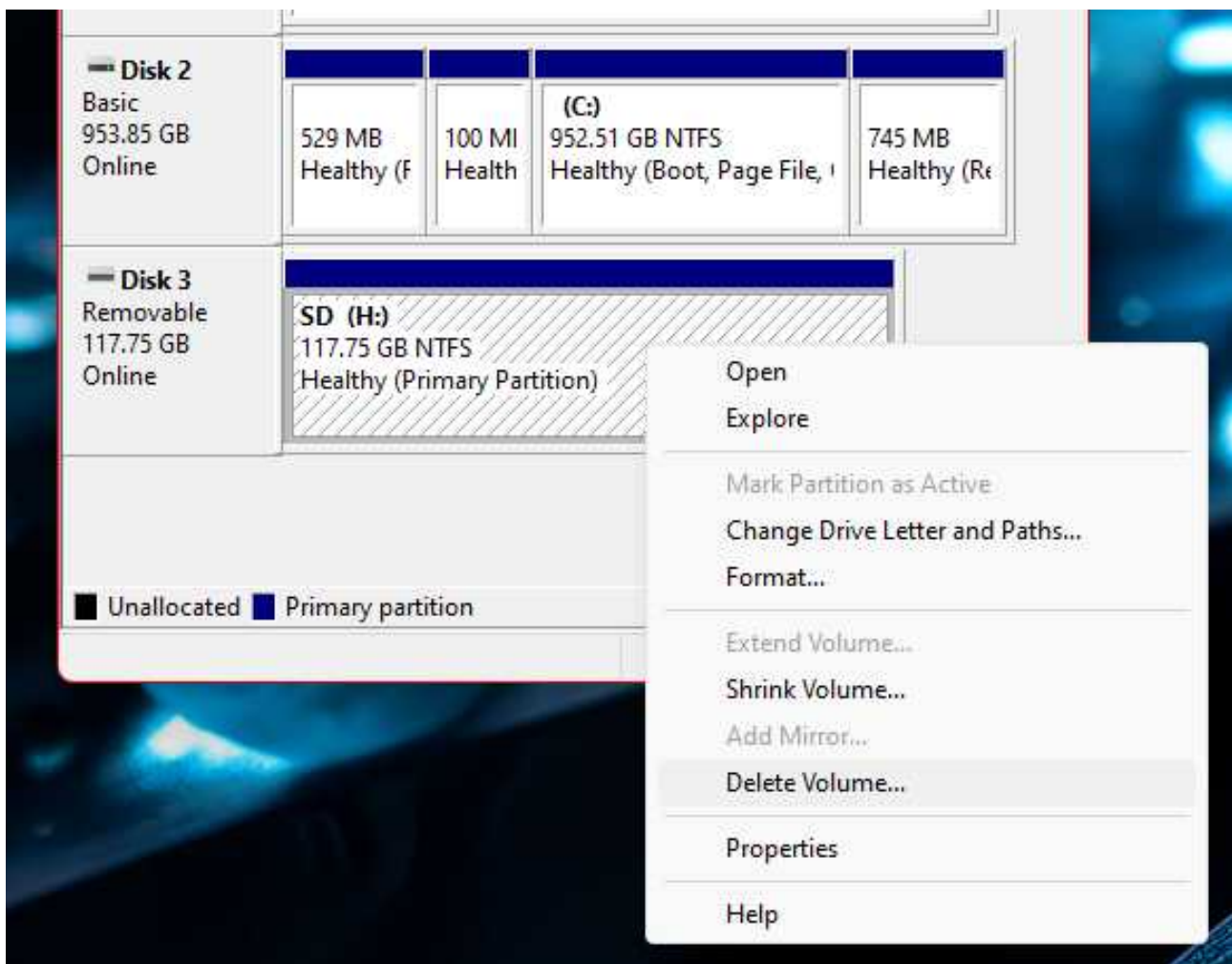
Modern SD cards from major vendors still retain SPI compatibility but some cheaper off-brand cards may not. For this reason it is strongly recommended to only use SD cards from leading brands such as SanDisk, Kingston, Samsung and Kioxia. These also use better quality flash memory and are less likely to suffer data loss.

MMFS does not officially support cards of greater than 8GB capacity. However, larger cards will often work if partitioned to

8GB or less and formatted as FAT32. An overview of how to do this is presented below.

These instructions are for Windows 10/11 only. For Mac or Linux please consult your operating system documentation on how to partition and format removable drives.

Right click on the start menu and select 'Disk Management'. When the disk management window appears, find the SD card (it should be marked 'removable') and right click in the area below the blue bar.



Confirm this action and the partition will be deleted. Next, right click in the area below the bar, which should now be black, and select 'New Simple Volume'.

Follow the prompts, choosing a volume size of 8000MB, FAT32 as the filesystem and default allocation size. The SD card is now ready for use.

## 4.2 MMFS BASICS

*Please note, this information relates to MMFS 1, which the Plus 1 Mini uses by default. MMFS 2 is not covered here.*

MMFS works by storing a number of 'disk images' inside a single file on the SD card. This file is named 'beeb.mmb' and can hold up to 512 images.

A pre-prepared beeb.mmb files containing a number of games and utilities can be found here:

<http://ramtop-retro.uk/Plus-1-Mini.html>

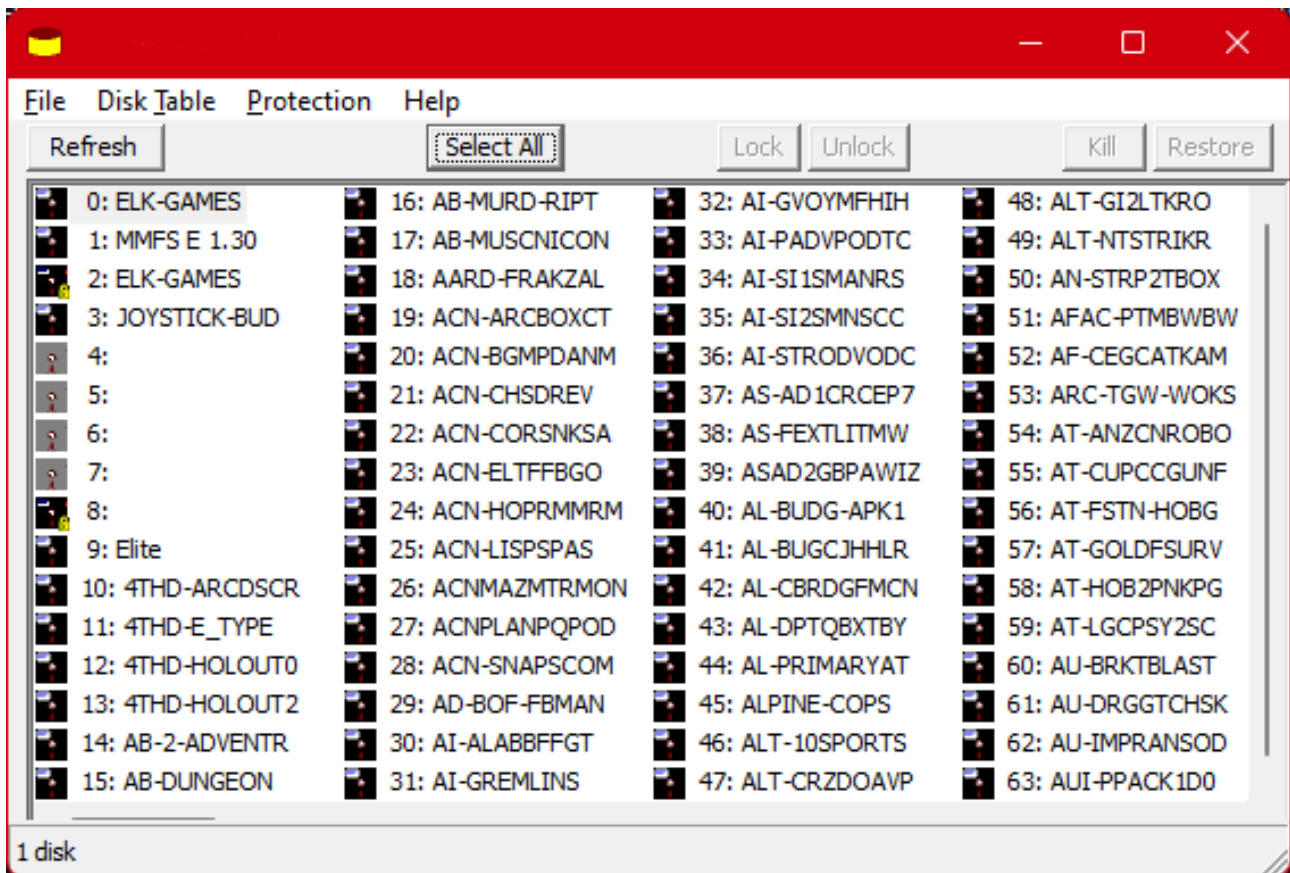
To use this file simply copy it to a prepared SD card and insert the card into the Plus 1 Mini's SD slot. This slot is of the push-push type, so the card can be ejected by gently pushing into the slot.

If you wish to create your own beeb.mmb or modify the prepared one, you will need a utility called MMBImager. Download it here:

<https://github.com/dandelion-labs/MMBImager>

Start MMBImager and if you wish to create a new beeb.mmb file select 'File/New Image' then click OK, select your SD card and enter 'beeb.mmb' as the file name.

Once the file has been created, or if you already have a pre-prepared beeb.mmb file, click 'File/Open Image' and select the file.



The Plus 1 Mini uses Single Sided Disk (.SSD) image files, to install these into the beeb.mmb file just drag the SSD file and drop it into a slot in the MMBImager window.

When you're done just click 'File/Close Image'.

### 4.3 MANAGING DISK IMAGES

Full documentation for MMFS commands can be found at the link below, however this section will give an overview of the most used commands.

<https://github.com/hoglet67/MMFS/wiki>

**\*DCAT**

Shows a list of disk images contained within the beeb.mmb file.

**\*DIN number**

Selects the specified disk image and makes it available for use.  
Eg, \*DIN 32 will load the image in slot 32.

**\*CAT**

Display the contents of the currently selected disk image.

**\*DRIVE number**

Selects the virtual floppy drive to use, 0-4. Each virtual floppy can contain a different image, so

\*DRIVE 0

\*DIN 10

\*DRIVE 1

\*DIN 20

will load select the image in slot 10 for drive 0 and slot 20 for drive 1.

**\*DBOOT number**

Boots the disk image in the specified slot.

**\*DDRIVE**

Shows which images are selected for each virtual drive.

**\*DOP P**

Write-protects the currently selected disk image in the specified virtual floppy.

### **\*DOP U**

Set the currently selected image to read-write.

### **\*DOP K**

Marks the current disk image as unformatted.

### **\*DOP R**

Mark the current disk image as formatted.

### **\*TITLE string**

Give the current disk image a title. This will be shown in the disk image list when using \*DCAT.

## **4.4 LOADING & SAVING PROGRAMS AND DATA**

The commands used for loading and saving are generally the same as used for floppy disks. They are summarised below.

### **SAVE “name”**

Saves the current BASIC program with the specified name.

### **LOAD “name”**

Loads the specified BASIC program in to memory.

### **\*SRSAVE “name” bank**

Saves the contents of the specified sideways bank. Note the bank number must be given in decimal, 0-15.

### **\*SRLOAD “name” bank**

Loads the specified file in to a sideways bank.

### **\*SRWIPE bank**

Clears the specified sideways bank.



# 5. THE TUBE INTERFACE AND PITUBE

The Tube co-processor interface originated on the BBC Micro as a way of connecting CPUs with greater performance or additional features compared to the beeb's 2MHz 6502.

The Electron does not natively support the Tube, however the Elk's similarity to the Beeb makes it fairly easy to add. Traditionally this is done via a cartridge which contains the physical logic required to provide a Tube port and a ROM with the necessary code to enable the operating system to support a co-pro.

The co-processor units built in the 1980s were expensive and are now quite rare. This led to the development of the PiTube Direct software, which enables a Raspberry Pi to connect to the Tube port and emulate a series of co-processors, including 6502, ARM, Z80, 6809, 68000, etc.

A 40-pin header on the Plus 1 Mini PCB provides the ability to connect a Pi Zero or Pi Zero 2 to act as a PiTube co-processor. The Pi must be fitted with a 40-pin male header.

## 5.1 PREPARING THE PITUBE SOFTWARE

Download the PiTube software from GitHub:

<https://github.com/hoglet67/PiTubeDirect/releases>

and extract the archive to a MicroSD card formatted as FAT32. Insert the card into the Pi.

## 5.2 INSTALLING THE PI

In order to connect the Pi it should have header pins soldered to the 40-pin GPIO header. Just push the Pi's header pins into the 40-pin socket on the Plus 1 Mini. The Pi should be mounted upside-down with the component side of the PCB above the DIP switch block on the Plus 1 Mini.



Please note, with the Pi fitted the Plus 1 Mini's rear cartridge slot will not be able to accommodate cased 'long' cartridges. Uncased ones may still fit, but take care nothing on the cartridge shorts the exposed Pi PCB.

## 5.3 USING THE PITUBE

When you boot for the first time after installing the PiTube nothing will change. This is because the Pi takes a few seconds to boot up, so the PiTube software is not running yet.

After a moment press Ctrl-Break to reboot and the screen should look like this:

Any programs you run are now executing on the Pi's 6502 co-processor emulation. One side effect of this is the co-processor emulation has a flat memory map of 64K, sideways memory on the Electron side cannot be accessed. It can, however, still be used to store ROM images. These will work as normal.

Any programs that use the operating system for I/O should work on the 6502 co-processor, but anything that writes to the hardware directly (such as games) will not work.

If you wish to disable the co-processor across reboots without removing the Pi used the DIP switch on the rear of the Plus 1 Mini. Or it can be temporarily disabled by typing \*TUBE OFF

Much more in-depth information can be found in the PiTube Direct wiki: <https://github.com/hoglet67/PiTubeDirect/wiki>

## 6. THE JOYSTICK PORT

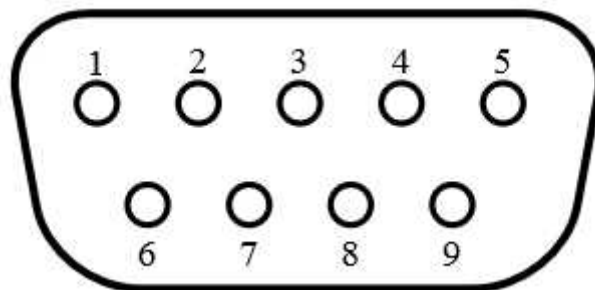
The Electron does not feature any joystick ports as standard, so third party suppliers introduced their own interfaces and multiple systems for connecting joysticks sprung up over time. Acorn officially supported joysticks with the Plus 1 expansion unit but these were analogue type joysticks and were not widely used.

Among the most popular joystick interfaces were those from Slogger and First Byte Software, both used 9-pin Atari/Commodore type digital joysticks. Irritatingly, while these systems were broadly similar they are not compatible.

The Plus 1 Mini's joystick port supports Atari/CBM 9-pin joysticks, including two independent fire buttons. Both Slogger and First Byte interfaces are implemented at hardware level, as is a compatibility system which makes the digital joystick appear as a Plus 1 type analogue stick.

### 6.1 HARDWARE

The joystick port is a 9-pin D-type connector with the following pinout:



1. Up
2. Down
3. Left

4. Right
5. NC
6. Fire 1
7. +5v
8. Ground
9. Fire 2

All input pins have current limiting resistors and it is therefore safe to short them to ground. No more than 50ma should be drawn from the +5v pin.

Only Atari/Commodore type joysticks should be attached. Spectrum +2 sticks with a grey plug should not be connected, nor should joypads, unless they are simple ones which conform to the Atari type standard.

The joystick inputs can be used as general purpose input pins, however do take care not to expose them to voltages above 5v.

## 6.2 JOYSTICKS AND GAMES

As the Electron lacks joystick ports only a minority of games offer joystick support. Some can use the Plus 1 joystick system (mainly Acornsoft titles) and these should work with the Plus 1 Mini joystick without issues. Others support one of the several incompatible third-party joystick systems.

The Plus 1 Mini features four methods of joystick support:

1. Acorn Plus 1 joystick port
2. First Byte interface
3. Slogger interface
4. Keyboard emulation

Acorn's Plus 1 expansion unit provided a single analogue type joystick port, compatible with that found on the BBC micro. If a game supports the Plus 1 joystick port no configuration is generally necessary to use it with the Plus 1 Mini. Select the joystick option in the game (some games will prompt you to press fire to detect the joystick) and it will work.

The First Byte and Slogger interfaces are devices that added a digital joystick port to the Electron. Games may prompt you to enter the I/O address of the joystick interface, which on the Plus 1 Mini is &FCC0 for the First Byte and &FCD0 for the Slogger. Games may also ask for a 'code' or 'polarity' to be entered, being either 0 or 1 – this determines if the joystick signals are active high or low at a hardware level. Try both, one should work correctly.

Finally the Plus 1 Mini offers a keyboard emulation system. This detects joystick movements and generates a pseudo key-press so games with only keyboard control can use the joystick. Many games will work with this system, but not all.

After plugging in the joystick for the first time we recommend you run the joystick test utility to check it is working correctly. This can be done with the command:

### **\*JOYSTICK 2 TEST**

(Which can be abbreviated to **\*J. 2 T.** for brevity)

Configuring the keyboard emulation system can be done by typing:

### **\*JOYSTICK 2 SETUP**

(or **\*J. 2 S.**)

You will be prompted to enter keys for UP, DOWN, LEFT, RIGHT and FIRE, and also an 'address'. This is where the small machine code program that enables keyboard emulation resides in memory. The default address is &150 – you can press enter to accept the default.

After this you can start your game. Note that you must boot the game by entering a command rather than pressing SHIFT-BREAK or the keyboard emulation will not work.

For games stored on SD card this is usually done using the

**\*DBOOT** command, followed by the 'slot' number of the game.

If you have downloaded the games collection from the Plus 1 Mini web site, you can boot the main menu using

**\*DBOOT 0**

### 6.3 SOFTWARE DEVELOPMENT

If you wish to create your own joystick-aware software, the first consideration is which joystick standard to use. The Acorn Plus 1 system is generally recommended as the Plus 1 has a larger installed base of compatible hardware than the third-party systems and it is the only one to support two fire buttons. Also, provided you use the operating systems calls it retains compatibility with the BBC Micro joystick port.

However, the Slogger and First Byte systems offer a small performance advantage as only one read operation is required to determine the joystick status.

When using the Plus 1 joystick system the Plus 1 Mini's joystick can be read in BASIC by using the ADVAL function.

For example,

### **PRINT ADVAL(1)**

will return 32768 for no joystick movement, 0 for right and 65280 for left.

### **PRINT ADVAL(2)**

will similarly return 32768 for no movement, 0 for down and 65280 for up.

For compatibility with systems using analogue joysticks it is best to not specifically depend on these numbers, but check for something close, ie, below 1000 for down or right, above 60000 for up or left. Some fine-tuning may be needed for best compatibility over a range of hardware.

Reading the fire buttons is also done using ADVAL.

### **PRINT ADVAL(0) AND 3**

This will return 0 for no press, 1 for first button pressed, 2 for second button and 3 for both. A fuller explanation of these commands can be found in the BBC Micro user manual page 202.

The Slogger and First Byte interfaces are mapped into addresses &FCC0 and &FCD0 respectively, with the joystick signals as below (from LSB to MSB):

0 - Up



- 1 - Down
- 2 - Left
- 3 - Right
- 4 - Fire
- 5 - 1
- 6 - 1
- 7 - 1

These signals are active low, so for an example reading `&FCC0` with no joystick movement will return 255, or 253 with the joystick in the down position.

# 7. COMPATIBILITY SWITCHES & COMMANDS

The Plus 1 Mini features several compatibility settings which can be controlled in hardware or by star commands.

On the rear of the unit is a four-pin DPI switch block. Each switch controls one setting, as below:

Up – Software controlled  
Down – Function disabled

1. Tube
2. SD card
3. Cartridge slots
4. Safe mode

(switch 1 is on the left, 4 on the right)

Available commands are:

## **\*TUBE OFF**

Disables the tube interface

## **\*NOSD**

Disables the MMFS ROM. Note that MMFS copies itself into RAM, so to disable the SD card completely you will need to use the **\*SRWIPE 7** command to clear the RAM copy and then reboot.

## **\*NOCARTS**

Disables the cartridge slots and uses the four sideways banks (0-3) to provide an extra 64KB of sideways ram.

### **\*SAFEMODE**

Disables all the Plus 1 Mini's hardware except the cartridge slots to ensure maximum compatibility.

### **\*ROMREC**

In the event of a failed flash of the MMFS ROM, this command places the flash chip in an emergency mode where a backup copy of MMFS is available. If the command fails, use **?&FC84 = 1** from BASIC.

### **\*VER**

Reports the CPLD firmware version.

# 8. TECHNICAL INFORMATION

## I/O REGISTER MAP

&FC70 R/W – Plus 1 ADC port. See Acorn documentation.

&FC71 R/O – Plus 1 joystick port. See Acorn documentation.

&FC80 R/W – SPI data port. Bytes written to this register are sent to the SD card via SPI. Responses are available by reading from the port after a write.

&FC81 R/W – SPI control and status port. Bit 0 of this register controls the SPI clock, 0 for slow (~200-400KHz, depending on screen mode) and 1 for fast (4MHz). The SPI interface busy status can be determined by reading bit 0; 1 for busy, 0 for idle.

&FC82 R/O – CPLD firmware version

&FC83 W/O – Compatibility options. Bits 0-3 are respectively Safe mode, No cartridges, No SD ROM, No TUBE.

&FC84 W/O – Recovery mode switch. Writing 1 to bit 0 enables ROM recovery mode.

&FCC0 R/O – Slogger joystick. Bits 7-5 are always 1, bits 4-0 are joystick fire, right, left, down, up. Active low.

&FCD0 R/O – First byte joystick. Bits 7-5 are always 1, bits 4-0 are joystick fire, right, left, down, up. Active high.

&FCE0 R/W – TUBE data port. See Acorn documentation.



## 9. ACKNOWLEDGEMENTS

No project as complex as the Plus 1 Mini is done entirely by one person. Many thanks to following people, upon who's work the Plus 1 Mini builds:

Martin Mather, developer of MMFS

David Banks, MMFS maintainer

Also thanks to ABUG Scotland members Derek (WalkerWorks) and Dave (DaveLecky) for kindly providing cartridges for compatibility testing, and to everyone on the Stardot forum for suggestions and encouragement.

The Plus 1 Mini joystick ROM is based on the Slogger joystick ROM.

## REVISION HISTORY

1.0 – 10/11/2024 : Initial release