

DIY Pi Desktop User Manual



Table of Contents

1.	Introduction	3
1.1	Overview	3
1.2	Features.....	3
1.3	Kit Content	3
2.	Getting Started.....	4
2.1	Additional Required Items.....	4
2.2	Power Supply Selection Guide	4
2.3	Assembly Instructions	4
2.4	Starting your Pi Desktop.....	6
2.5	Software Installation	6
2.6	SSD / USB Storage Device Settings (OPTIONAL).....	7
2.6.1	Partition and Format the SSD	7
2.6.2	Initialize Booting from the SSD	10
2.7	Camera Installation (OPTIONAL)	13
2.7.1	Setting up the Camera Hardware	13
2.7.2	Setting up the Camera Software:.....	13
3.	Usage.....	14
3.1	Raspberry Pi Pins Used.....	14
3.2	Power Control	15
3.3	Real Time Clock (RTC).....	18
3.3.1	Adding a Real-Time Clock on Raspbian Jessie.....	18
4.	Software Upgrade and Support.....	21
4.1	GitHub Repository.....	21
4.2	Technical Support.....	21
5.	Appendix	21
5.1	Electrical Characteristics	21
5.2	Dimensions.....	22
5.3	FAQs	22

1. Introduction

1.1 Overview

The Pi Desktop is a desktop computer kit based on Raspberry Pi 2 & 3. It includes a case and an expansion board that can turn a Raspberry Pi into a real desktop PC. It provides an intelligent and safe power controller, a real-time clock, and a high capacity Solid State Drive (SSD) expansion card socket for additional storage.

1.2 Features

- Intelligent On/Off power switch
- mSATA SSD socket for up to 1TB on-board storage
- Integrated RTC (Real Time Clock)
- Includes case and heat sink

1.3 Kit Content

- Add-On board
- Heat sink
- USB Adapter (Micro-Type A)
- Long Spacer (x4)
- Short standoff (x4)
- Screws (x2)
- Enclosure (Base and Lid)
- Button cell, CR2032



2. Getting Started

2.1 Additional Required Items

- Raspberry Pi 3 or 2
- Pre-programmed Micro SD Card
- Power Supply (5V @ 2.5A)
- HDMI Monitor
- HDMI Cable
- USB Keyboard & Mouse
- Camera Module (Optional)
- mSATA SSD, max. up to 1 TB or USB Flash Drive (Optional)

2.2 Power Supply Selection Guide

IMPORTANT: To ensure stability, a minimum requirement of 5V @ 2A power supply is required to power the Pi Desktop. Any power supply with current capacity less than 2A may damage the file system and/or hardware.

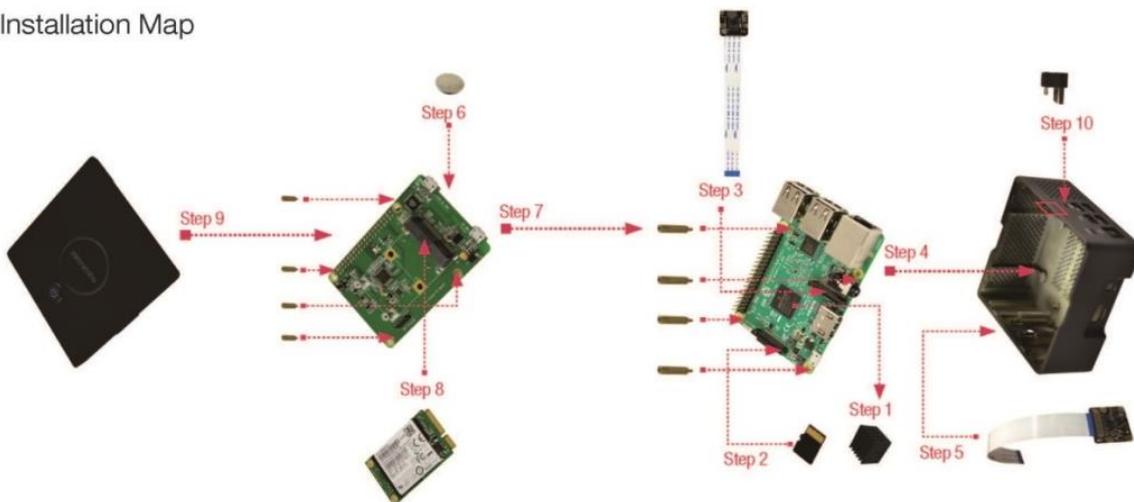
It is recommended to use an official Raspberry Pi power supply:

5V @ 2.5A with Pi3

5V @ 2A with Pi2

2.3 Assembly Instructions

Installation Map



- 1) Remove the protective film from the bottom of the heat sink and place it on the top of the Processor on the Raspberry Pi.
- 2) Insert the pre-programmed micro-SD card into the Raspberry Pi SD card slot.

Don't have one? Download the latest **Rasbian Jessie with PIXEL** image from the below link and write to the micro-SD card using preferred image writer (recommended tool – Win32DiskImager) following the official guide.

<https://www.raspberrypi.org/downloads/>

- 3) (Optional: Complete only if you are installing a Pi camera) – Connect the Pi Camera

into the camera port on the Raspberry Pi.

- 4) Mount the Raspberry Pi into the enclosure using the provided four long spacers.

Note: Ensure the Raspberry Pi orientation is correct as per the connectors on the Raspberry Pi and the slots on the enclosure.

- 5) (Optional: Complete only if you are installing a Pi Camera) – Place the camera into the camera slot in the enclosure. Refer to section 2.6 for further Camera installation instructions.
- 6) Install the button cell on the back of the add-on board.
- 7) Mount the add-on board onto the top of the Raspberry Pi 40 pin GPIO and fasten the board to the Raspberry Pi using the provided four short standoff screws.
- 8) (Optional: Complete only if you are using SSD for booting and storage) – Connect the SSD to the mSATA connector and mount the other end using the provided two small screws. Refer to section 2.6 for further SSD instructions.
- 9) Finally align the enclosure lid with the flap power button so that it is directly above the switch/button on the add-on board and press down gently on the flap – you will hear a click sound when the lid is fit correctly to the enclosure base. (Make sure all items are connected and fastened properly and that there are no loose connectors or screws.)
- 10) Connect the provided USB adapter externally (Type A to micro USB) to the Raspberry Pi USB port and micro USB port with symbol ().
- 11) (Optional: Complete only if you are using USB Flash Drive for booting and storage) – Insert the USB Flash Drive into one of the Raspberry Pi USB ports.
- 12) You are now ready to power your Pi Desktop.

Note: Always ensure your software is up to date by connecting you Pi to the internet, opening a terminal and running the following commands:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

2.4 Starting your Pi Desktop



Connect your Raspberry Pi Desktop to an HDMI monitor using an HDMI cable.

Connect a USB keyboard and a mouse to the Pi Desktop USB ports.

Connect a USB power supply (recommended 5V@2.5A) to the micro USB power port marked with PWR. Then turn the supply ON.

Press the power button on the Pi Desktop and wait for the system to boot.

2.5 Software Installation

To enable the power control, RTC, and the SSD/USB boot feature, a software package will need to be installed. To do so, follow the instructions below:

- 1) Connect your Pi Desktop to the internet using an Ethernet or WiFi network.
- 2) Open your browser and go to www.element14.com/PiDesktop. Located under the download section is a file named "pidesktop-base.deb". Download this file.
- 3) Open a terminal window and go to the directory with your downloaded file "pidesktop-base.deb".
- 4) Install the package typing in the following command into the terminal window:

```
$ sudo dpkg -i pidesktop-base.deb
```

```

pi@raspberrypi: ~/Downloads
File Edit Tabs Help
1a802979493c277008232dd435d2f12f.jpg pidesktop-base-1.1.0.zip
pidesktop-base-1.1.0.deb
pi@raspberrypi:~/Downloads $
pi@raspberrypi:~/Downloads $ sudo dpkg -i pidesktop-base-1.1.0.deb
Selecting previously unselected package pidesktop-base.
(Reading database ... 115363 files and directories currently installed.)
Preparing to unpack pidesktop-base-1.1.0.deb ...
Unpacking pidesktop-base (1.0.0) ...
Setting up pidesktop-base (1.0.0) ...
Created symlink from /etc/systemd/system/reboot.target.wants/embest-shutdown.service to /lib/systemd/system/embest-shutdown.service.
Created symlink from /etc/systemd/system/poweroff.target.wants/embest-shutdown.service to /lib/systemd/system/embest-shutdown.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/embest.service to /lib/systemd/system/embest.service.
sudo systemctl disable fake-hwclock.service
Synchronizing state for fake-hwclock.service with SysVinit using update-rc.d...
Executing /usr/sbin/update-rc.d fake-hwclock defaults
Executing /usr/sbin/update-rc.d fake-hwclock disable
insserv: warning: current start runlevel(s) (empty) of script `fake-hwclock' overrides LSB defaults (S).
insserv: warning: current stop runlevel(s) (0 1 6 S) of script `fake-hwclock' overrides LSB defaults (0 1 6).
pi@raspberrypi:~/Downloads $

```

Press enter to execute the command. If you need to update the old deb packet with the new one you need to remove it first and then install it again. Use the follow command to remove the old deb. Take the new pidesktop-base-1.1.0.deb for example:

```

# Remove the old package first
$ sudo dpkg -r pidesktop-base
# Then install the new one
$ sudo dpkg -i pidesktop-base-1.1.0.deb

```

2.6 SSD / USB Storage Device Settings (OPTIONAL)

The SSD device can only be used in one of two ways, as a storage device or as a booting device.

2.6.1 Partition and Format the SSD

To use the SSD as a storage device, you will initially have to partition and format it.

If you use ppp-hdclone(piclone) to move the exist file system to you SSD you do not need to create a partition table and format the partition. This is because piclone can do these automatically. If instead you use the SSD as a USB storage device you will need to create a partition table and format it yourself.

We can partition/format the mSATA SSD on Raspberry Pi or other Linux PC. The OS should be based on Linux.such as debian.

Note: If you are connecting the mSATA SSD add-on board to a computer other than the Raspberry Pi inside the Pi Desktop, then you will need two micro-USB cables, one for power and the other for USB communication. Both of the micro-USB cables should be plugged into the USB host before you press the Power-On switch. The USB host will then recognize the mSATA. Powering on the USB host will also power on the Raspberry Pi if it is still connected.

Disk Identity

Firstly, ensure you know the device identifiers. The Raspbian OS is based upon Debian Linux, and as such supports the majority of commands and functions that you would use to mount drives within this OS (these typically involve `fstab` - <https://wiki.debian.org/fstab>, and `pmount` - <https://wiki.debian.org/pmount>).

From the Graphical User Interface (GUI), depending on your version of Raspbian and how you have it configured, when you boot your Raspberry Pi to the GUI and connect the attached mSATA SSD you will be prompted on what to do with your inserted “Removable media”. The drive will be automatically mounted.

Note: It is possible that the GUI will not behave in this way and instead you will see a transparent icon representing the drive on your desktop. Double-clicking on this icon with the left mouse button will cause Raspbian to attempt to mount the drive again.

From within the Command Line Interface (CLI), by mounting the drive from the GUI we will make the drive accessible from the CLI. Typically it is located within the folder `/media`, however it is good to have the drive accessible from the CLI as sometimes we can't use the GUI or merely don't want to.

To mount the drive we must first know how Linux is referring to it. Linux has the majority of its hardware listed under the `/dev` folder structure with connected devices typically using the format of `/dev/sd x` where `x` is a letter ranging from 'a' to 'z'. These can even extend further with `/dev/sd xn` where `n` is the number of the partition(s) on the device. For example the USB driver with 2 partitions can be recognized by the kernel like this:

```
pi@raspberrypi:/$ ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2
```

There are a few commands in which we can determine what the device is of a USB device we connect, first, connect the USB device, and then type the following:

```
$ sudo dmesg
```

This will tell you the contents of a system log, providing information about the device you have just plugged in. You can also issue the following commands:

```
$ sudo lsusb
```

Which will tell you about the device identifiers, and also:

```
$ sudo lsblk
```

This command will tell you the `/dev/` mount points.

To select a location to mount the drive, make a directory and set the permissions on it by issuing the following commands:

```
$ sudo mkdir /media/ssd
```

```
$ sudo chmod 755 /media/ssd
```

Once we set the device mount point, you can then issue the following command:

```
$ sudo mount /dev/sd /media/ssd
```

Note: Mounting will only work if the mSATA SSD has a file system and partitions setup. You will want to mount the relevant partition (which has a number, such as `/dev/sdb1`) as opposed to the device itself (`/dev/sdb`).

For more information on this view an article from The Raspberry Pi Foundation's official magazine article on this:

[https://www.raspberrypi.org/magpi/connecting-disks-with-the-command-line/Partition & Format](https://www.raspberrypi.org/magpi/connecting-disks-with-the-command-line/Partition%20&%20Format)

There are two utility tools we recommend for managing partitions are `gparted` from GUI and the `fdisk-mkfs` command from CLI.

1) gparted

An easy to use utility for managing the partitions on the drive is 'gparted' from the Raspbian (OS) (for example if you booted from the SD Card):

<http://gparted.org/>

This is a graphical user interface (GUI) tool that can be used to manage the partitions on your drives, connected via USB or otherwise. It can be installed via the apt package manager from the command line interface (CLI):

```
$ sudo apt-get update
$ sudo apt-get install gparted
```

Then you can either run it from the menu of your Raspbian OS, or from the CLI:

```
$ sudo gparted
```

Start the GUI and follow the prompts.

2) fdisk-mkfs

The fdisk command is a text-based utility for viewing and managing hard disk partitions on Linux.

Assume that the un-partitioned disk is mounted at `/dev/sdc`.

```
sudo fdisk /dev/sdc
```

Follow the prompts to create a new partition-table and create new partition.

Command (m for help): m

Command action:

```
a -> toggle a bootable flag
b -> edit bsd disklabel
c -> toggle the dos compatibility flag
d -> delete a partition
l -> list known partition types
m -> print this menu
n -> add a new partition
o -> create a new empty DOS partition table
p -> print the partition table
q -> quit without saving changes
s -> create a new empty Sun disklabel
t -> change a partition's system id
u -> change display/entry units
v -> verify the partition table
```

- w -> write table to disk and exit
- x -> extra functionality (experts only)

Creating a partition table and new partitions can be done step-by-step or it also can be done by using just one command. For example, the following command can make two partitions on the disk:

```
$ echo -e "o\nn\np\n1\n\n+64M\na\n1\nt\nc\nn\np\n2\n\n\nw\n" | sudo
fdisk /dev/sdc
$ ls /dev/sdc*
sdc1 sdc2
```

As you see sdc1 and sdc2 appears on /dev/, which tells that two partitions have been created on sdc.

After creating the partition you need to format it. Once this is done we can make use of the mkfs tool. Type 'mkfs' then click Tab. This will allow you to see the following types of filesystem formats:

```
$ mkfs
mkfs mkfs.cramfs mkfs.ext3 mkfs.ext4dev mkfs.minix mkfs.ntfs
mkfs.bfs mkfs.ext2 mkfs.ext4 mkfs.fat mkfs.msdos mkfs.vfat
```

For example, we choose fat to format the /dev/sdc1:

```
$ sudo mkfs.fat /dev/sdc1
```

With the default parameters set, type 'Enter' to finish the process.

After all, remember type 'sync' to sync all data to the disk. This is very important for Linux OS.

3) Write an image to the mSATA SSD

Simply, you can write an image to the mSATA SSD just as you would write an image to an SD Card. The Raspberry Pi Foundation has instructions on how you write images here:

<https://www.raspberrypi.org/documentation/installation/installing-images/>

Note: If you write an image to your mSATA SSD, the data used and accessible from the OS will only be of the size of the image written unless you resize the partitions. And the image you want to write must be smaller than the disk size.

2.6.2 Initialize Booting from the SSD

If you want to clone your micro-SD card to the SSD and then boot from the SSD, there are two methods you can follow. The main difference between these two methods is booting with or without using a micro-SD card.

The default method of the Pi Desktop utilizes a micro-SD card; therefore this is the process we will focus on below. For further details on booting without a micro-SD card refer to the following link:
<https://github.com/pi-desktop/pi-desktop/blob/master/Boot-From-a-USB-Mass-Storage-Device/Boot-From-a-USB-Mass-Storage-Device.md>

The Pi Desktop default is to utilize a micro-SD card to initialize booting from an SSD device. In doing so, a command will call the "SD Card copier" to clone the

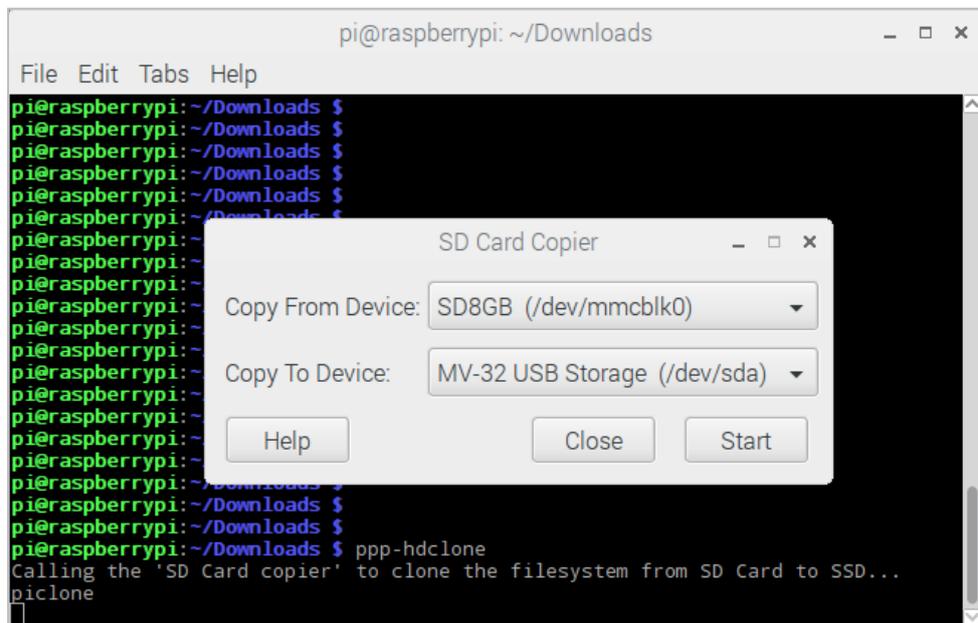
filesystem. You will be asked to select the SSD or USB drive. Select the connected SSD or USB drive and then click “Start”.

Once completed, you will be asked for confirmation that you want to change the filesystem to SSD. Make sure the "SD Card copier" has executed correctly. If it has, select yes ('y') to confirm booting your PiDesktop directly from the SSD, otherwise select no ('n').

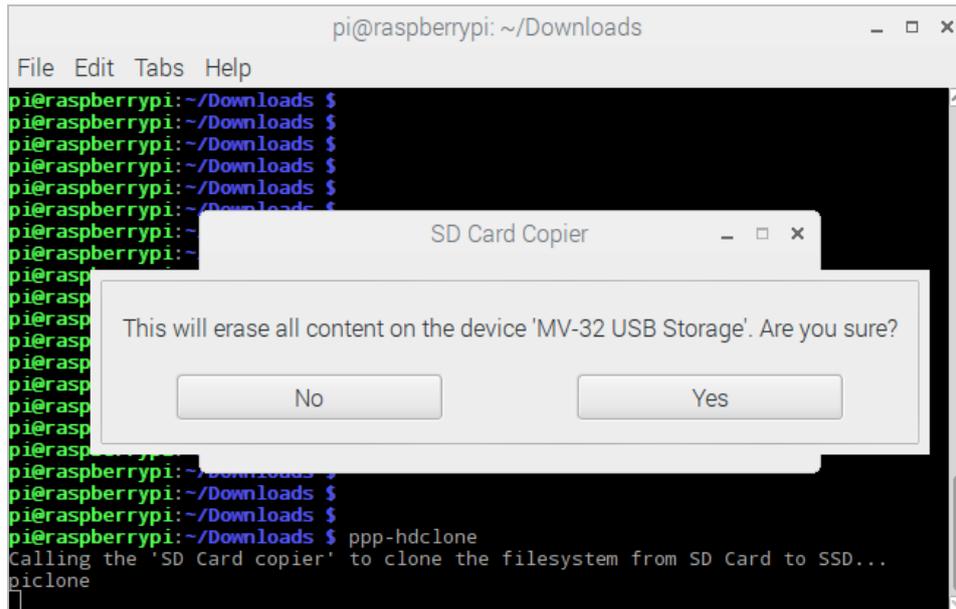
You will now be asked to reboot your system to implement the new configuration and put it into effect.

Once you have successfully rebooted your system, continue with the following steps:

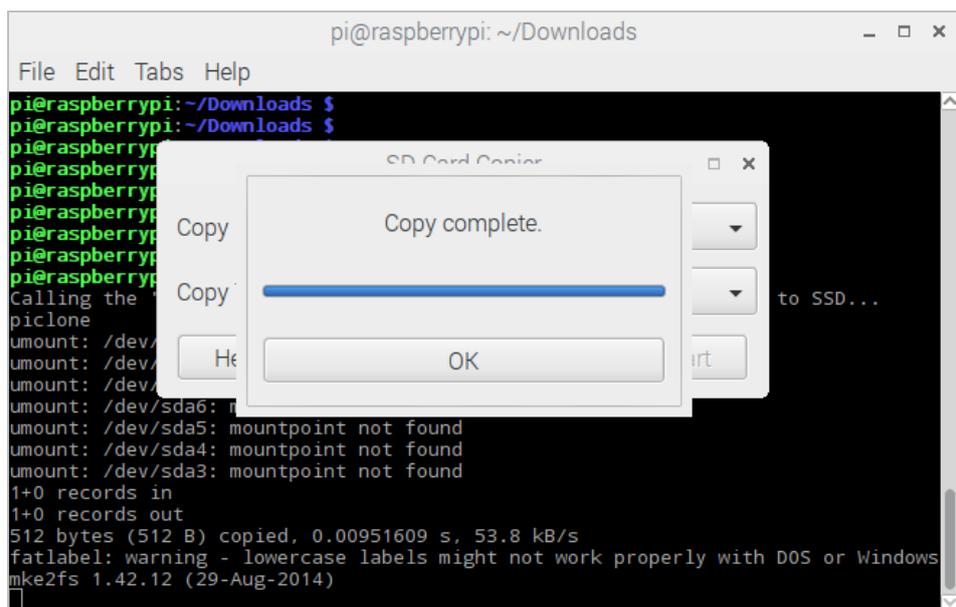
Step 1: Run ppp-hdclone to call the SD Card Copier. DO NOT use ‘sudo’ forward in the new Raspbian image. Select both the source device and the destination device from the drop down lists provided.



Step 2: Make sure your destination USB device contains no useful data, as anything currently on your USB device will be erased.



Step 3: Click 'OK' to complete the copy.



Step 4: You will now be asked if you want to switch the rootfs to your USB device. If so, make sure the former copy process has been successful and input 'y'. You will then be asked to reboot the system.

```

pi@raspberrypi: ~/Downloads
File Edit Tabs Help
pi@raspberrypi:~/Downloads $
pi@raspberrypi:~/Downloads $ ppp-hdclone
Calling the 'SD Card copier' to clone the filesystem from SD Card to SSD...
piclone
umount: /dev/sda9: mountpoint not found
umount: /dev/sda8: mountpoint not found
umount: /dev/sda7: mountpoint not found
umount: /dev/sda6: mountpoint not found
umount: /dev/sda5: mountpoint not found
umount: /dev/sda4: mountpoint not found
umount: /dev/sda3: mountpoint not found
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.00951609 s, 53.8 kB/s
fatlabel: warning - lowercase labels might not work properly with DOS or Windows
mkfs2fs 1.42.12 (29-Aug-2014)
Do you want to change the file system from SD card to SSD?
If 'YES', please make sure the 'SD Card Copier' execute correctly.(y/N):

```

Note: When booting from SSD, ensure there are no other USB Flash Disks plugged into your PiDesktop as this may result in an error of the Raspberry Pi not finding the correct boot device.

2.7 Camera Installation (OPTIONAL)

2.7.1 Setting up the Camera Hardware

Note: The RaspiCam camera modules are static-sensitive. Earth yourself prior to handling the PCB.

The camera board attaches to the Raspberry Pi via a 15-way ribbon cable. The ribbon cable must be attached to the camera PCB, as well as to the Raspberry Pi itself. The camera will not work if the cable is in the incorrect orientation. To ensure correct orientation follow the instructions below:

Although the connectors on the Camera PCB and the Raspberry Pi are different, they work in a similar way. On the Raspberry Pi, pull up the tabs on each end of the connector. It should slide up easily and be able to move back and forth slightly. Fully insert the ribbon cable into the slot, ensuring it is straight and that the blue backing is facing towards the AV and USB connections of the Raspberry Pi. Gently press down the tabs to secure the cable in place. The cable connector on the camera PCB also requires you to pull the tabs away from the board, gently insert the cable, then push the tabs back down to secure the cable. For correct orientation, ensure the blue backing of the cable end is facing away from the PCB (away from the camera lens).

The cable connector on the camera PCB is a little more awkward than the one on the Raspberry Pi itself. You can watch a video showing you how to attach the connectors at www.raspberrypi.org/archives/3890 (scroll down for the video).

2.7.2 Setting up the Camera Software:

Execute the following instructions on the command line to download and install the latest kernel, GPU firmware, and applications. You will need an internet connection for this to work correctly.

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Now you need to enable camera support using the raspiconfig program used when you first set up your Raspberry Pi.

```
$ sudo raspi-config
```

Use the cursor keys to move to the camera option and select *enable*. On exiting raspiconfig it will ask to reboot. The *enable* option will ensure that on reboot the correct GPU firmware will be running (with the camera driver and tuning), and the GPU memory split is sufficient to allow the camera to acquire enough memory to run correctly. To test that the system is installed and working, try the following command:

```
raspistill -v -o test.jpg
```

The display should show a 5-second preview from the camera and then take a picture, saved to the file test.jpg, while displaying various informational messages.

Note: When installing a Raspberry Pi 3 Camera the image orientation within the software will need to be rotated 180 degrees.

To do so, use the command **--rotation, -rot**

This sets the rotation of the image (0-359) in viewfinder and resulting image. This can take any value from 0 upwards, but due to hardware constraints only 0, 90, 180 and 270-degree rotations are supported.

For troubleshooting and more info on the Raspberry Pi Camera Module visit: <https://www.element14.com/community/docs/DOC-78598/1/raspberry-pi-camera-module>

3. Usage

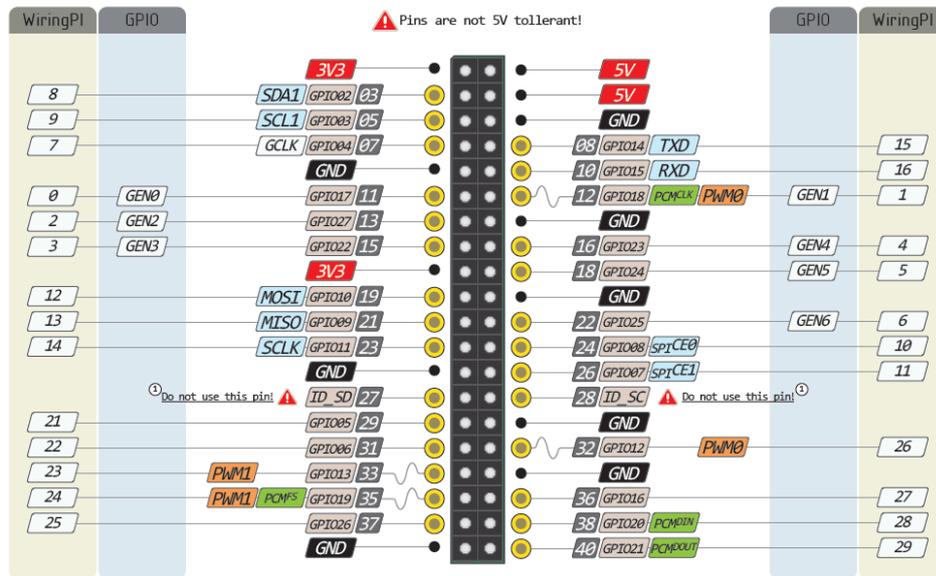
3.1 Raspberry Pi Pins Used

The PiDesktop utilizes the following pins on the Raspberry Pi board in order to achieve reliable power control and RTC function.

Using listed pins listed below (with exception of the two I2C pins) may cause issues. The two I2C pins may be used for I2C communication, but should not be used as General IO pins as this can also cause errors to occur.

GPIO	Function on Pi-desktop	Details
GPIO6	Pin2	The state of the RPi as read by the MCU
GPIO13	Pin1	The pulse signal from the MCU to the RPi
GPIO17	RTC_INTn	Interrupt signal of the RTC

GPIO2	RTC_SDA/MCU_SDA	SDA for both the RTC and MCU
GPIO3	RTC_SCL/MCU_SCL	SCL for both the RTC and MCU
GPIO26	INT_MCU	A GPIO connected to the MCU (reserved)
GPIO19	NRST_MCU	Used by the MCU



3.2 Power Control

The power to your PiDesktop can be controlled in two ways; pressing the power button on the PiDesktop or using a command line or GUI.

PiDesktop Power Button

OPERATION	ACTION
Turn on	Press the power button
Shutdown	Hold the button pressed for approx. 2 seconds
Forced Shutdown	Hold the button pressed for about approx. 5 seconds to cutoff power regardless of any other conditions

Command Line & GUI

A reboot command can be sent from the command line or GUI. During the reboot process the LED will flash. When the system has finished the shutdown process it will automatically restart. During restart LED will be consistently on, not blinking.

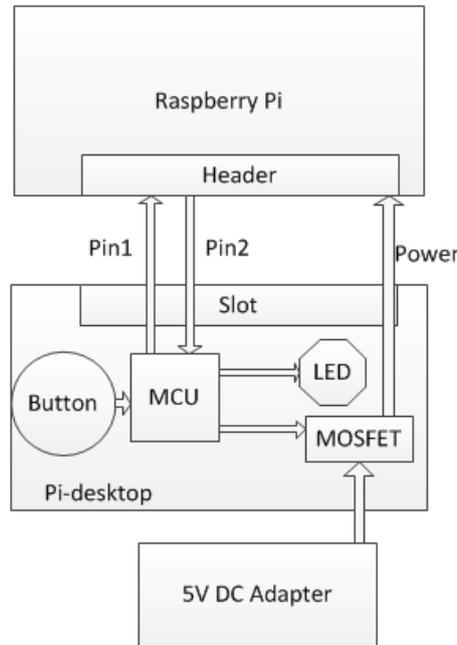
Running the shutdown command from the command line or GUI will command the LED flash for 30 seconds before shutting off, at which point the power to the PiDesktop will be cut.

Safe Power Control

NOTE: Directly cutting power to the RPi can cause damage to the board, therefore the PiDesktop must perform safe startup and shutdown procedures.

Hardware

The hardware architecture used by the PiDesktop to perform safe power control is shown below:



A low power MCU on the PiDesktop add-on board is used to control the power supply and monitor the state of Raspberry Pi board. Two pins (Pin1, Pin2) are used to communicate between these two boards. Information sent on Pin1 is used to inform the Raspberry Pi board to prepare for shutdown. The PiDesktop MCU utilizes Pin2 to continually monitor the run state of the Raspberry Pi board. For added robustness, close-cycle control has been implemented. This allows the PiDesktop power switch to be utilized to perform safe and reliable startup and shutdown procedures.

Action	Button Time
Startup	at least 150ms
Switch off	2 seconds
Force switch off	5 seconds

Software

The PiDesktop add-on board MCU works with the Raspberry Pi board through Pin1 and Pin2. Below are the details of the FSM (finite state machine).

PiDesktop Add-On Board MCU

When USB power is supplied to the PiDesktop add-on board, the MCU will run in an **Initialize** state. The MCU will turn **OFF** (LED off) when the Raspberry Pi board is shutdown (power removed).

When the PiDesktop is turned on (power button action is initially detected), the add-on board will open the power to the Raspberry Pi and go into a **Waiting ON** state.

In the **Waiting ON** state the MCU will continually read the level of the Raspberry Pi board Pin2.

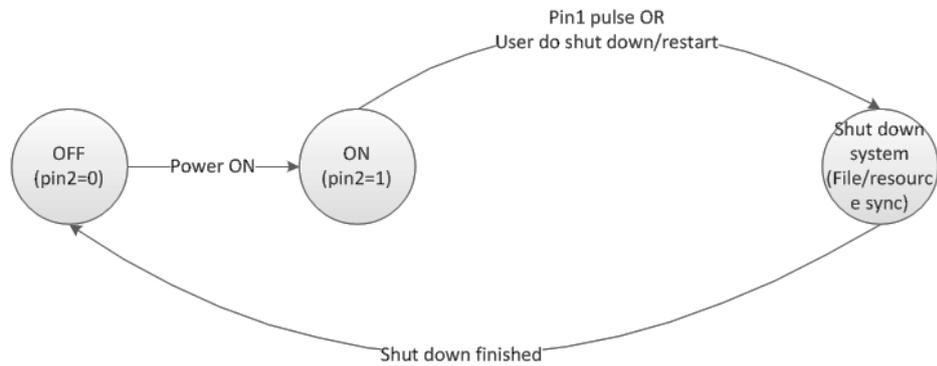


Image: Raspberry Pi Software flow-chat

The PiDesktop power control strategy ensures the main power is cut off only after the Raspberry Pi has been completely shut down. This is important for the safety of the flash chip and the filesystem of the Raspberry Pi board. Without this strategy errors can occur, causing delayed file system checking or a situation where the system cannot start up.

3.3 Real Time Clock (RTC)

Each time your Pi Desktop starts up it will connect to what is called a Network Time Protocol (NTP) server and request the time.

If there is no internet connection the Pi Desktop utilizes the integrated RTC on the Pi Desktop add-on board. This contains a battery powered clock chip that tells the Raspberry Pi what time it is. To ensure accuracy the time will need to be set on the system initially.

Note: *Installing the software package then rebooting the system will enable the RTC.*

3.3.1 Adding a Real-Time Clock on Raspbian Jessie

The Raspberry Pi is designed to be an ultra-low cost computer; therefore the RPI board itself does not include a battery for the RTC. Instead, the RPI is intended to be connected to the Internet via Ethernet or WiFi, allowing it to update the time automatically from the global ntp (network time protocol) servers.

For stand-alone projects with no network connection, the Raspberry Pi will not keep time when power is removed.

This section will explain how to add a low cost battery-backed RTC to your Raspberry Pi to enable time keeping and introduce what the pi-desktop debian package can do to sync time from RTC.

Raspbian Jessie (Systemd)

Step1: Adding a device tree overlay

You can add support for the RTC by adding a device tree overlay.

To edit the pi configuration run the following command:

```
sudo nano /boot/config.txt
```

then add the following to the end of the file:

```
dtoverlay=i2c-rtc,pcf8563
```

```

pi@raspberrypi: /etc
File Edit Tabs Help
GNU nano 2.2.6 File: /boot/config.txt

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable the lirc-rpi module

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

dtoverlay=i2c-rtc,pcf8563
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Save your script and then restart by running:

```
sudo reboot
```

Step2: Disable the "fake hwclock" service

Run:

```
sudo systemctl disable fake-hwclock.service
```

Step3: Comment out three lines in hwclock-set

Run:

```
sudo nano /lib/udev/hwclock-set
```

Now comment out the following three lines of code:

```
#if [ -e /run/systemd/system ] ; then
```

```
# exit 0
```

```
#fi
```

```

pi@raspberrypi: /etc
File Edit Tabs Help
GNU nano 2.2.6 File: /lib/udev/hwclock-set
#!/bin/sh
# Reset the System Clock to UTC if the hardware clock from which it
# was copied by the kernel was in localtime.

dev=$1

#if [ -e /run/systemd/system ] ; then
#   exit 0
#fi

if [ -e /run/udev/hwclock-set ]; then
    exit 0
fi

if [ -f /etc/default/rcS ]; then
    . /etc/default/rcS
fi
    
```

Step4: Sync time from Pi to RTC

Setting the correct time is simple. First, run the date to verify the time is correct. Connect the Raspberry Pi to the internet via an Ethernet or WiFi to allow the Raspberry Pi to initially sync the correct time from the Internet. Once this has been done, run:

```
sudo hwclock -w
```

to write the time, followed by:

```
sudo hwclock -r
```

to read the time

```

pi@raspberrypi: /etc
File Edit Tabs Help
pi@raspberrypi:/etc $ date
Mon 31 Jul 08:27:40 UTC 2017
pi@raspberrypi:/etc $ sudo hwclock -w
pi@raspberrypi:/etc $ sudo hwclock -r
Mon 31 Jul 2017 08:27:53 UTC -0.389645 seconds
pi@raspberrypi:/etc $
    
```

Once the time is set, make sure the coin cell battery is inserted so that the time is saved. You only have to set the time once. The next time you boot the Raspberry Pi, the time will automatically be synced from the RTC module.

Step5: Update the RTC date during reboot and poweroff

A systemd service is required in order to update the RTC date during a reboot or power off. This will cause systemd to execute a script called sync-hwclock to

run `sudo hwclock -won` reboot and power off.

[Unit]

`Description=Reboot test`

`DefaultDependencies=no`

`Before=shutdown.target`

[Service]

`Type=oneshot`

`ExecStart=/bin/bash /usr/share/PiDesktop/script/sync-hwclock`

[Install]

`WantedBy=reboot.target poweroff.target`

What pi-desktop deb package do

After you installed the deb package and reboot the Raspberry Pi you will be able to read the time directly from the RTC using the following command:

`sudo hwclock -r`

Within the debian package there is a systemd service set up called sync-hwclock. This writes the latest time onto the RTC on boot and power off.

4. Software Upgrade and Support

4.1 GitHub Repository

Source code and technical documents can be found in the GitHub repository: <https://github.com/pi-desktop>.

The latest Debian package can be downloaded here: <https://github.com/pi-desktop/deb-make/releases>.

4.2 Technical Support

For upgrades and technical support please visit the PiDesktop product homepage: www.element14.com/PiDesktop.

5. Appendix

5.1 Electrical Characteristics

CHARACTERISTIC	VALUE
----------------	-------

Operating temperature	0~50°C
Storage temperature	-40~80°C
Operating voltage	5V
Power Consumption	Value
Start-up process (Max)	1100mA

5.2 Dimensions

DIMENSION	VALUE
Case	108 x 108 x 43.12 mm
Add-on Board	87 x 56 x 1.4 mm

5.3 FAQs

- *My power control does not work. How do I fix this?*

First, ensure you install the software package. Once this has been done, reboot the system.

- *Can I use a third party power charger as the power supply?*

To ensure stability, use a power charger/supply that is rated with a minimum of 5V @ 2A. Any power supply with current capacity under 2A may damage the file system and/or hardware.

- *Do I have to have and SSD drive?*

No, an SSD drive is not necessary.

WARNING: Avoid damage and contact with water.