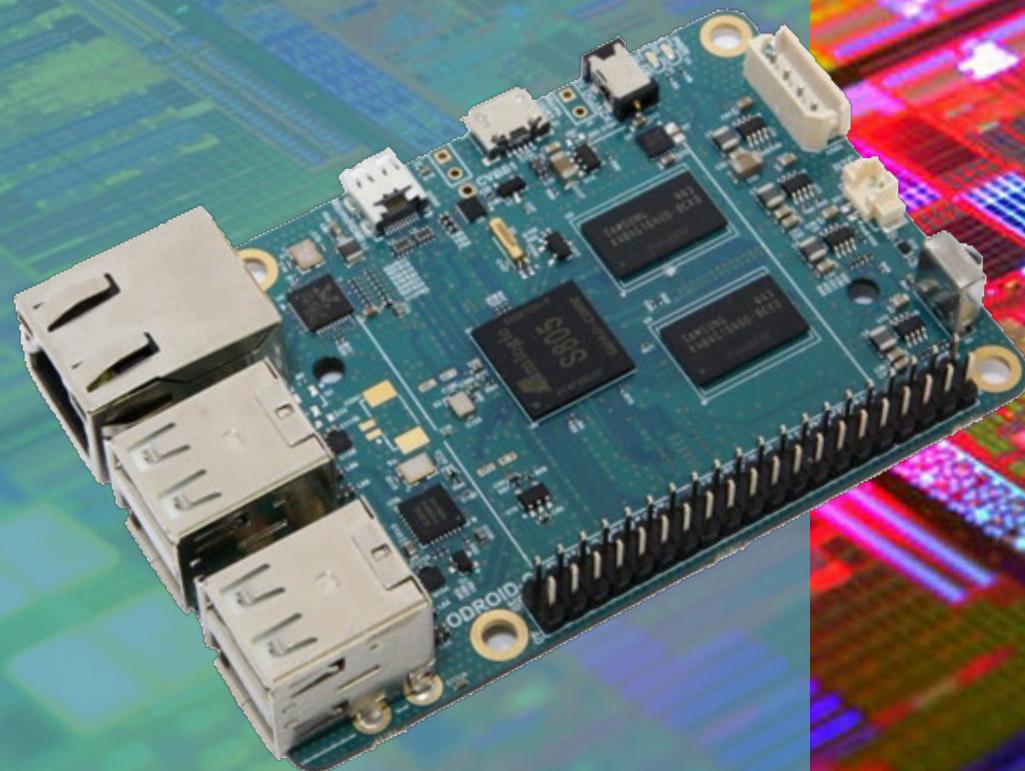




HARDKERNEL

USER MANUAL



ODROID-C1

Index

ODROID-C1 Beginner's Guide

Chapter 1 • Welcome	1
Differences between a typical PC and a Single Board Computer (SBC)	2
Components Included on an SBC	2
Block Diagram	3
Board Image	3
Attaching Peripherals	3
Power Supply	5
Monitor.....	6
Keyboard and Mouse	6
Ethernet and Wifi.....	7
MicroSD Card	7
eMMC Module	7
LED Status	8
Technical Specifications	8
Heatsink.....	11
Temperature Measurement	14
Frequently Asked Questions.....	14
Chapter 2 • Getting Started.....	19
Home Computing Network	19
Preparing the Work Area	20
Flashing an Image	21
Boot Media	21
Windows 7+	24
Linux.....	27
OSX.....	29
Powering Up	29
Troubleshooting	29
Configuring Linux.....	31
Configuring Android	32
Powering Down	33

Index

ODROID-C1 Beginner's Guide

Chapter 3 • Operating Systems	34
Ubuntu/Debian.....	34
Linux Basics	35
Kernel	36
GUI	36
720p vs 1080p.....	36
Progressive vs Interlaced Video	37
Video Downconversion.....	37
Video Upconversion	38
HDMI Overscan	38
Disabling Monitor Overscan	39
Display Setting Button on Remote Control.....	39
Command Line Interface	40
Disk Partitions.....	40
Web Browsing	41
Kodi (formerly XBMC).....	41
Office and Productivity Applications	42
Music and MIDI.....	43
How do I Add a MIDI Interface to the C1?.....	43
Experimental Music with the C1	44
Android	44
Desktop Environment	44
ODROID Utility and Updater.....	45
Setting the Display Resolution.....	45
Kodi	46
Netflix.....	46
Gaming	46
Using Bluetooth Devices with Android.....	47
Adding an ODROID-VU Touchscreen	47
Chapter 4 • Hardware Tinkering	48
USB/UART	48
Bluetooth Module 2.....	53

Index

ODROID-C1 Beginner's Guide

Tinkering Kit.....	58
ODUINO ONE	61
ODROID-SHOW2.....	63
16x2 LCD I/O Shield.....	65
Weather Board	66
3.2inch TFT+Touchscreen Shield.....	68
USB-DC Plug Cable 2.5x0.8mm	71
USB-SPDIF	71
USB-CAM 720P.....	74
UPS2	76
USB GPS Module.....	76
Conclusion.....	79
Additional Resources.....	79
Addendum	
C1+ Features.....	80

Credits

Authors: Rob Roy
Venkat Bommakanti

Art Editor: Bruno Doiche

Contributors: Alan Tung
Anawat Hokoon
Patrick Oberli
Robert Cleere

What we stand for.

We strive to symbolize the edge of technology, future, youth, humanity, and engineering.

Our philosophy is based on Developers. And our efforts to keep close relationships with developers around the world.

For that, you can always count on having the quality and sophistication that is the hallmark of our products.

Simple, modern and distinctive. So you can have the best to accomplish everything you can dream of.



HARDKERNEL

© 2015 Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 431-815
Hardkernel manufactures the ODROID family of quad-core development boards and the world's first ARM big.LITTLE single board computer.
Read our monthly magazine at <http://magazine.odroid.com>.
You can join the growing ODROID community with members from over 135 countries at <http://forum.odroid.com>.
Explore the new technologies offered by Hardkernel at <http://www.hardkernel.com>.

Welcome

Congratulations on purchasing the ODROID-C1! It is one of the most powerful low-cost Single Board computers available, as well as being an extremely versatile device. Featuring a quad-core AmLogic processor, advanced Mali GPU, and Gigabit ethernet, it can function as a home theater set-top box, a general purpose computer for web browsing, gaming and socializing, a compact tool for college or office work, a prototyping device for hardware tinkering, a controller for home automation, a workstation for software development, and much more.

Some of the modern operating systems that run on the ODROID-C1 are Ubuntu, Android, Fedora, ARCLinux, Debian, and OpenELEC, with thousands of free open-source software packages available. The ODROID-C1 is an ARM device, which is the most widely used architecture for mobile devices and embedded 32-bit computing. The ARM processor's small size, reduced complexity and low power consumption makes it very suitable for miniaturized devices such as wearables and embedded controllers.

Chapter 1

Differences between a typical PC and a Single Board Computer (SBC)

If you are used to using a standard PC such as an OSX or Windows machine, there are a few small differences to note when transitioning to an ARM device. To begin with, the speed of an ARM processor is not directly comparable to the speed of an Intel processor. Because of the efficiency of the CPU, the C1 can give great response time that feels just as fast as using a more expensive computer. The operating systems available for the C1 are also highly optimized, and benefit from the expertise of many open-source contributors that continually review each others' work that bring daily improvements to the OS.

In addition, nearly all of the applications available for the C1 also have their source code publicly available, which means that you can freely modify and update the applications to fit your specific needs. Program authors often maintain a GitHub repository, where suggestions can be submitted, reviewed and distributed to all of the application's users.

The C1 also uses Solid State technology for its storage media, although a conventional hard disk may be used as an auxiliary device. The boot partition can be stored on either a microSD card or the much faster eMMC module, and Hardkernel's products have the unique distinction of supporting removable eMMC modules, so that operating systems may be switched out conveniently and easily. An eMMC module is a type of storage typically used in a smart phone, and is one of the more advanced compact media devices available.

The power consumption of a typical personal computer can be anywhere between 100W and 1000W or more, depending on the peripherals, processor and type of power supply used. However, the ODROID-C1 uses between 5W and 10W, greatly reducing your electricity bills, as well as allowing unique power configurations such as compact solar power cells and long-running batteries.

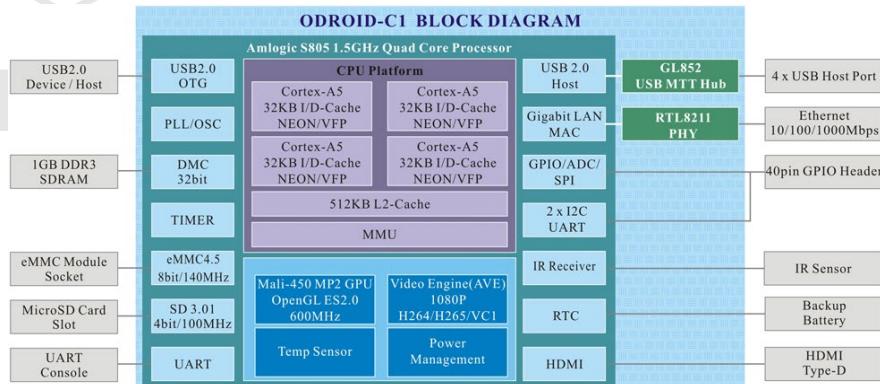
Components Included on an SBC

The ODROID-C1 contains many of the same connections as a typical computer, with 4 USB ports, a microUSB OTG port, an Ethernet port that supports Gigabit transfer speeds, a microHDMI connector for 720p and 1080p monitors, and a 5V/2A DC power connector. In addition to these standard inputs, the C1 also includes a 40-pin GPIO port, an external RTC battery connector, a USB-UART serial console port, an eMMC module connector, and a dedicated slot for a microSD card. For more details, refer to the ODROID-C1 introductory video at <http://youtu.be/DvJSCzIGr1E>.

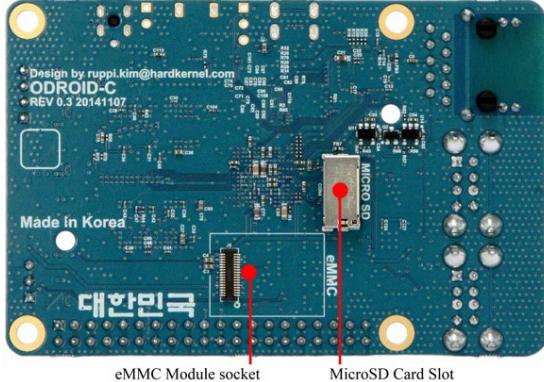
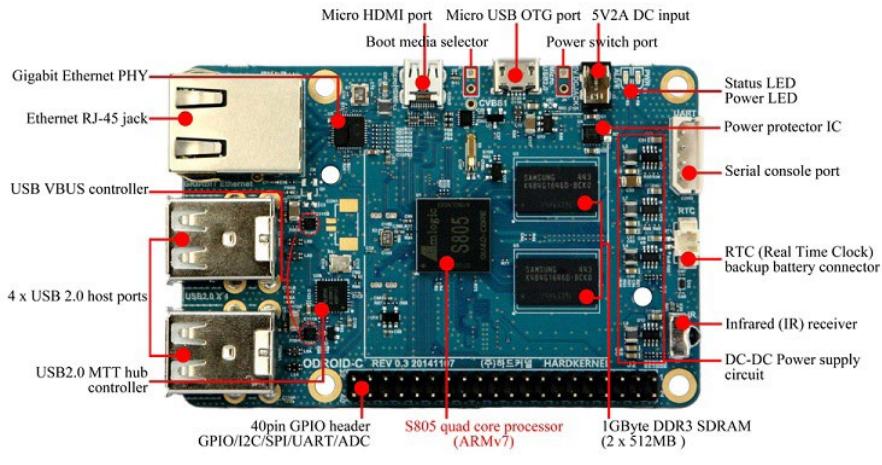
Chapter 1

Block Diagram

The following diagram illustrates conceptually how the components of the C1 fit together:



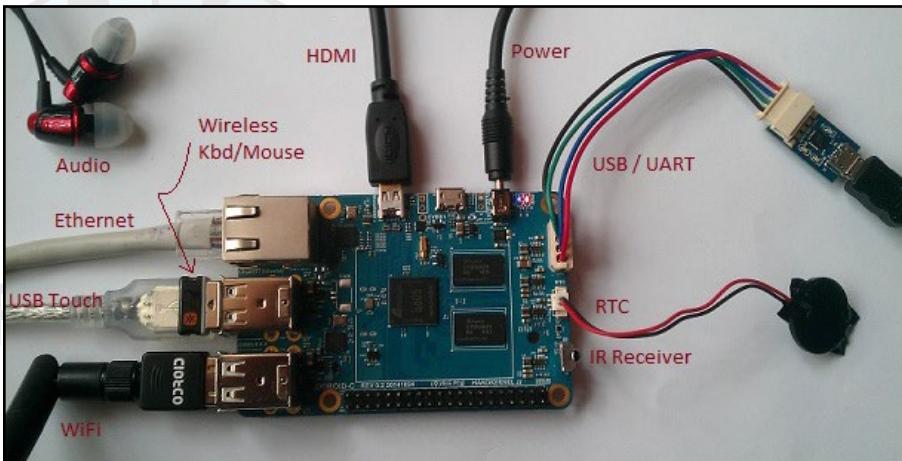
C1 Block Diagram and Annotated Board Image



Attaching Peripherals

The C1 supports the services of numerous peripherals. The image below depicts the following peripherals: Power Supply Unit (PSU) cable, USB / UART debug cable, RTC battery cable, Infra Red (IR) receiver, USB WiFi dongle, HDMI cable, USB / Wireless keyboard / mouse dongle, USB Touch cable, Ethernet cable, and the audio headphone cable attached to the 3.5mm port on an HDMI monitor such as an ODROID-VU.

Chapter 1



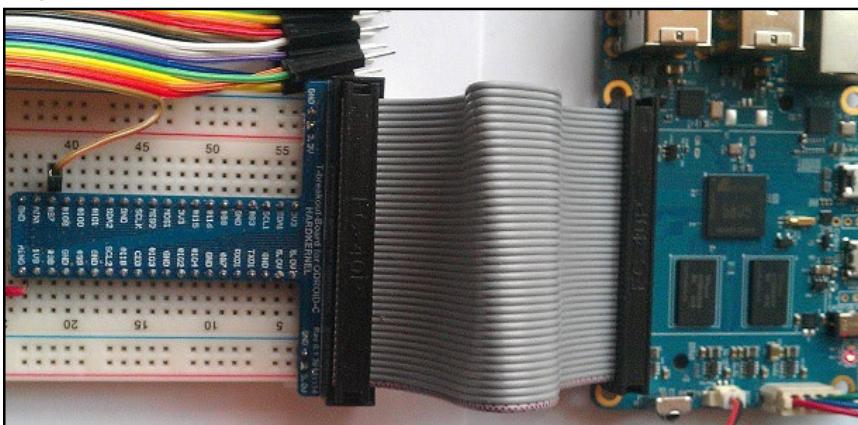
ODROID-C1 peripherals

The image below shows the use of an ODROID-VU HDMI touch-screen monitor supporting stereo audio, and 10-point touch control.



ODROID-VU HDMI Monitor

This image depicts the use of the Tinkering kit designed for C1 useful to experiment with circuit design and related software monitoring and control, which is detailed in Chapter 4.



C1 Tinkering kit

Chapter 1

Power Supply

The C1 board requires a 5V/2A DC power source. The dedicated power connector (miniature barrel jack) can accept a DC plug cable with a plug that has an outer diameter of 2.5mm and an inner diameter of 0.8mm. The plug inner core (center) is positive (attached to the RED wire in the cable) and outer cylinder is negative (attached to BLACK wire in the cable). The C1 can be powered using different options, which are outlined below.



5V DC 2A PSU

Attach the plug to the power connector on the C1. Plug the 2-pin PSU into the power outlet. The pins are of Asian standard, and you may need an adapter to use in your region - such as the Americas. The PSU pictured above is available from Hardkernel.



USB DC plug cable

Attach the USB end of this cable to the USB port of a charging device that can generate 2A @5V, and attach the plug to the C1. This cable is also available from Hardkernel.



DC plug cable

If you have a 5V DC 2A PSU which does not have the required plug, you can cut off the plug from such a power supply. Expose about $\frac{1}{2}$ " of the red/black wires on the psu cable and attach them to the same colored cables of this cable, then solder the joints. You can cover the joint using electrical tape or a heat-shrink wrap. Attach the plug to the

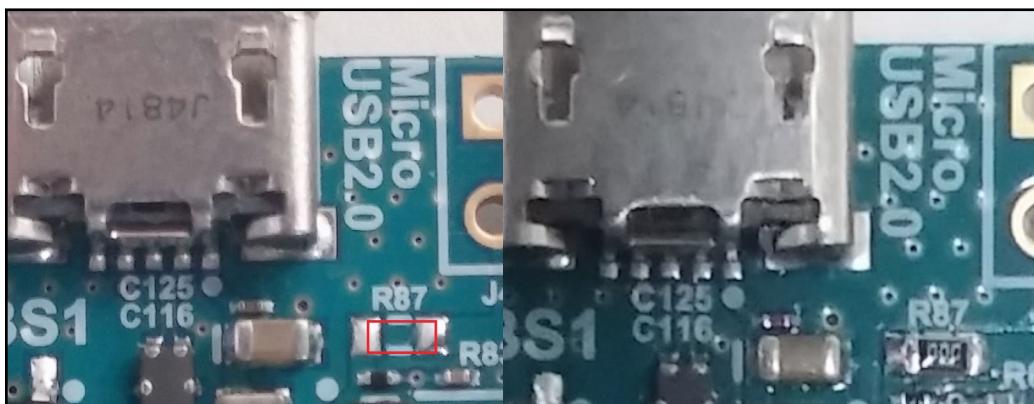
Chapter 1

C1 and insert the PSU pins into a power outlet.

This cable is also available from Hardkernel, and may be paired with the SmartPower peripheral, which is an excellent bench power supply with variable voltage.

MicroUSB Power

The ODROID-C1 is capable of being powered by the microUSB port, but the option isn't enabled by default on the original C1, although the C1+ model comes with this modification. A simple hardware bridge permits the USB power feature, which is detailed in the images below.



Powering the C1 via microUSB

To perform the modification, you may select one of the following two methods: Bridge the two pads of R87 with soldering, as shown in the image on the left, or mount a 0-ohm resistor on R87 (type 1608), as shown in the image on the right.

Monitor

The C1 offers a microHDMI port for connecting an HDMI-compliant monitor. It is recommended to use the Hardkernel supplied microHDMI to HDMI adapter cable, but many other high quality standards conform cable should also work. There are some reported issues with cables that lack HDMI grounding wires inside the cable, so it is best to order this cable directly from Hardkernel or one of its certified distributors.

Keyboard and Mouse

Nearly all USB HID-compliant keyboard and mouse will work when connected to one of the four USB ports. The use of a bluetooth mouse or keyboard requires a bluetooth dongle and, for first pairing, either an USB mouse/keyboard, an SSH access from remote or a working touchscreen.

To pair a bluetooth keyboard or mouse via the Linux console, run

Chapter 1

the following command in a Terminal window:

```
$ sudo hcitool scan
```

Push the Connect button on the bluetooth device and it will appear in the console:

```
Scanning ...  
XX:XX:XX:XX:XX:XX      Rapoo E6700  
$ sudo bluez-simple-agent hci0 XX:XX:XX:XX:XX:XX
```

Push the Connect button again, and enter the shown pin on the keyboard, followed by the Enter key. If no pin is shown, try 000000. Then, type the following to trust the device and restart the bluetooth service:

```
$ sudo bluez-test-device trusted XX:XX:XX:XX:XX:XX yes  
$ sudo /etc/init.d/bluetooth restart
```

In Android, the Settings application may be used to connect to the bluetooth keyboard or mouse. More details on using bluetooth with Android are covered in Chapter 2.

Ethernet and WiFi

The Ethernet port accepts a standard Ethernet cable, and is capable of up to 1 GB (1000 MB) per second transmission rate. The WiFi USB adapter fits in any of the 4 USB ports.

MicroSD Card

Align the metal strips of the microSD card with the pins of the microSD card connector, and slowly push it in until it clicks in place. Be gentle. If you are unable to push it in, it may be misaligned. Recheck and flip the microSD card if you notice a wrong insertion direction. The image below shows a properly mounted microSD card.

eMMC Module

Align the eMMC module and the eMMC connector on the C1 board, using the white rectangle on the PCB as a guide. The female portion of the eMMC module should line up with the male connector on the board. Slowly push it in until it clicks in place. Be gentle. If you are unable to push it in, it may be misaligned. Recheck and turn the eMMC module if you notice a wrong insertion direction. The writing on the card will be exposed after insertion. The image below depicts how the eMMC module appears after it is mounted.

Chapter 1



C1 closeup with eMMC module attached

LED status

The ODROID-C1 includes several LED lights that indicate the status of the device:

The red LED

- Is on when power is available

The blue LED

- Is on (solid light) when the bootloader is running

The blue LED

- Blinks slowly when the kernel is running, like a heartbeat

The blue LED

- Blinks quickly when the kernel is in panic mode

Technical specifications

Processor

Amlogic S805: Quad Core Cortex™-A5 processor with Dual Core Mali-450 GPU

eMMC module socket

8GB/64GB : Toshiba

16GB/32GB : Sandisk iNAND Extreme

The eMMC storage access time is 2-3 times faster than the SD card. You can purchase 4 size options: 8GB, 16GB, 32GB and 64GB. Using an eMMC module will increase speed and responsiveness, similar to the way in which upgrading to a Solid State Drive (SSD) in a typical PC also improves performance over a mechanical hard drive (HDD).

Chapter 1

Micro Secure Digital (microSD) card slot

There are two different methods of storage for the operating system. The first is by using a microSD Card and another is by inserting an eMMC module, which is normally used for external storage for smartphones and digital cameras. The ODROID-C1 can utilize a newer UHS-1 SD model, which is about twice as fast as a class 10 card.



C1 closeup with microSD card attached

There are some microSD cards which cause an additional boot delay time of around 30 seconds. According to our testing, most Sandisk microSD cards don't cause a long boot delay. The ODROID-C1+ model is compatible with a wider array of microSD cards than the original C1.

5V 2A DC input

The DC input is for 5V power input, with an inner diameter of 0.8mm, and an outer diameter of 2.5mm. The ODROID-C1 consumes less than 0.5A in most cases, but it can climb to 2A if many passive USB peripherals are attached directly to the main board.

USB host ports

There are four USB 2.0 host ports. You can plug a keyboard, mouse, WiFi adapter, storage or many other devices into these ports. You can also charge your smartphone with it! If you need more than 4 ports, you can use a powered external USB hub to reduce the power load on the main device.

MicroHDMI port

To minimize board size, we used the Type-D micro-HDMI connector. A regular HDMI cable may be used with a suitable microHDMI adapter. The C1+ model uses a standard Type-A HDMI connector.

Chapter 1

Ethernet RJ-45 jack

The standard RJ45 Ethernet port for LAN connection supports 10/100/1000 Mbps speed. The green LED Flashes when there is 100 Mbps connectivity, and the yellow LED Flashes when there is 1000 Mbps connectivity.

Infrared (IR) receiver

This is a remote control receiver module that can accept standard 37.9Khz carrier frequency based wireless data in NEC format.

MicroUSB OTG port

You can use the standard microUSB connector with Linux gadget drivers on your host PC, which means that the resources in the ODROID-C1 can be shared with other PCs. You can also add a microUSB to HOST connector if you need an additional USB host port. This port can also be used for power input with the modification described earlier in this chapter.

General Purpose Input and Output (GPIO) ports

These 40pin GPIO port can be used as GPIO/I2C/SPI/UART/ADC for electronics and robotics. The 40 GPIO pins on an ODROID-C1 are a great way to interface with physical devices like buttons and LEDs using a lightweight Linux controller.

If you're a C/C++ or Python developer, there's a useful library called WiringPi that handles interfacing with the pins. We've already ported the WiringPi v2 library to ODROID-C1, as described in Chapter 4. Note that all of the GPIO ports are 3.3Volt, and the ADC inputs are limited to 1.8Volt.

Serial console port

Connecting to a PC gives access to the Linux console. You can monitor the boot process, or to log in to the C1 to perform root maintenance. Note that this serial UART uses a 3.3 volt interface, and it is recommended to use the USB-UART module kit available from Hardkernel. A Molex 5268-04a (2.5mm pitch) is mounted on the PCB, and its mate is Molex 50-37-5043 Wire-to-Board Crimp Housing.

RTC (Real Time Clock) backup battery connector

If you want to add a RTC functions for logging or keeping time when offline, just connect a Lithium coin backup battery (CR2032 or equivalent). All of the RTC circuits are included on the ODROID-C1 by default. It connects with a Molex 53398-0271 1.25mm pitch Header, Surface Mount, Vertical type (Mate with Molex 51021-0200).

Chapter 1

Gigabit Ethernet

The Realtek RTL8211F is a highly integrated Ethernet transceiver that complies with 10Base-T, 100Base-TX, and 1000Base-T IEEE 802.3 standards.

USB MTT hub controller

A GENESYS LOGIC GL852G is used to implement the 4-port Hub function which fully complies with Universal Serial Bus Specification Revision 2.0.

USB VBUS controller

A NCP380 Protection IC for USB power supply from OnSemi.

Boot media selector

If this port is open, the first boot media is always eMMC. If this port is closed, the first boot media is always microSD card.

Power switch port

You can add a slide switch or rocker switch on this port if you want to implement a hardware on/off switch. If this port is open, the power is on. If this port is closed, the power is off.

Power supply circuit

Discrete DC-DC converters LDOs are used for CPU/DRAM/IO power supply.

Power protector IC

The power protected is a NCP372 over-voltage, over-current, reverse-voltage protection IC from OnSemi.

Heatsink

Electronic components all generate heat while operating, and different components generate different levels of heat. Some components do not require any cooling, while others do.

Complex components such as the C1 processor may reach temperatures as high as 85°C. At high temperatures, the processor will throttle itself and operate slower so that temperatures do not continue to increase.

Some owners prefer that the temperatures do not reach such high temperatures and install a heatsink, which is available from the Hardkernel store. Heat transfer from components to the surrounding air is related to the surface area available to transfer heat to the surrounding air. The processor of the C1 provides a relatively small area

Chapter 1

to dissipate heat (approximately 1cm x 1cm). The heatsink is much larger and is therefore able to dissipate more heat into the surrounding air than the processor itself.

Heatsink Installation

Check that you have both parts of the heatsink. You should have a silver heatsink and a rectangular grey, gummy piece (the thermal pad). These two items are shown in the image along with the C1 unit.

- Remove the transparent plastic backing on the grey thermal pad. There may be transparent plastic on both sides. Then place the thermal pad over the processor and memory modules as shown in the following image.
- Remove the plastic film from the white pad on the underside of the heatsink. The white pad should be stuck to heatsink.
- Align the white pushpins to the correct holes and push both pins through the holes until a click is heard from each. Check that the pushpins have been inserted correctly pressing down on them from the top. They should move a small distance downward freely and then come back up when you let go.

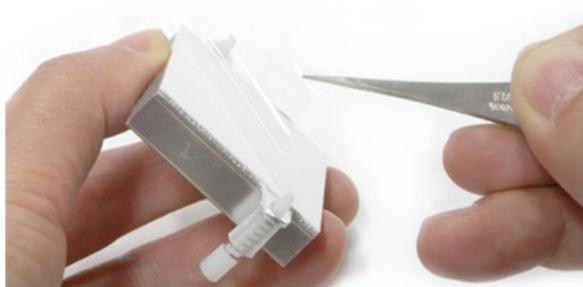


Heatsink components

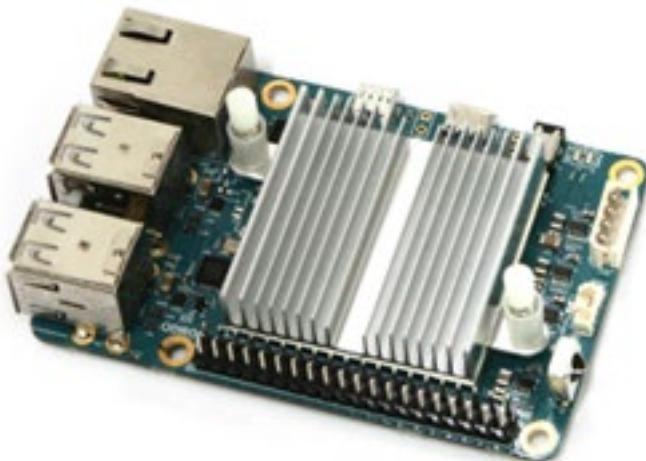
Chapter 1



Heatsink thermal pad applied



Removing heatsink adhesive protection



Heatsink applied to ODROID-C1

The heatsink's purpose is to transfer heat from the processor and memory modules. Therefore it is normal for the heatsink to feel warm when the C1 is operating. A warm heatsink is actually a sign that the installation was done correctly and heat is being transferred from the modules to the heatsink.

If the C1's operating environment has higher temperatures, it may be useful to install the heatsink to avoid throttling and lowered performance. An optional fan is also available for applications that require intensive CPU use.

Chapter 1

Temperature Measurement

To help verify that the heatsink was correctly installed, it is useful to get temperature readings before and after installation. The processor's built-in sensor can be used for this. It would be ideal to have the processor working under the same load when the temperatures for before and after heatsink installation are taken. It is assumed that your unit has been through the initial setup procedure and is running the Ubuntu image from Hardkernel.

- Open two terminal windows, where one window will be used to run the application used to test the processor. Ensure that there are no other programs or applications being run.
- In the first terminal window, type in the following command:

```
$ sysbench --test=cpu --num-threads=4 \
--cpu-max-prime=1000000000
```

- After waiting at least 5 minutes, type the following into the second window:

```
$ cat /sys/class/thermal/thermal_zone0/temp
```

- This result is in Celsius if it has 2 digits. If it has 5 digits that value should be divided by 1000 to get the temperature in Celsius.
- Re-type the command to get the temperature every few minutes until the temperature no longer increases.
- Record the temperature.

This procedure should be done both before and after the heatsink installation to help verify that the installation was done correctly.

Frequently Asked Questions

What's an ODROID?

ODROID means Open + Android. It is a development platform with hardware as well as software.

What SoC are you using?

The SoC is an Amlogic S805 quad-core.

Chapter 1

What GPU does it include?

An ARM Mali-450 MP2. In fact, there are 4 GPU cores: 2 fragment processors and 2 vertex shader processors, but normally we count the number of fragment processors only.

How does it boot?

You may boot from either microSD card or eMMC module.

Can I buy OS pre-loaded SD card or eMMC?

Yes, they are available from Hardkernel as well as other distributors.

Can I add extra RAM memory?

No. The RAM is not removable or swappable.

How can I get the Ubuntu or Android BSP source code?

The U-boot, Kernel and OS source code are released via Github from the first shipping date at <http://github.com/hardkernel>.

Does it play 1080p content well on Ubuntu Linux?

Yes, H.264/H.265/VC1/MPEG4/MPEG2 video clips are playable with Kodi (formerly XBMC) in most cases.

How can I root Android?

Android for the ODROID is unlocked and rooted by default for development.

How can I install the Google Play Store?

It is very simple. Just download an installer from <http://bit.ly/1gkv4PM>, click the APK, and follow the instructions inside the application.

Can I get PCB layout file and gerber file?

The ODROID project is not a full open source hardware, and only the schematics are released to the public.

What peripherals are available?

The following peripherals are available from the Hardkernel store and certified distributors:

- WiFi Module
- Micro-HDMI Cable
- MicroSD 8GB, 16GB cards (with a pre-installed OS)
- eMMC 8GB, 16GB, 32GB, 64GB module (with a pre-installed OS)
- GPIO breakout board with tinkering kit

Chapter 1

- 16x2 character LCD shield
- MicroUSB cable
- USB-UART module kit
- USB CAM 720p
- Infrared (IR) remote control
- Backup Battery for RTC
- LCD HDMI 9" capacitive touchscreen Kit
- ODROID Smart Power Supply

What are the minimum peripherals to run Ubuntu or Android with an ODROID-C1?

The following peripherals are not included with the basic ODROID-C1 board, but may be purchased separately from the Hardkernel store, or from a certified distributor:

- HDMI monitor for output device & Micro-HDMI Cable
- Mouse for input device
- MicroSD card for the operating system and user data (8GB or higher is required)
- MicroSD card reader to install the operating system image
- Ethernet cable
- Power supply: DC 5V/2A
- An eMMC module is recommended over the microSD card due to much higher I/O speed

I am a platform/OS developer. What should I buy with ODROID-C1?

You should obtain the minimum peripherals, along with:

- USB-UART module Kit for debugging and system console
- MicroUSB cable for fastboot and ADB interface

I have a USB-Serial converter. Should I buy your USB-UART module kit?

We strongly recommend using our USB-UART module kit because it includes the proper connector and voltage.

How do I access the Internet with an ODROID-C1?

Use a 10/100/1000 LAN Ethernet connection, or purchase the WiFi module kit for a wireless connection.

What display can I use?

There is an HDMI Type-D output port on the ODROID-C1, and an HDMI Type-A on the C1+. Below is a list of resolutions that are currently supported:

Chapter 1

- 1920x1200 (*WUXGA*)
- 1920x1080 (*1080p*)
- 1280x720 (*720p*)
- 720x480 (*480p*)
- 720x576 (*576p*)
- 1280x800 (*800p for ODROID-VU*)
- 1280x1024 (*SXGA*)
- 1024x768 (*XGA*)
- 800x600 (*SVGA*)
- 800x480 (*WVGA*)
- 640x480 (*VGA*)

Can I use an HDMI to DVI converter?

An HDMI-DVI converter may work with many DVI monitors, but a few of them will not work due to compatibility issues. We recommend our 9-inch HDMI LCD kit (ODROID-VU) with a capacitive touch screen if you want to develop a modern user interface.

Is there a touch screen on the 9-inch HDMI LCD Kit?

Yes. It supports 10-point multi-touch via a standard USB interface.

Is sound over HDMI supported?

Yes.

Is SPDIF and optical pass-through supported?

It is possible with our USB to SPDIF interface, available at the Hardkernel store, but it only works with Kodi (XBMC) on Ubuntu. The Android platform does not yet support the 5.1 channel pass-through.

Is there any analog audio output or input?

Not on the board itself, but you can use our USB Audio Adapter for analog audio.

Is there an I2S port to connect a HiFi DAC?

No, but an ODROID Hi-Fi Shield may be used with the ODROID-C1+.

What is an eMMC module?

The eMMC module is a NAND flash-based storage IC which is mounted on a custom PCB for easier upgrade/replacement, and is much faster than SD card. The transfer rate of a typical eMMC is approximately 65MB/sec, while a microSD UHS-1 card is approximately 30MB/sec.

Do you have a SATA port?

No, but you can use a SATA-to-USB bridge device.

Chapter 1

Which power adapter should I use?

The ODROID-C1 consumes less than 0.5A in most cases, but it can climb to 2A if many passive USB peripherals are attached directly to the main board. It is recommended to use the Hardkernel 5V/2A PSU or USB-to-DC Plug cable with a 5V/2A charger. Due to the limited power output from a computer's USB port, we suggest only powering the ODROID-C1 with a good quality 5V/2A PSU.

What operating system (OS) does it use?

We recommend Android and Ubuntu as our default distribution. The OS is stored on the SD-card / eMMC.

Which Android and Ubuntu version are included?

Android 4.4.x and Ubuntu 14.04, which both run on the Linux kernel 3.10 LTS.

Which OpenGL and OpenCL are included in Android and Ubuntu?

OpenGL ES 1.1 and 2.0 are included. OpenCL is not supported.

Where is the OpenGL ES SDK?

Refer to the ARM Mali Developers site at <http://bit.ly/1FRJEi0> for information on OpenGL ES.

Is the full source code open, and can I build it by myself?

Yes, the bootloader, kernel and OS platform source code are available. However, the GPU userland drivers are in binary format due to ARM's policies.

Do I need to patch for optional accessories that are purchased from the Hardkernel store?

No, they will work out-of-box with the kernels supplied with the operating system.

Does the device support networking?

Yes, there is 10/100/1000 RJ45 Ethernet port.

Is there WiFi?

It is available via an optional USB dongle.

I still have more questions!

You may ask questions in our user support forums at <http://forum.odroid.com>.

Getting Started

Given the introduction to the C1 Single Board Computer (SBC) development board and the fact that you have all of the necessary peripherals, you must be excited to get started with your C1 now. To be able to use the computer on the Internet, we first need an idea of how the C1 will fit into a home computing network.



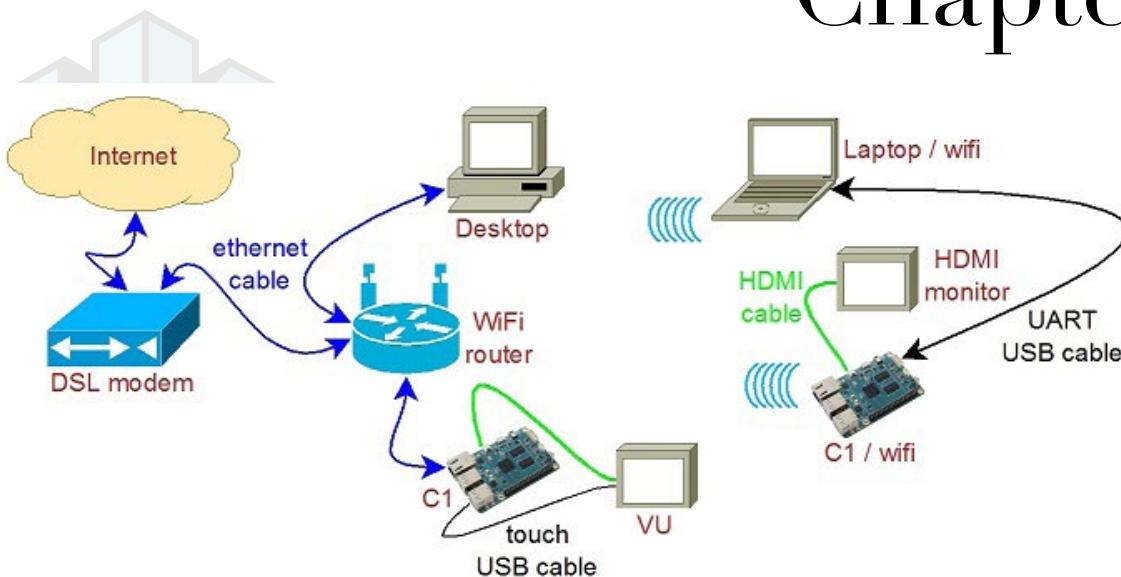
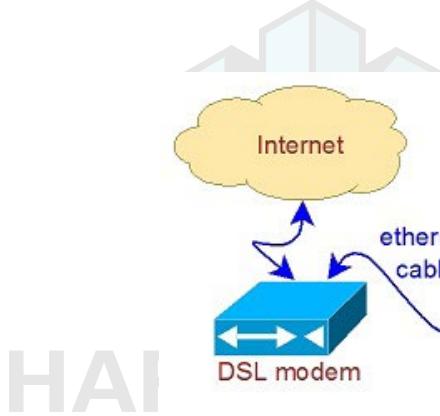
Basic C1 board

Home Computing Network

A typical simplistic home computing network environment includes the following, as shown in the diagram below:

- A Digital Subscriber Line (DSL) modem, typically provided by your Internet Service Provider (ISP), which connects your network to the Internet,
- A single (2.4GHz) / Dual (2.4 and 5GHz) band 802.11b/g/n WiFi router, with at least four (4) Gigabit Ethernet (gigE) ports and an additional Wide Area Network (WAN) port, where the WAN port connects to the DSL modem,
- An Intel/AMD based laptop (running MS Windows 7+ or Ubuntu 14.04 LTS+ Linux), that connects via wired ethernet or WiFi to the router, and
- At least one C1, that connects (wired ethernet or WiFi) to the router, which uses a High-Definition Multimedia Interface (HDMI) monitor (possibly touch-capable, such as a ODROID-VU).

Chapter 2



Typical home computing network

After preparing the boot media, it is theoretically possible to use a C1 in a network disconnected from the Internet. However, many use cases require an always-ON functioning secure connection to the Internet. In either case, it is presumed you have a fully functioning intranet (i.e., the internal network within the home). There are numerous online guides to setting up a secure network and as such, the setup of such a network is beyond the scope of this guide.

Preparing the Work Area

Your work area should be similar to the following:

- A well-lit and ventilated area,
- A flat-top desk spanning at least 2m x 1m with a height of 1m or appropriate to optimize posture and minimize overall fatigue,
- A freely available area of non-conductive surface of at least 1m x 1m, preferably covered by a well-grounded anti-static mat with you wearing the grounding wrist band/cable attached to mat. This safeguards the handling and placement of unprotected electronics such as C1 board, etc., on this area,
- A desktop with accessible monitor, keyboard/mouse or a laptop to prepare the boot-media for the C1 and debug the C1 bringup process,
- If needed, a 4-port USB3 (USB2 compatible) hub attached to the computer within the reach of the free area - for easy image flashing using a USB-based SD/eMMC module reader/writer and attaching the USB UART debug cable, and
- A well-grounded surge-protector with 6 or more outlets, within reach of the free area.

The above setup will go a long way in ensuring a safe workspace for you and your projects involving boards such as the C1.

Chapter 2

Flashing an Image

All SBCs require an operating system (OS) for booting. The operating system will be transferred to and reside on boot media, either built on to the board or attachable to the board. The process of placing the operating system on the boot media is termed flashing. Given this process, it is obvious that another computer system will be needed to flash the image on to the boot media, before the C1 can boot up.

The flexible C1 design allows you to use a boot media device that will be attached to it through an appropriate receptacles on the board. If you refer to the annotated board image, you can see that the C1 supports two boot media types - microSD card or an eMMC module. They can be obtained from Hardkernel directly, either with Linux or Android, pre-installed. If you purchased one such device and if it has a factory-installed image, you can skip the flashing steps in this section and proceed to the next section.

Boot Media

microSD/microSDHC card

- Slower, less longevity
- 8GB, 16GB, 32GB
- Linux, Android
- Requires USB SD card reader/writer



16GB Linux MicroSD card and USB reader/writer

eMMC version 5.0 module

- Faster, more longevity
- 8GB, 16GB, 32GB, 64GB
- Linux, Android
- Requires eMMC/microSD adapter
- Requires USB reader/writer



16GB Linux eMMC and adapter

§ = Available from Hardkernel

Chapter 2

Development boards such as the C1, by definition, will at some point either require an OS upgrade or a simple reflash to revert to a known good base state. The following section describes flashing the media.



USB microSD card adapter

No matter which boot media you select for your C1, you will need an additional device to perform the flashing process, which is called a USB SD card reader / writer. It should be compatible with your host desktop or laptop (flashing computer) and with the OS running on it. The image above shows a typical microSD card adapter. If you recall in the previous section, we recommended the placement of a USB3 hub close to your free work area. This hub, if present, is where you would connect the USB SD card reader / writer.

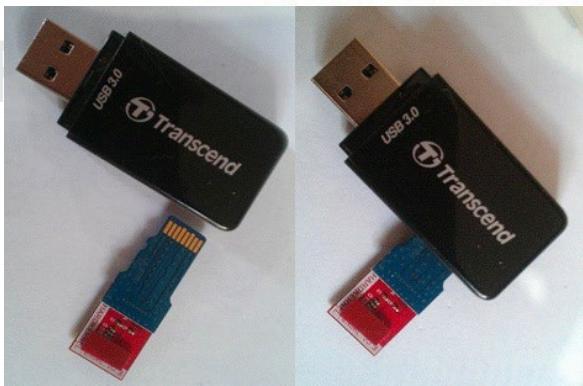
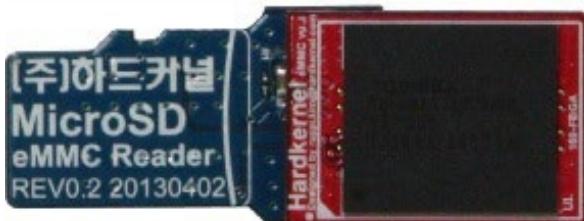
Now you may see the point of the suggested hub location - basically, it allows for frequent flashing of the boot media and its convenient attachment to the C1. If all points in the connection are USB3 type, you will get the fastest possible read / write speeds. USB2 compatibility will allow one or more points to be of type USB2, but will result in slower speeds.

Start using the designated non-conductive free work area for all subsequent activities. In the microSD receptacle of the USB SD card reader / writer, look for the metal contact pins. Then, if you are using a microSD card to flash the image, align the metal strips of the microSD card to make contact with the pins. Insert the microSD card into the USB SD card reader / writer. The image below indicates the alignment when using the Transcend USB3 microSD card reader / writer model. The alignment may be different with your reader / writer model.

However, if you are using an eMMC module to flash the image, you first need to attach the eMMC module to the eMMC/microSD USB adapter. Place the eMMC module and the adapter such that the 34 pin female/male connectors overlap and gently press them together.

Align the metal strips of the microUSB adapter to make contact with the pins in the microUSB receptacle of the USB SD card reader / writer. Insert the assembly by pressing firmly on the eMMC module until it clicks into place.

Chapter 2



Preparing the eMMC module

This is the alignment when using the Transcend USB3 microSD card reader / writer model. The alignment may be different with your own microSD USB adapter. Next, ensure that the computer you are about to use to flash the image, has a functioning connection to the Internet. Now that you have the USB SD card reader / writer with the boot media inserted, attach the male USB end of the USB SD card reader / writer to the appropriate (USB3 or USB2) female port on the computer/USB3-hub.

The ODROID-C1 is not able to boot immediately with some models of microSD cards. However, you can still use any brand or speed of memory card as storage. Sandisk Class 10 or UHS-1 cards are working well with ODROID-C1, and below is a list of the memory cards that have been tested:

Fully compatible microSD cards

- SanDisk microSDHC Memory Card
- SanDisk Ultra microSDHC/microSDXC UHS-I Memory Card
- Samsung SDXC EVO Memory Card
- Toshiba Exceria Memory Card

Partially compatible microSD cards

The following cards are not able to boot immediately on ODROID-C1, but instead take about 30 seconds to start the boot up process. Once the boot processing is done, and the operating system is running, they perform normally:

- Transcend microSDXC Class 10 UHS-I 300x (Premium)

Chapter 2

- Transcend microSDHC Class 10 (Premium)
- Transcend microSDHC Class 6 (Standard)
- Samsung microSD Pro Class 10
- Samsung microSDHC EVO Class 10
- Kingston microSD Class 4
- PNY Turbo Performance Class 10

Class 2/4/6 microSD cards have not been tested with the ODROID-C1, therefore it is likely that the ODROID-C1 will not boot properly with them. We strongly recommend to use Class 10/UHS-I microSD cards, and a SanDisk microSD for fully compatibility.

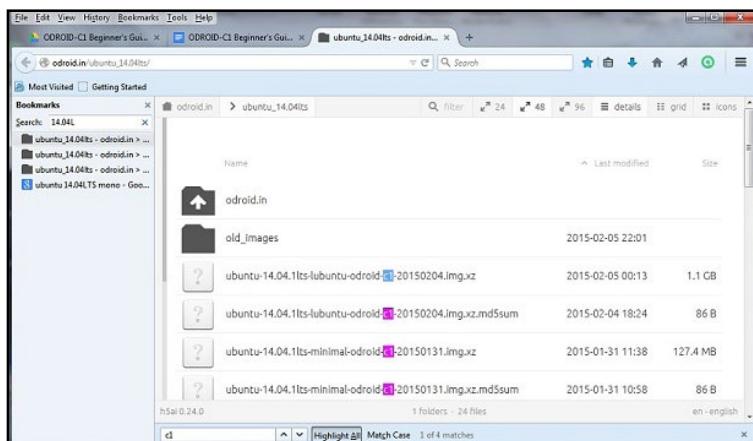
If your board doesn't boot after approximately 30 seconds, connect the microSD card to a host computer using a USB adapter, and edit the boot.ini file. Comment out the line that pertains to disabling UHS cards to match the one below, then try the boot process again:

```
# setenv disableuhc "disableuhc"
```

Flashing an Image on a Microsoft Windows 7+ Computer

As soon as you attach the USB SD card reader / writer for the first time, the Windows OS will sense the presence of a new USB device through its plug-and-play (PNP) infrastructure. It will obtain the device identifiers from the device and search through its local database. If found, it will install it for the first time. If not found, it will request you to provide a location for the driver or seek to find it over the Internet. Once the driver is obtained for the first time, it will install it and request permission to reboot the system, which should be allowed. Once installed, it will use the driver on all subsequent sessions.

After the system has booted, launch a web-browser such as fire-fx and navigate to http://odroid.in/ubuntu_14.04lts/, which will list all of the available Ubuntu images. Click on the Last Modified column header to sort the list by descending order of date. Search for



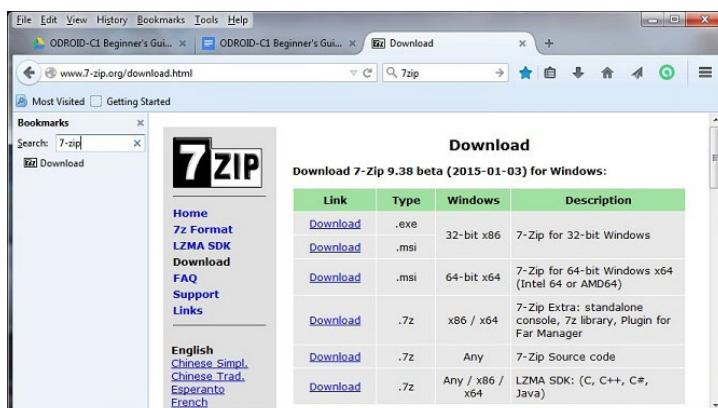
List of Available Ubuntu Images for the ODROID-C1

Chapter 2

images with “C1” in their names. You will see a highlighted list as shown in the screenshot above.

Each of the entries in the list is also a link. Click on the link that corresponds to the latest relevant C1 image and download it. You will need to uncompress this file before flashing it to the boot media. To do so, you will need a specialized utility to uncompress the downloaded file. A reliable utility to uncompress such files is the 7-zip program. Launch a web-browser such as firefox and navigate to <http://www.7-zip.org/download.html>, which will list all of the available installation files for this utility. Click on a link that corresponds to your host computer’s OS. Details of your OS can be found by running either the msinfo32.exe or systeminfo.exe command in a command window.

The Windows OS used for this guide is a 64-bit type. Click the download link that corresponds to the 64-bit Windows OS, in this case, 7z938-x64.msi.



List of 7-zip installation files

After the file is downloaded, run it and select the default options, and wait for 7-zip to finish installing. Next, launch the Windows Explorer application and browse the download directory. Select the downloaded compressed image file and right-click the file with the mouse. Select the 7-zip menu option and under that select the Extract Here option. After the uncompress process, it will result in the image file that can be flashed. In this example, it will be ubuntu-14.04.1lts-lubuntu-odroid-c1-20150204.img.

Another essential utility is one that will be used to flash the un-compressed image file, such as the Win32DiskImager utility. Hard-kernel has an updated version of it that adds a useful step to the flash process, which clears the boot media prior to writing the image. This version of Win32DiskImager.exe can be downloaded from the following URL:

http://dn.odroid.com/DiskImager_ODROID/Win32DiskImager-odroid-v1.3.zip

Chapter 2

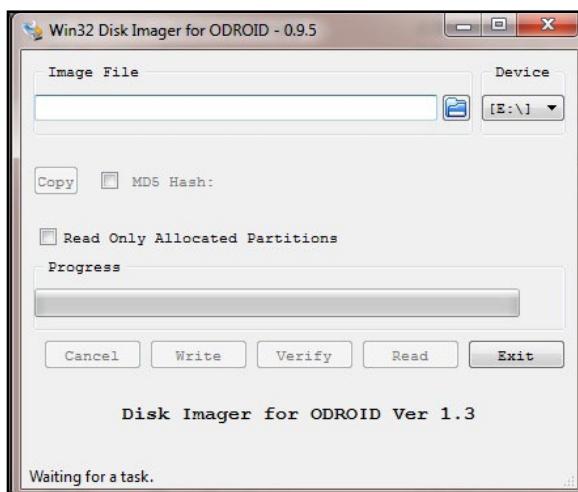
Uncompress the compressed image file using the 7-zip application, which generates the directory win32diskimager2-binary that includes the Win32DiskImager2.exe application. Move the entire directory to C:\Program Files (x86):

C:\Program Files (x86)\win32diskimager2-binary\

You will need to be logged in to Windows OS as an administrator to move this directory. Then, create a shortcut for the application on the desktop using the following link:

C:\Program Files (x86)\win32diskimager2-binary\Win32DiskImager2.exe

Next, change the properties of the shortcut (right-click of mouse) so it will run with Administrator privileges. Finally, double-click this shortcut in order to launch the Win32DiskImager2.exe application. Click the folder (directory) icon to browse and select the directory that has the uncompressed image file.



Flashing an image using Win32DiskImager2

Select the uncompressed image file. Note that the application automatically selects the device (G:) that corresponds to the boot media that was placed in the USB SD card reader / writer. The screenshot above shows the Win32DiskImager2 interface. Click on the Write button to start the flash process.

You will be notified of the completion of the flash process. Launch the Windows Explorer application and right-click the device that was used in the flash process by Win32DiskImager2.exe with the mouse and select the Eject option. When permitted, remove the USB SD card reader / writer from the USB port of the computer. Remove the boot media and place it on the non-conductive surface.

Chapter 2

The above steps are to flash the Linux image. What if you wish to flash an Android image? While you would download a different compressed file, all subsequent steps would be that same as those listed above.

You would need to launch the browser once again and navigate to <http://bit.ly/1FRHJK1>, look for the latest Android image and click the link to download it. The screenshot below shows the webpage that lists the Android images.



List of Android images for the ODROID-C1

Once the compressed file has been downloaded, uncompress it and flash it, using the steps listed earlier in this section.

Flashing an Image on a Linux Ubuntu 14.04 LTS Computer

In your Ubuntu Linux desktop, launch a terminal window. Create a working directory:

```
$ cd ~ && mkdir linux-img && cd linux-img
```

You can download the compressed image using a browser like Firefox for Ubuntu using the same (Windows OS) steps described earlier. If you follow this method, you should use the mv command to move the downloaded compressed file to the working dir listed above. Instead, if you wish, you can use a Linux utility like wget right from within the working directory:

```
$ wget http://odroid.in/ubuntu_14.04lts/ubuntu-14.04.1lts-lubuntu-odroid-c1-20150204.img.xz
```

Wait for the download process to complete, then uncompress the file using the following command:

```
$ unxz ./ubuntu-14.04.1lts-lubuntu-odroid-c1-20150204.img.xz
```

Chapter 2

This will result in an uncompressed image file called `ubuntu-14.04.1lts-lubuntu-odroid-c1-20150204.img`. Attach the USB SD card reader / writer with the boot media into an appropriate USB port. After a few moments, check for an entry that corresponds to the USB SD card reader / writer (Transcend model in this case) using the following commands:

```
$ cd /dev/disk/by-id/  
$ ls *Transcend*  
usb-TS-RDF5_SD_Transcend_00000000039-0
```

This tells us that the write destination path to the boot media is `/dev/disk/by-id/usb-TS-RDF5_SD_Transcend_00000000039-0`. Given this information, we can carefully construct the image copy command using appropriate input and output paths:

```
$ cd ~/linux-img  
$ sudo dd \  
if=~/ubuntu-14.04.1lts-lubuntu-odroid-c1-20150204.img \  
of=/dev/disk/by-id/usb-TS-RDF5_SD_Transcend_00000000039-0 \  
bs=1M conv=fsync  
$ sync
```

The `dd` command is very powerful, so use it with a lot of care. If incorrect parameters (especially the `of` parameter) are used here, you could potentially ruin the OS installation of the computer. The `dd` command is hence often referred to as the disk destroyer command.

This step could take a while. So, wait for it to complete before proceeding. Once completed, remove the USB SD card reader / writer from the USB port of the computer. Remove the boot media and place it on the non-conductive surface.

The above steps are to flash the Linux image. To flash an Android image, simply download a different compressed file, and all subsequent steps would be that same as those listed above.

You would need to launch the browser once again and navigate to <http://bit.ly/1FRHJKI>. Look for the latest Android image and click the corresponding Self-installation image link. In the example case here, we would be downloading the image called `selfinstall-odroidc-eng-s805_4.4.2_dev_master-195-v1.2.img.xz`.

If you wish to use the `wget` command instead, you can follow the steps listed earlier but using the appropriate Android URL. Once the compressed file has been downloaded, uncompress it and flash it, using the steps listed earlier in this section.

Chapter 2

Flashing an Image on an OSX Computer

The procedure for creating an image file using OSX is similar to Linux, with some small differences. First, download and install the `xzutils` package from <http://tukaani.org/xz/>, making sure to select the OSX binaries, and use it to uncompress the image:

```
$ xz -d selfinstall-odroidc-eng-s805_4.4.2_dev_master-195-v1.2.img.xz
```

Other differences are that the block size (`bs`) parameter for the `dd` command is in lowercase, and the USB adapter's device name is in the format `/dev/diskX`. Before writing the image to SD Card or eMMC module using the “`dd`” utility, run the command “`diskutil`” to determine the device name, then unmount it:

```
$ diskutil list  
$ diskutil unmountdisk /dev/diskX  
$ sudo dd if=/dev/diskX bs=1m of=./selfinstall-odroidc-eng-s805_4.4.2_dev_master-195-v1.2.img
```

Inserting the eMMC Module or SD Card

Now that the desired image has been flashed on to the boot media, it's time to insert the boot media into the appropriate port on the C1.

Ensure that the work area is clear of unnecessary items. Wear the grounded wristband. Touch a metallic surface, like that of your desktop. Then, open the C1 packaging box. Place the anti-static bag containing the C1 board, on to the non-conductive work surface. Open the bag and remove the board, holding the edges. Avoid touching any of the components, peripheral connectors, ports, exposed circuits or on board solder. Place the board on the work surface, with the bottom side up. Check the annotated diagrams and follow the instructions from Chapter 1 for inserting the eMMC module or SD card.

Powering Up

Before powering the board, attach all peripherals as described in Chapter 1. Insert the power cable, which will be indicated by the red LED lighting up, and wait for the image to boot to desktop. It will take anywhere from 20 seconds to 2 minutes for the desktop to appear, depending on the operating system and the type of boot media used.

Troubleshooting

Red LED not glowing

If the red LED is not glowing, power is not being supplied to the board. Check the connections, and verify that the plug is inserted

Chapter 2

properly. Also, make sure that a 5V/2A power supply is being used, and that it is plugged in.

Blue LED not flashing or glowing

If the blue LED is not flashing or glowing at all, something went wrong with the image writing or the card is not correctly inserted. Try to download, extract and write the image again. If that doesn't solve the issue, try downloading and flashing a different image.

Blue LED solid glowing

The C1 has stopped or frozen due to a kernel panic or software bug and should be rebooted.

Mouse/Keyboard not working

Ensure that it's correctly plugged in, or try a different USB port. You can change it while the C1 is running.

No desktop image

Be patient, since some SD cards require several minutes to boot and anywhere from 30 seconds to 2 minutes to show the first image. Refer to the microSD card compatibility list above. Verify that the HDMI cable is correctly plugged in, and edit the boot.ini on a PC and select the correct resolution (see Chapter 1 for details).

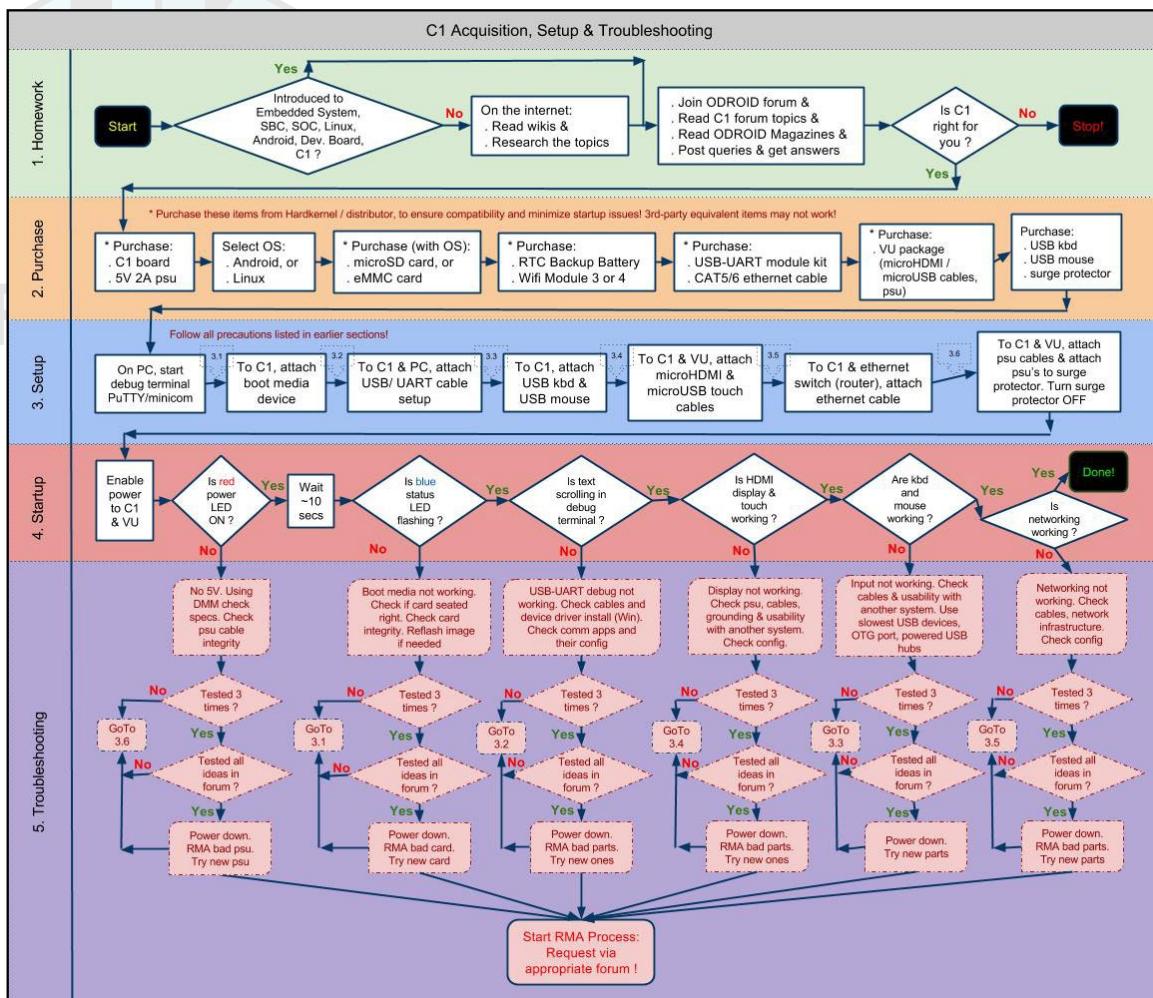
Some HDMI cables have compatibility issues, so make sure to use an official Hardkernel HDMI cable. Some monitors have non-standard EDID functions. In this case, you need to select the non-EDID mode in the ODROID configuration utility.

Some cables have a shorter plug, and it may be necessary to remove the plastic case for a better connection. If there is a USB host port on the monitor, connect a microUSB cable from the ODROID to the monitor, which creates a ground path between the TV and the device. Add a short jumper wire between the outer casing of the microHDMI and microUSB ports to create a ground path.

On the next page the series of steps that should be followed in order to troubleshoot issues is outlined in detail. Before requesting an RMA, research the relevant forums and verify that a solution or work-around doesn't already exist for your issue.

If your issue has not been previously encountered by anyone, create a new issue and post it on the forums at <http://forum.odroid.com>, then carefully read any feedback and follow the recommendations. Provide as many details as possible, indicating not only your software and hardware environment, but also the exact steps to recreate the issue.

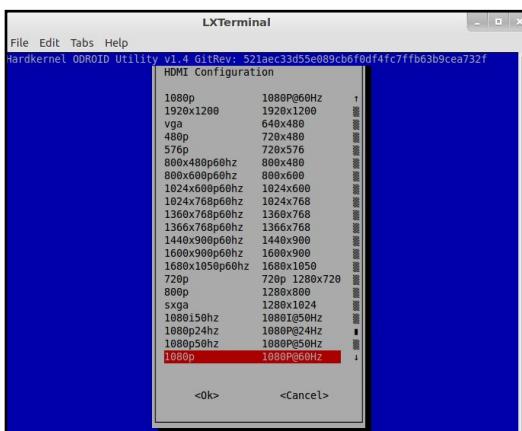
Chapter 2



Acquisition, setup and troubleshooting

Configuring Linux

After booting the ODROID to a Linux desktop, run the included ODROID Utility, which is linked on the desktop, and expand the file system to have full use of the microSD card or eMMC module. Then, run the ODROID-Utility again to update the system and reboot.



Configuring HDMI resolution with the Linux ODROID Utility app

Chapter 2

After the system has been updated, select the desired HDMI resolution by running the ODROID Utility again:

- 1920x1200 (*WUXGA*)
- 1920x1080 (*1080p*)
- 1280x720 (*720p*)
- 720x480 (*480p*)
- 720x576 (*576p*)
- 1280x800 (*800p for ODROID-VU*)
- 1280x1024 (*SXGA*)
- 800x480 (*WVGA*)
- 640x480 (*VGA*)

The SSH daemon is enabled by default in the Ubuntu template, and the username:password is odroid:odroid. The root password is also odroid.

Finally, the language and timezone settings may be configured with the following command, after installing the desired language in the “Language Support” settings application:

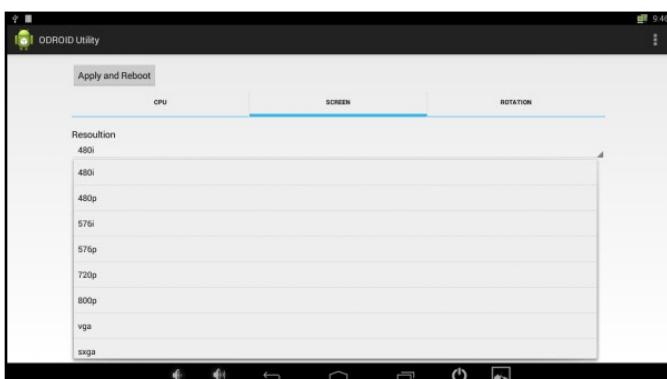
```
$ sudo dpkg-reconfigure locales
```

After the command completes, drag the selected language to the top of the list in the “Language Support” settings. The timezone may be selected by typing the following command:

```
$ sudo dpkg-reconfigure tzdata
```

Configuring HDMI Resolution in Android

Once the Android desktop has loaded, run the ODROID Utility app, and select the desired CPU frequency, monitor resolution and orientation.



Configuring HDMI resolution with the Android ODROID Utility app

Chapter 2

To set the timezone and other configurable options, use the built-in Settings panel that comes with the Android installation.

Powering Down

Shutting down the ODROID-C1 is very important in order to prevent damage to the microSD card or eMMC module. In Android, powering down is done by selecting the power button icon at the bottom of the desktop. In Linux, powering down may be done by either selecting the shutdown option from the Applications menu, or by typing the following into a Terminal window:

```
$ sudo shutdown -h now
```

Wait until the blue LED goes out, which indicates that all system activity has completed. It is now safe to unplug the ODROID-C1 power supply, remove the boot media, and detach the peripherals.

Operating Systems

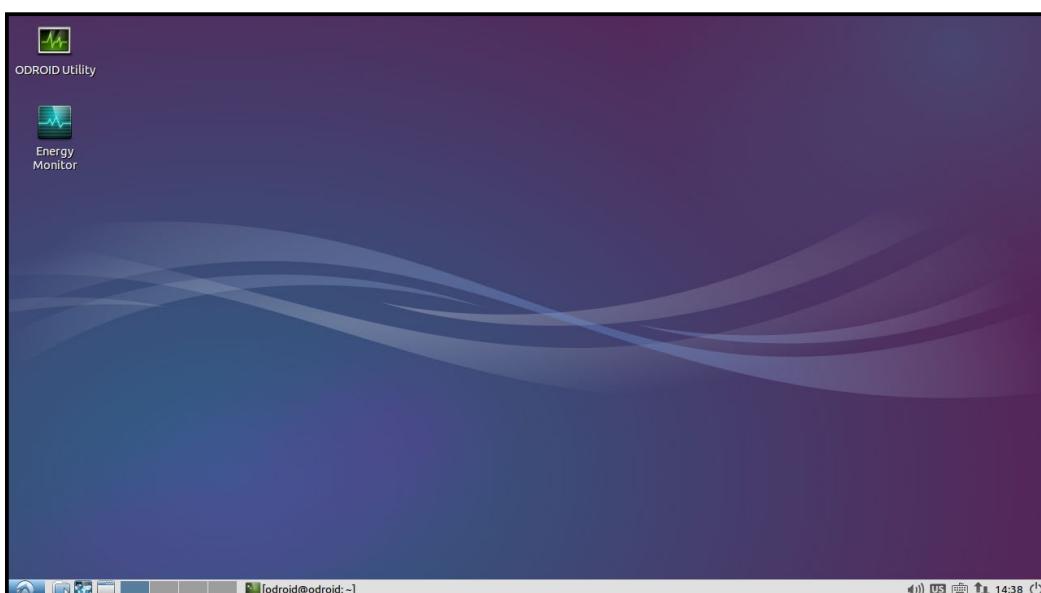
The ODROID-C1 can run many free, full-featured, Linux-based, open-source operating systems. Two of the most popular, Ubuntu and Android, are available for download from Hardkernel's website, while many other flavors of Linux may be compiled from source, such as ARCH Linux, FreeBSD, Fedora, CentOS, OpenSUSE, Slackware, and Mint. All of them have a common customized kernel which allows the operating system to communicate with the ODROID hardware. Hardkernel publishes kernels that are specific to the ODROID architecture, and maintains a repository where they may be downloaded and installed as improvements are made.

Ubuntu

Ubuntu is based on the Debian operating system, and both offer a desktop as well as a large library of applications that may be freely installed, used, and modified. The main GUI tool for obtaining applications on Ubuntu is Synaptic Package Manager, although the command line tool called apt-get is often used by advanced users and scripts in order to streamline the installation process. Applications are stored on servers called repositories, which allow you to receive updates and new versions of software automatically.

Booting Up

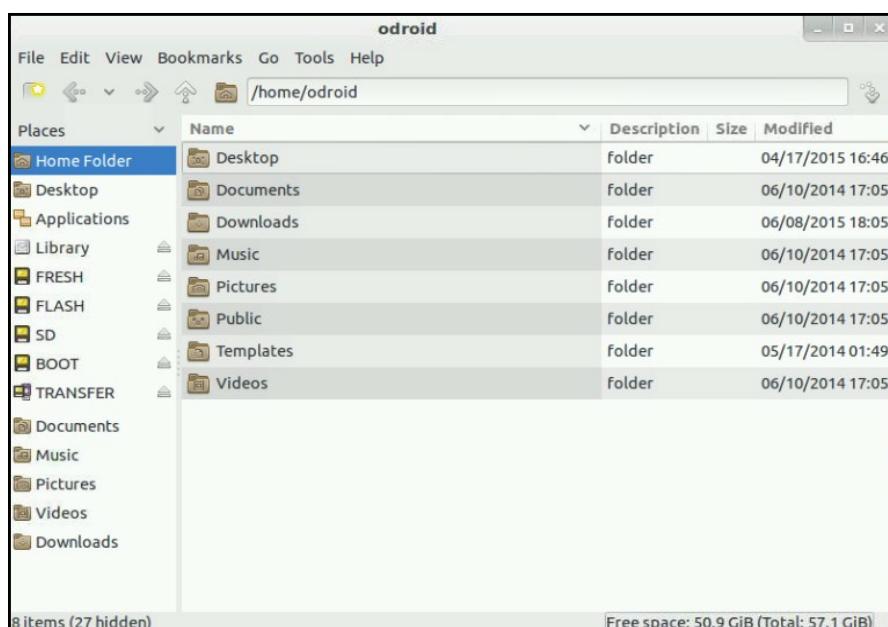
Ubuntu can be ordered as a pre-installation on the eMMC or microSD card that comes with the ODROID-C1. Please refer to Chapter 2 for instructions on downloading and flashing Ubuntu onto your C1's boot media. Once the C1 has been powered on, it will boot to a desktop similar to the one shown in the following image.



Chapter 3

Linux Basics

Ubuntu and Debian, along with most Linux distributions, have a home folder structure where documents, files and media may be stored for each user of the system. For example, downloaded files from the Internet are automatically saved to the Downloads folder, and the other folders may be used to organize various public and private files such as pictures, video, and word processing documents. There are also many special folders that are normally hidden from view that start with a dot (.), where settings for various applications are stored. To view the Home folder, start the File Manager application from the Applications menu in the bottom left corner of the screen, which will look like the screenshot below.



Both Ubuntu and Debian offer many different types of desktop environments, such as Xubuntu, LXDE (Lubuntu), Gnome, KDE (Kubuntu), and Blackbox, to name a few. Each one includes specific applications that are useful for different types of users. For instance, a typical desktop user who wishes to browse the Internet, use LibreOffice to create, edit and read documents, and listen to music or watch videos might choose Lubuntu, Kubuntu, or Xubuntu,



Chapter 3

since they include a robust Applications menu and Software Center for downloading new applications, as shown in the above image.

If the ODROID-C1 is being used as part of robotics project, a lightweight environment such as Blackbox may be more suitable, in order to free up the amount of memory available for the main controller application. Other specialized environments are also available from Synaptic Package Manager, including Ubuntu Studio, which includes many applications that are primarily used for creating and producing music, videos, artwork, and photography.

Kernel

At the heart of the Linux system is the kernel, which is responsible for allowing the desktop environment to communicate with the hardware through a common interface. Each ODROID has its own customized kernel code that is modified, tested and published by Hardkernel. Programmers may also download the source code from the official repository at <http://github.com/hardkernel> and make any modifications or contributions that they wish.

The Linux kernel is compatible with all versions of Linux, so any operating system that has been compiled for the ARM hard-float architecture (ARMHF) may be converted to run on the ODROID by installing an ODROID-C1 kernel and modifying the boot partition. This makes Linux one of the most versatile operating systems available, since it can be configured to run on almost any device, including laptops, desktop PCs, smartphones, and ODROID microcomputers.

Graphical User Interface (GUI)

All versions of Linux offer a command line interface (CLI), and sometimes include a window manager and desktop environment that is launched on startup. Most modern Linux systems such as Ubuntu use a library called X11 to create a windowing environment and provide graphics libraries so that users may interact with applications using a mouse. The ODROID-C1 includes a Mali 450 Graphics Processor Unit (GPU) which is controlled by X11, in conjunction with the Open Graphics Library (OpenGL), in order to render graphics on a 720p or 1080p monitor. ODROIDs use a subset of the popular OpenGL library called OpenGL ES, which is specifically designed to work with ARM processors, especially smartphones. Applications that are written for OpenGL ES can use low-level graphics functions on the GPU chip itself in order to quickly and efficiently render graphics, resulting in a much faster and smoother user experience than using the CPU alone.

720p vs 1080p

ODROIDs support both 720p and 1080p monitor configurations,

Chapter 3

which are video resolutions commonly supported on LCD TV's and on many LCD computer monitors. 720p is sometimes referred to as HD, and 1080p as Full HD. The following comparison table lists the native display resolutions along with some sources that use these resolutions.

Standard	Resolution	Aspect Ratio	Refresh Rate	Typical Sources
1080p	1920 x 1080	16:9(1.78:1)	24Hz, 50Hz, 60Hz	HDTV, Blu-ray
1080i	1920 x 1080	16:9(1.78:1)	50Hz, 60Hz	HDTV, Blu-ray
720p	1280 x 720	16:9(1.78:1)	50Hz, 60Hz	HDTV, miniDV

As a comparison, the standard DVD Video format uses a display resolution of 720 x 480 with a 4:3 aspect ratio (non-widescreen). This resolution is also referred to as D-1 video and is the standard used or miniDV digital camcorders.

Progressive vs Interlaced Video

1080p and 720p are progressive video formats. For these formats, a complete video image is transferred for each frame of the video signal, so that a 60fps video displays 60 complete, full frames per second. This format is visually superior to interlaced video, and in general should be used when supported by your display.

1080i and 720i are interlaced video formats, and video interlacing is the historical standard for broadcast video. In interlacing, two frames (or more correctly, fields) are transferred for each frame of the video signal, so that a 60fps video has 30 complete, full frames per second. The display device sends the two fields to the screen in the horizontal odd and even (or interlaced) pixel positions. Given the visual persistence of the human eye, this reconstruction of full frames is undetectable to the viewer.

In general, faster frame rates result in a visually smoother video playback, and interlacing results in a halving of the native frame/field transfer rate because two fields are required for the full image. Frame rate and visual experience are subjective, however. Traditionally, movies produced with conventional film use a 24 frames per second rate, and many viewers prefer the cinematic experience that 24fps video provides. In fact, much of the professional video produced with high-end gear is actually shot at 24fps, and later converted to video frame rates to provide the look and feel of cinematic film.

Video Downconversion

When selecting a screen resolution for your C1, you should first consult the specifications for the display that you plan to use. In general, using the native resolution of your display will provide the best performance and quality.

Chapter 3

For example, if you use a display that only supports native 720p, and you set your C1 to a 1080p screen resolution, it will probably work, but is not the ideal configuration. The mismatch in resolutions indicates that your display is automatically performing a downconversion of the incoming signal. Despite appearing to work OK, the C1 will be performing unnecessary work in order to create the 1080p output.

If a 720p video is viewed with this configuration, the C1 will upconvert the original 720p video by extrapolation to produce a 1080p output, only to have that 1080p video downconverted by interpolation back into 720p by the video display. In this case, setting the C1 screen resolution to 720p is the logical choice.

Video Upconversion

When the C1 is set to a higher output resolution than the source video, an upconversion is performed by extrapolation to produce the higher resolution frame in the framebuffer. The average video player can display videos with a number of different source resolutions. These videos are upconverted by the C1 in order to fit the screen, and the user therefore has a uniform viewing experience.

However, if the C1 is used solely for such purposes as video editing, video effects, or historical restoration, and the source material resolution is lower than the maximum resolution of the video display, the user may wish to setup the video settings differently. In this case, matching both the C1 video settings and the video display settings to the source material's native format would be appropriate. With this configuration, the video playback would provide an image that is true to the original source material, as well as freeing up more of the C1's CPU bandwidth for other processing tasks.

HDMI Overscan

If the display used on your C1 shows a slight cropping of the visible image on the screen, you may be experiencing overscan. This is not an uncommon problem, and especially so for LCD TV monitors. The fix is usually a simple one, and the underlying issue is most likely due to a setting with the LCD monitor. Some PC Monitors with HDMI inputs will also apply overscan to the HDMI input, assuming that a broadcast TV signal is being used.

Why Do Monitors Have Overscan?

Monitors that are used for broadcast television usually have overscan enabled by default. This is a normal feature of TV monitors and has been present from the very beginning of television. Overscan is used to crop the edges of the video frame in order to remove any erratic or distorted edges that often exist with broadcast video. To the

Chapter 3

viewer, this results in a cleaner picture, and the overscan simply isn't noticed. For a computer display however, this can be an issue. For this reason, computer LCD monitors usually do not provide overscan, and if they do have this feature, it is disabled by default.

Disabling Monitor Overscan

Prior to attempting to fix overscan using the methods detailed below, first verify that the monitor resolution setting and the C1 screen resolution setting match. Unfortunately, there isn't a standard method or terminology for disabling overscan used by monitor manufacturers, so you may have to search through the monitor menus a bit until you find the overscan setting.

Display Setting Button on Remote Control

Using the LCD monitor remote control, first look for a display mode button labeled Display, Screen Mode, marked with a display icon, or marked with |<>| (for wide mode). If your remote has one of these dedicated display mode keys, pressing this key should allow you to cycle through the display modes.

Finding Setting in Monitor Menus

If you cannot find a display mode button, you will need to enter the display's menu setting mode, usually marked Menu or Settings. You will need to search for the menu item that controls the overscan setting, and on some displays you will need to enable the Advanced mode.

The table below describes several possible menu locations for the overscan setting on different displays:

Menu > Picture Mode > Aspect ratio
Change to "Just Scan"
Menu > Picture > Screen adjustment > Picture Size
Change to "Screen Fit"
Menu > View Mode
Change to "Dot by Dot"
Menu > Tools
Change HDMI Source to "PC"
Menu > Picture > Screen adjustment > Picture Size Screen
Auto Config
Menu > AV Preset > HDMI
Change to "PC"

As you can see, finding the setting for your LCD monitor may take some time to locate. Other terms manufacturers are known to use for overscan disable include HD size, full pixel, unscaled, native, and 1:1.

Chapter 3

On some monitor models, one of the HDMI inputs is intended to work with a PC and will disable overscan for that input (often HDMI 2). You can also look at the monitor's HDMI connectors for a label such as PC Input, or read the monitor users manual to find the correct input to use if applicable.

If you have used one of the above methods to change your LCD monitor's display mode and it does not help, or if your monitor does not provide an overscan adjustment, use one of the following methods to adjust the C1 video output directly.

Adjusting Linux Display Output

Installing xrandr and using the transform command can help fix an overscan issue inUbuntu. Here is a typical adjustment using xrandr, which should be typed into a Terminal window:

```
$ xrandr --output HDMI-1 \
--transform 1.05, 0, -35, 0, 1.05, -19, 0, 0, 1
```

Command Line Interface

The Linux terminal CLI is the basic method of interacting with the Linux system, normally using a shell called BASH. When a graphical system is running, a Terminal window may be launched in order to issue commands using the keyboard. Many Linux systems also come with the Secure Shell (SSH) server, which allows a command line interface to be invoked from a remote computer via Ethernet. Other protocols, such as Virtual Network Computing (VNC) have the ability to project the graphical environment to a remote computer as well.

BASH is a powerful scripting language as well as a means of reading, creating, modifying, and launching files and programs that are stored in the Linux file system. A collection of BASH commands, called a script, can be saved to a file with the extension ".sh" and used as an executable file. A shell script can perform many tasks, such as configuring program variables, launching an application, and copying or moving files.

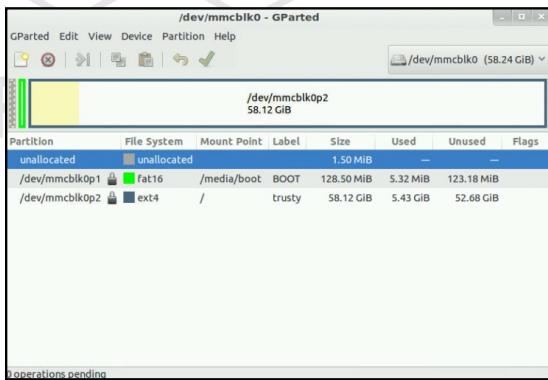
Disk Partitions

Linux can read and write to a variety of disk formats, with the most common called EXT3, EXT4, and FAT. If disk compatibility with Windows and OSX is a priority, it is recommended to format external drives as FAT32 which can be read by nearly all operating systems. The root filesystem of Linux is usually in EXT3 or EXT4 format, which can only be read by other Linux systems.

Disk partitions may be easily resized in order to use all of the available space on a disk using a utility called Gparted. The image

Chapter 3

below shows the Gparted application being used to resize a root partition. It is recommended to use the ODROID Utility to resize the main partition in Ubuntu in order to have as much storage space as possible on the microSD card or eMMC module.



Web Browsing

One of the main uses of modern computers is to browse the World Wide Web, and the quad-core ODROID-C1 delivers a smooth, enjoyable browsing experience. Using the Ubuntu Software Center, a browser such as Firefox and/or Chromium may be installed, along with open-source versions like Iceweasel.

When watching streaming videos, it is recommended to use the Kodi/XBMC YouTube plugin in order to launch videos in full-screen, since mainstream browsers are not specifically written to use the video decoding chip available on ODROIDs. However, the ODROID-C1 CPU is powerful enough to play standard videos within a browser window at normal size using software rendering.

Kodi (formerly XBMC)

Kodi is an application which can turn an ODROID-C1 into an amazingly powerful home media center. In fact, using the C1 to run Kodi is the reason why many, if not most people purchased their C1 in the first place. If you're not familiar with Kodi, here are links to the official Kodi Intro FAQ at <http://bit.ly/1G8wDjD> and the Kodi Wikipedia page at <http://bit.ly/1Ir2z3R>.

The current versions of Ubuntu Linux and Android provided by Hardkernel already have Kodi installed. However, you may wish to re-install or upgrade Kodi in the future when a new release becomes available. Pre-release (beta) versions of Kodi are made available for testing, and you can join a group of users that test out new features before the formal release.

Troubleshooting

If you experience problems with playback of some video formats in Kodi, the first troubleshooting step would be to set video acceler-

Chapter 3

ation to “software” by selecting the System menu, the pressing **Settings >Video >Acceleration >Decoding Method >Software**.

Kodi Upgrade

If you are installing a newer version of Kodi, you do not need to uninstall the current version first. However you should create a Kodi backup just to be safe, which is covered later in this section.

Kodi Installation

To install Kodi, enter the following commands into Terminal:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com \
--recv-keys AB19BAC9
$ sudo echo "deb http://deb.odroid.in/c1/ trusty main" > \
/etc/apt/sources.list.d/odroid.list
$ sudo echo "deb http://deb.odroid.in/ trusty main" >> \
/etc/apt/sources.list.d/odroid.list
$ sudo apt-get update && sudo apt-get dist-upgrade
$ sudo apt-get install kodi
```

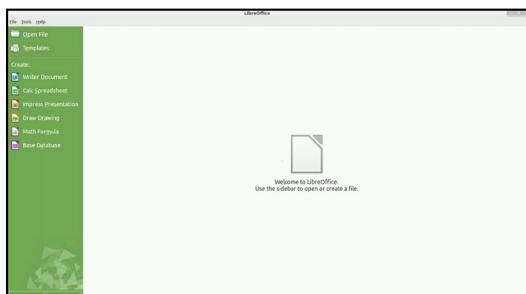
Kodi Backup

Backing up your data is always a good idea, and Kodi is no exception. Kodi configurations and databases can become quite detailed, but backup and restoration is quite simple. You can perform full or partial backups/restores, as well as copy your complete Kodi setup to another system - even to your PC or laptop.

In order to perform a Kodi backup or restore, you will first need to install the Kodi “Backup AddOn”. More information may be found at <http://bit.ly/1JER8XL>, along with details on performing Kodi backups at <http://bit.ly/1KRNF6K>, and on the Kodi forum at <http://bit.ly/1Q-GOfGF>.

Office and Productivity Applications

LibreOffice is a popular and powerful office productivity suite that includes word processing, spreadsheets, presentations, drawing, and flowcharting applications. It is a free alternative to Microsoft Office, and can be installed on Ubuntu via the Ubuntu Software Center or Synaptic Package Manager. The loading screen of LibreOffice is shown below.



Chapter 3

Music and MIDI

Not long after the first personal computers became available, creative individuals began to find inventive ways to use them for music composition and performance. Before long, the need for I/O standardization became clear, and in 1983 the MIDI specification was developed. MIDI (Musical Instrument Digital Interface) gained rapid industry acceptance, and within a few years, a MIDI Interface was found on virtually every sound card and on most PCs with sound chips on the motherboard. This is no longer the case, but the MIDI interface, MIDI instruments, and external MIDI synthesizers are still widely used by musicians and sound engineers. In fact, MIDI is still the de-facto standard instrument interface, and does not show any signs of falling into obscurity as have many early computing standards.

What is MIDI?

MIDI encompasses not only the MIDI connector hardware specification, but also includes the communication protocol used for instruments and synthesizers, and a MIDI file format specification for MIDI recording and playback. The standard MPU-401 physical MIDI interface is a 5-pin DIN connector. Internal sound cards no longer include this connector, but it can be found on some high-end external USB sound modules. However USB-to-MIDI adapters are inexpensive and widely available. The MIDI Manufacturers Associated website at <http://bit.ly/1Gn2cYw> is a good source for additional information.

Linux Support for Music and MIDI

MIDI adapters, USB Sound Devices, and Music Applications have wide support on Linux. For example, the Ubuntu Studio project is an excellent open-source Linux build that has extensive device support with pre-configured applications. The official Ubuntu Studio site page has an <http://bit.ly/1FFKyvh> which will give you an idea of the range of Music and MIDI applications possible on Linux.

Ubuntu Studio includes some of the most popular audio apps available - including tools for DAW (Digital Audio Workstations) for multi-track mixing, sequencers for MIDI music, drum machines, software synthesizers, and even music creation via programming. Ubuntu Studio is available from the Ubuntu Software Center and Synaptic Package Manager.

How do I add a MIDI Interface to the C1 ?

Three basic options for MIDI are available for the C1:

USB sound module: If you wish to use a USB sound module, look for a device that has solid Linux and/or Android device driver support

Chapter 3

which includes the MPU-401 interface.

USB-to-MIDI-Adapter: A number of basic devices are available which have been proven to work on both Android and Linux. Compatible devices will be advertised as a “no device driver required” model.

Direct UART to MIDI interface: Since MIDI is a asynchronous serial interface, standard serial ports can often be easily converted to a MIDI interface.

Sound Cards and Devices for MIDI File Playback on the C1

Not all sound cards and modules provide native MIDI file playback capability, including the ODROID USB sound card. Many other USB sound cards do however, so make sure to consult the device’s specifications. Professional-level USB MIDI synthesizers are also available, including sampling synths by Roland, Ensoniq, Kurzweil, Roland, and Korg.

Experimental Music with the C1

Using an ODROID-C1 for computer music opens up possibilities beyond those of traditional computer music performance, productions, and engineering. A MIDI Interface and any number of sensors could be used to create unique musical instruments. For example, a pressure transducer could be used as a MIDI breath controller, ultrasonic transducers could be used to make a Vulcan Lute, a string-less harp, or a super-theremin, or a capacitive touch controller such as the Freescale MPR121. Using conductive paint for the capacitive inputs could inspire even more interesting inventions.

Android

Android is an easy-to-use, yet powerful operating system, intended for smartphones, tablets and other portable devices. It also runs very well on the ODROID-C1, not only as an inexpensive testing platform for building Android applications, but also as a set-top box for video and music streaming and playback, a general-purpose operating system capable of web browsing, social networking, remote control of other computers, and much more.

Desktop Environment

The Hardkernel Android image offers several Android tablet features, including two pulldown menus at the top of the screen. As shown below, the top left menu shows notifications and application information. The top right menu offers direct access to the operating system

Chapter 3

settings. To open the applications menu, click on the circle with the six dots, which displays an alphabetical list of all installed apps. The Recent menu shows the recently opened applications. Power Options allows you to reboot, shutdown the device and put it in airplane mode, which disables all wireless functions, including Bluetooth.



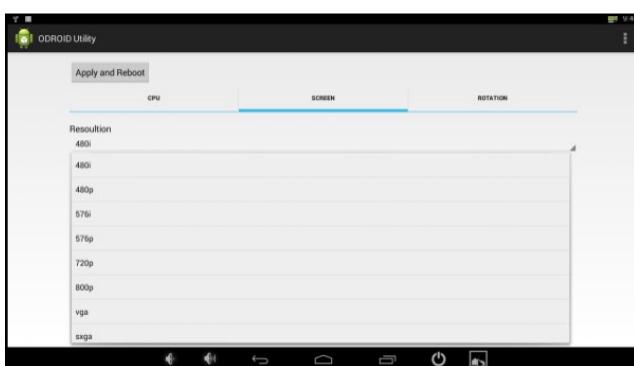
ODROID Utility and Updater

To run the ODROID Utility application, open the applications menu and click on the ODROID Utility icon. When you start the ODROID Utility for the first time, it will ask for superuser permissions, which should be confirmed. This application contains several tabs which can be used to configure various aspects of the hardware:

CPU changes the CPU governor settings and can set the Kodi media center to launch at system startup. The Performance governor will give the best speed but consume the most power.

Screen allows you to change the resolution as well as stretch and move the screen.

Rotation allows you to rotate the screen to either Portrait or Landscape mode.



Setting the Display Resolution

On the Screen tab of the ODROID Utility application, select the preferred monitor resolution, then click on Apply and Reboot. The new resolution will be active after the reboot has completed.

An overscan adjustment tool is also provided in the ODROID Utility application. To adjust the screen, click on the Screen Tab. You will then see a group of arrows that represents the four screen edges. For the edges that you wish to adjust, click on the corresponding arrow

Chapter 3

in order to set the number of offset pixels. After adjusting the settings, click on the “Apply and Reboot” button.

Installing Google Play and applications

The official Hardkernel Android image comes without any Google applications installed. The best way to install them, is to use the “Universal 1-Click GApps Installer for ODROID” application, available at <http://bit.ly/1gkv4PM>.

Kodi

To install Kodi, you will first need to download the most recent .apk file from the Kodi Download page at <http://bit.ly/1yIrUDd>, as shown in the image below. Find the link to the ARM build, then click on the link to download the .apk installation package. After download, simply click on the .apk file to install. The official Kodi website provides more detail on the Android installation at <http://bit.ly/1FFK7B9>.



Netflix

Install Netflix from Google Play, then click on its icon, which should be automatically added to the Android desktop. Login with your existing Netflix account or create a new one. When you start a video, it always begins playing at low quality, then switches to higher quality after about 30 seconds, depending on the speed of your Internet connection.

Gaming

Most Android games are compatible with the ODROID-C1, but some applications that use multitouch gestures won't be playable without a touchscreen. The ODROID-VU, available from the Hardkernel Store, will allow all Android games to be played without the need for an auxiliary controller.

Music and MIDI

There are a number of Android Apps available for music and MIDI. In order to work correctly with Android, USB MIDI adapters and

Chapter 3

sound cards typically need to be class-compliant USB MIDI devices that do not require a device driver, and need to be used on a USB connector that supports host mode. For some devices, the USB OTG connection must be used with an adapter.

The MIDI Manufacturers Association webpage at <http://bit.ly/1IzGkvb> is a good source for additional links on this topic. Other good webpages for MIDI and computer music on Android include <http://bit.ly/1FFKoUH> and <http://bit.ly/1IzGumw>.

TouchDAW is an excellent and mature app which allows the C1 to be a MIDI Controller for DAW and general-purpose MIDI control on secondary workstations. TouchDAW supports the leading computer music production applications on Windows, IOS, and Linux systems.

Using Bluetooth Devices with Android

Bluetooth device detection and pairing are very easy with Android using the Settings application. After plugging in a USB bluetooth adapter and powering up the C1, you will need to enable the bluetooth receiver. Open the Settings application, find Bluetooth in the settings list under the Wireless and Networks category, and select ON to enable your USB bluetooth adapter and bluetooth services. To begin pairing your bluetooth device to the C1, turn on your device and put the device into pairing mode. The Bluetooth manager will display a list of detected devices, as well as devices that have been successfully paired.

If your device was detected, but not shown to be a paired device, click on the sliders icon on the right side of the listed device to open the bluetooth settings for this device. Complete the pairing process by entering the pairing code or passkey for the device. If the bluetooth device was not detected, clicking on search for devices will perform a manual scan for any detectable bluetooth devices within range of the C1.

Adding an ODROID-VU Touchscreen

The ODROID-VU is a 10-point multitouch display that is capable of both 720p and 1080p resolutions. To use the ODROID-VU with Android, connect the USB cable from the VU to the ODROID-C1's USB port, then use a microHDMI to HDMI cable to connect the display. Hardkernel's Android images support the ODROID-VU without any additional configuration. Use the ODROID Utility application to adjust the resolution to a comfortable level, if necessary.

Hardware Tinkering

Now that you have been introduced to the C1 and have become aware of booting it up with a image, it is time to get acquainted with one its main purposes – to be able to interact with the external world through the age-old activity of tinkering. While its design and production is a marvel in itself, it really shines in its use to interact with the external world, through a slew of breakout-boards and sensory devices – all under the control of powerful operating systems like Linux or Android.

Here are the devices covered in this section:

- **USB / UART Module Kit, Bluetooth Module 2 and Tinkering Kit**
- **Oduino One and ODROID-SHOW; Weather Board**
- **16x2 LCD I/O Shield and 3.2inch TFT+Touchscreen Shield**
- **DC Plug Cable Assembly 2.5mm and USB-DC Plug Cable 2.5mm x 0.8mm**
- **USB-SPDIF and ODROID USB-CAM 720P**
- **UPS2 and USB GPS Module**

USB / UART Module Kit

It is beneficial to get a view of the early boot process, especially if you are working with newly added external devices or creating special purpose boot images. Typically these activities can be debugged by viewing their corresponding log entries using the dmesg application. This can be achieved using the USB / UART module and an microUSB male to USB male cable, which is typically used to charge smart-phones and tablets. Together, they form the kit, shown below:



USB / UART kit

When using Microsoft Windows 7+ as a host PC system, you will need to install a terminal application such as PuTTY. This can be used to set up commonly used configurations in order to access the

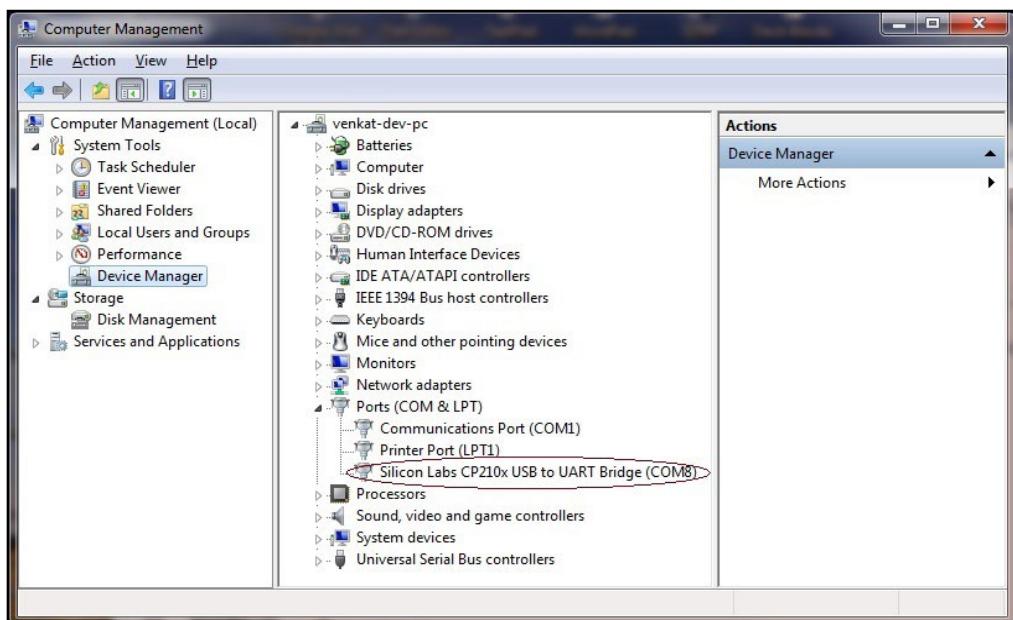
Chapter 4

C1 either via a serial connection or SSH.

Prior to setting up the kit, power off the C1. Attach one end of the USB / UART module to the C1. Attach the other (micro USB) end of the USB / UART module to the USB cable. Attach the free end of the cable to an available USB port of the host PC system, then power on the C1.

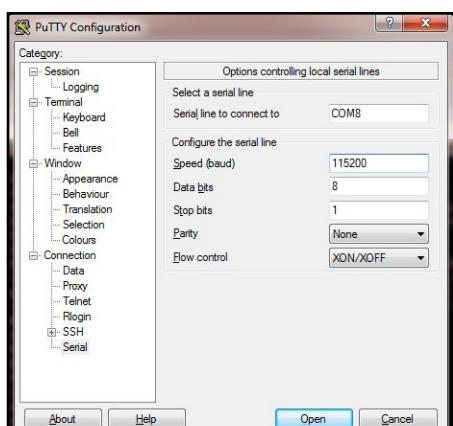
Windows Host Setup

The Windows 7+ host PC system will recognize the module as a Silicon Labs CP210x USB to UART Bridge and install the appropriate device driver. After installation has completed, you can verify which COM port the module is associated. Launch the Computer Management utility of Windows 7+, highlight the Device Manager option, and expand the Ports list. In this particular case, as shown in the below example screenshot, it can be see that the module is installed on COM8.



USB / UART module on COM8

Launch PuTTY and select the Serial option on the left pane.

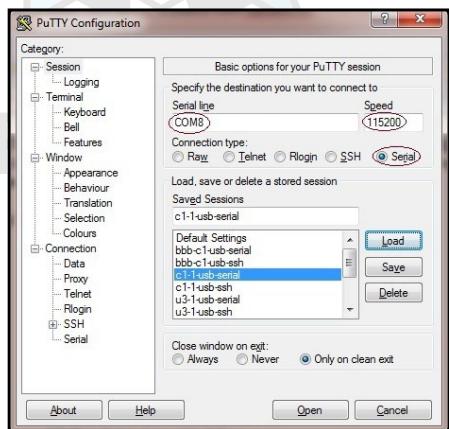


Serial line to connect to: COM8
Speed (baud): 115200
Data bits: 8
Stop bits: 1
Parity: None
Flow Control: XON/XOFF

Serial configuration in PuTTY

Chapter 4

Enter configuration information as shown below. Then select the Session option on the left pane. You will see a popup window as indicated in the following image.

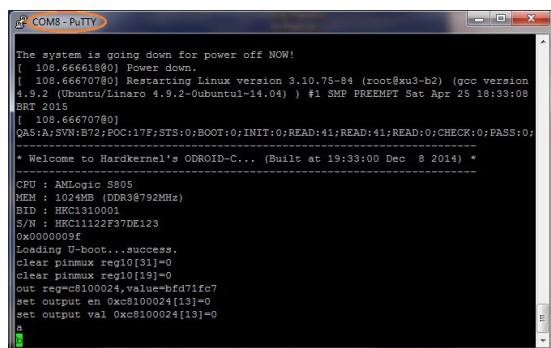


Serial line: COM8
Speed: 115200
Connection Type: Serial

Saved serial configuration in PuTTY

Give this configuration a name such as c1-1-usb-serial, then save it. This saved configuration can be reloaded and used anytime, as long as the COM port on the host PC system has not changed. Click on the Open button to start a session.

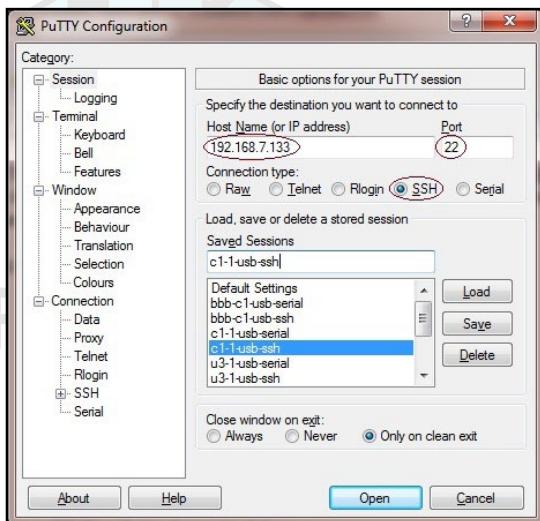
Next, reboot the C1. After a short delay, you will see messages scroll by in the PuTTY console, as shown in the below image. This is when one would notice messages relevant to external devices or software modules. These messages can help with debugging issues you may encounter. Most of these messages will also be listed through the dmesg command.



PuTTY serial console

Close the serial console, then set up the SSH connection option by relaunching the PuTTY application and selecting the Session option on the left pane. For this SSH option to work, the host PC system and the C1 need to be visible to each other on the same local area network (LAN). Refer to the screenshot below for an example PuTTY configuration.

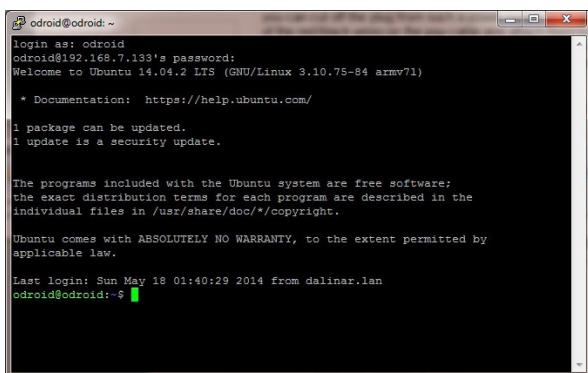
Chapter 4



Saved SSH configuration in PuTTY

Enter the C1's IP address or hostname (if it can be resolved from the host PC system) and a numerical value of 22 for the Port number. Select the Connection Type to be SSH. Enter c1-1-usb-SSH for the configuration name and save it. This saved configuration can be reloaded and used anytime, as long as the C1's ip-address and hostname remain the same. Click on the Open button to start a session.

Now, reboot the C1, wait approximately 60 seconds, then click on Open in Putty and you should be prompted for a username and password, which are typically "odroid" and "odroid". You will see a screen presented as shown in the image below.



PuTTY SSH console

You can now access the C1 system as if you had locally opened a terminal Window. While you cannot view the boot-time log in real-time, you can run a large number of Linux commands (including dmesg) from this SSH session.

So far, we've addressed the case where the host PC system is a Windows 7+ system. Next, we present information on the use of a Ubuntu desktop Linux host system. Most of the steps apply to a host system running any Linux version, including a server OS. The steps

Chapter 4

do not address the use of Linux hosted in a virtual environment, such as VirtualBox.

Linux Host Setup

Access the Linux desktop of the host system and launch a terminal session. Attach the USB cable from the USB / UART setup to the host system. The following commands can be used to verify proper installation:

```
$ lsusb
...
Bus 001 Device 006: ID 10c4:ea60 Cygnal Integrated Products, Inc.
CP210x UART Bridge / myAVR mySmartUSB light
...
```

Using the output above, you get obtain additional details of the USB / UART module, using the command:

```
$ sudo lsusb -D /dev/bus/usb/001/006
Device: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART
Bridge / myAVR mySmartUSB light
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         2.00
  bDeviceClass      0 (Defined at Interface level)
  bDeviceSubClass    0
  bDeviceProtocol     0
  bMaxPacketSize0     64
  idVendor        0x10c4 Cygnal Integrated Products, Inc.
  idProduct        0xea60 CP210x UART Bridge / myAVR mySmartUSB
light
  bcdDevice        1.00
  iManufacturer      1 Silicon Labs
  iProduct          2 CP2104 USB to UART Bridge Controller
  iSerial           3 00513B0C
  bNumConfigurations 1
...
```

The TTY port on which the module is detected, can be obtained:

```
$ sudo ls -lsa /dev/tty* | grep USB
0 crw-rw---- 1 root dialout 188,   0 May  7 16:48 /dev/ttyUSB0
```

The lockfile location can be checked using the command:

```
$ sudo ls -lsa /var/lock*
0 lrwxrwxrwx 1 root root 9 Apr 16 2014 /var/lock -> /run/lock
```

This information is used to setup the communications between the

Chapter 4

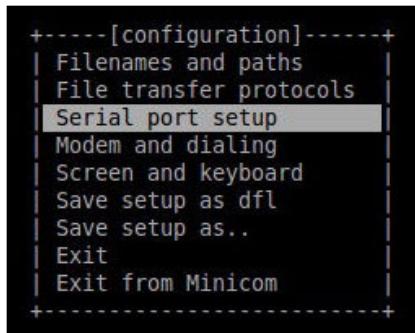
host system and USB / UART cable attached to the C1. The application used to enable communication will be minicom:

```
$ minicom -version  
minicom version 2.6.1 (compiled May 1 2012)  
Copyright (C) Miquel van Smoorenburg.  
...
```

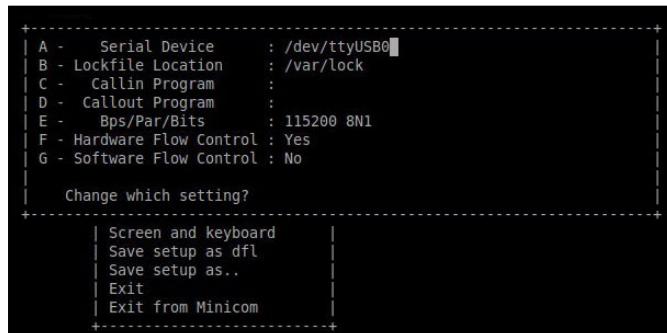
Launch minicom, using the following command:

```
$ sudo minicom -o -s
```

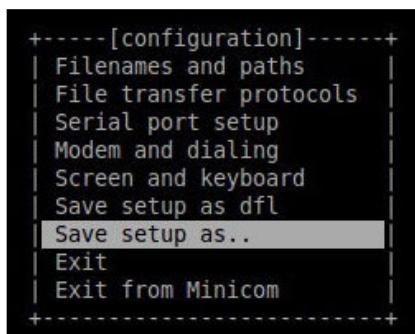
Here are the four screens you would go through to configure and use minicom:



Port option



Serial port data



Save setup



Minicom session

The four steps include:

Select the Serial port setup option and you will be presented with screen 2.

In screen 2, type A to select option A. Edit the serial device to that used on your system. In this case it is: /dev/ttyUSB0. Ensure the lockfile location is what was obtained earlier. Likewise check option E reflects the information shown in the figure: 115200 8N1. Ensure Hardware Flow Control is set to YES. Then hit the ENTER key to go to the next screen.

In screen 3, select the “Save setup as ...” option to save this config for future use. Enter a filename at the new prompt. Hit ENTER

Chapter 4

to accept the filename. Then select Exit option to complete the config process.

You will now return to the terminal windows, as shown in step 4. Reboot the C1 and you will observe boot-time information scroll through the minicom (terminal) session.

Bluetooth Module 2

Hardkernel offers a bluetooth adapter called the Bluetooth (4.0) Module 2 which is certified to work with the C1 under both Android and Linux. However, quite a few of the plug-and-play adapters that are compatible with Microsoft Windows 7+ can also be expected to work with the C1 using Linux, such as the Asus BT400 Bluetooth adapter.



Bluetooth Module 2 adapter and Asus BT400 adapter

While some of the required software modules may already exist in your C1's installed image, it is helpful to install the following additional modules and utilities, using the following commands:

```
$ sudo apt-get install bluez-dbg \
bluez-hcidump bluez-utils bluez-tools
$ sudo apt-get install bluewho blueman python-bluetooth
```

Some of them could be used to debug your setup, if you use a non-Hardkernel offered bluetooth adapter. Next, check the USB information related to the adapters:

```
$ lsusb
...
Bus 001 Device 005: ID 0b05:17cb ASUSTek Computer, Inc.
...
```

Check the Bluetooth adapter's support via Lubuntu for additional features by examining the dmesg logs:

```
$ dmesg | grep Blue
[    0.851848@0] Bluetooth: Core ver 2.16
[    0.859721@0] Bluetooth: HCI device and connection
manager initialized
[    0.866240@0] Bluetooth: HCI socket layer initialized
```

Chapter 4

```
[ 0.871245@0] Bluetooth: L2CAP socket layer initialized
[ 0.876447@0] Bluetooth: SCO socket layer initialized
[ 1.429422@2] Bluetooth: HCI UART driver ver 2.2
[ 1.433876@2] Bluetooth: HCI H4 protocol initialized
[ 1.438828@2] Bluetooth: HCI BCSP protocol initialized
[ 1.443919@2] Bluetooth: HCILL protocol initialized
[ 1.448782@2] Bluetooth: HCIATH3K protocol initialized
[ 1.453877@2] Bluetooth: HCI Three-wire UART (H5) protocol initialized
[ 3.236424@2] Bluetooth: btwake_control_init Driver Ver 1.1
[ 3.366366@2] Bluetooth: RFCOMM TTY layer initialized
[ 3.371156@2] Bluetooth: RFCOMM socket layer initialized
[ 3.376392@2] Bluetooth: RFCOMM ver 1.11
[ 3.380308@2] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[ 3.385744@2] Bluetooth: BNEP filters: protocol multicast
[ 3.397895@2] Bluetooth: BNEP socket layer initialized
[ 3.402975@2] Bluetooth: HIDP (Human Interface Emulation) ver
1.2
[ 3.409060@2] Bluetooth: HIDP socket layer initialized
```

You can then check the list of the installed Bluetooth modules:

```
$ dpkg -l | grep blue
ii  blueman           ...  armhf  Graphical bluetooth manager
ii  bluez              ...  armhf  Bluetooth tools and daemons
ii  bluezalsa:armhf   ...  armhf  Bluetooth ALSA support
ii  bluezcups          ...  armhf  Bluetooth printer driver for
CUPS
ii  libbluetooth-dev   ...  armhf  Dev. files for BlueZ Linux
Bluetooth lib
ii  libbluetooth3:armhf ...  armhf  Library to use the BlueZ Li-
nux BT stack
ii  libgnomebluetooth1 ...  armhf  GNOME Bluetooth tools - sup-
port library
```

With the positive signs observed in the output so far, we can check the presence of the Bluetooth device, which is useful for connection configuration later:

```
$ hcitool dev
Devices:
      hci0      00:02:72:CC:F4:CE

$ hciconfig
hci0:  Type: BR/EDR  Bus: USB
      BD Address: 00:02:72:CC:F4:CE  ACL MTU: 1021:8  SCO MTU: 64:1
      UP RUNNING PSCAN
      RX bytes:583 acl:0 sco:0 events:33 errors:0
      TX bytes:898 acl:0 sco:0 commands:33 errors:0

$ sudo rfkill list all
0: hci0: Bluetooth
      Soft blocked: no
```

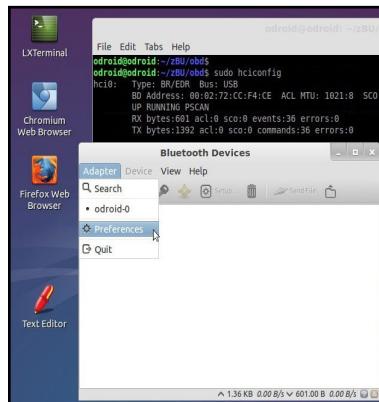
Chapter 4

Hard blocked: no

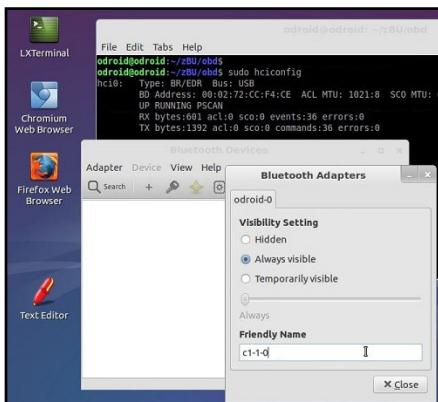
To configure the adapter from the Lubuntu desktop, launch the Bluetooth Manager configuration utility, then select the Preferences menu item to configure the application, as shown in the screenshots. Update the friendly-name of the Bluetooth adapter using a name like say, c1-1-0. You can make the device always visible when other Bluetooth devices scan for this adapter. Save the changes and reboot.



Launch Bluetooth Manager



Bluetooth Manager configuration



Bluetooth Manager config update

If there is a need to test/debug issues related to the adapter, you can use the command-line Bluetooth packet analyzer/sniffer to analyze Bluetooth traffic:

```
$ sudo hcidump
HCI sniffer - Bluetooth packet analyzer ver 2.5
device: hci0 snap_len: 1500 filter: 0xffffffff
< HCI Command: Write Class of Device (0x03|0x0024) plen 3
    class 0x700100
> HCI Event: Command Complete (0x0e) plen 4
    Write Class of Device (0x03|0x0024) ncmd 1
    status 0x00
< HCI Command: Write Extended Inquiry Response (0x03|0x0052) plen 241
    fec 0x00
```

Chapter 4

```
Complete local name: 'c1-1-0'
TX power level: 0
Complete service classes: 0x112d 0x1112 0x111f 0x111e 0x110c
0x110e 0x1105
> HCI Event: Command Complete (0x0e) plen 4
    Write Extended Inquiry Response (0x03|0x0052) ncmd 1
    status 0x00
< HCI Command: Write Extended Inquiry Response (0x03|0x0052) plen
241
    fec 0x00
    Complete local name: 'c1-1-0'
    TX power level: 0
    Complete service classes: 0x112d 0x1112 0x111f 0x111e 0x110c
0x110e ...
> HCI Event: Command Complete (0x0e) plen 4
    Write Extended Inquiry Response (0x03|0x0052) ncmd 1
    status 0x00
...
...
```

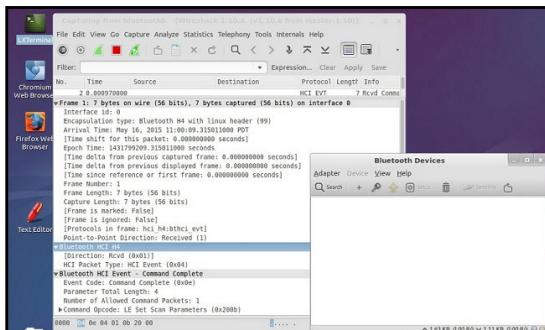
Another useful tool is wireshark. It gives a graphic view of the snooped data, etc. It can be installed using the command:

```
$ sudo apt-get install wireshark
```

After installation, you can launch wireshark following the steps shown in the following images. The welcome screen should appear, reflecting that the Bluetooth adapter on the C1 has been detected by wireshark. Click on the green shark-fin icon to request the start of data capturing.



Wireshark welcome screen with bluetooth adapter detected



Wireshark with snooped bluetooth traffic

Chapter 4

Then, launch the Bluetooth Manager application and click on its Search button. You should immediately see snooped information present itself in the wireshark application.

Tinkering Kit

While the power-efficient C1 is a very capable system for software development, it is also highly suited for prototyping hardware development. Towards that end, Hardkernel has developed the C1 Tinkering kit with numerous items to get you started, as shown in the following image.



The Hardkernel Tinkering Kit includes many useful items

Some of the components in the Tinkering Kit include:

Assembled T-breakout PCB - 40Pin GPIO Breakout board
Breadboard - 630 Tie-points with dual power lanes
40pin Ribbon cable - IDC Flat cable 100mm
40pin Male-to-Male Dupont jumper Wire 170mm
Green/Yellow/Red 3mm LEDs – 7 each
2 x Photo Cell (CdS Light sensor), 6 x Tact Switches
330 Ω / 10 KΩ 1/6W resistors – 50 each

Chapter 4

ODROID-C1 40pin Layout					
				Power Pin	
				Special Function	
				GPIO/Special Function	
ODROID-C PIN	Label	HEADER		Label	ODROID-C PIN
	3V3	1	2	5V0	
	I2CA_SDA	SDA1	3	4	5V0
	I2CA_SCL	SCL1	5	6	GND
	GPIOY.BIT3	#83	7	8	TXD1
		GND	9	10	RXD1
	GPIOY.BIT8	#88	11	12	#87
	GPIOX.BIT19	#116	13	14	GND
	GPIOX.BIT18	#115	15	16	#104
		3V3	17	18	#102
MOSI	GPIOX.BIT10	MOSI	19	20	GND
MISO	GPIOX.BIT9	MISO	21	22	#103
SCLK	GPIOX.BIT8	SCLK	23	24	CE0
		GND	25	26	#118
	I2CB_SDA	SDA2	27	28	SCL2
	GPIOX.BIT4	#101	29	30	GND
	GPIOX.BIT3	#100	31	32	#99
	GPIOX.BIT11	#108	33	34	GND
	GPIOX.BIT0	#97	35	36	#98
	ADC.AIN1	AIN1	37	38	1V8
		GND	39	40	AIN0
					ADC.AIN0

GPIO layout for the 40-pin header on the ODROID-C1

The T-breakout board breaks-out all the pins from the 40-pin header onto a solderless breadboard. The pins include, power, GPIO, I2C, and ADC pins. This simplifies prototyping of data acquisition and control projects.

It is advisable to refer to the Amlogic S805 datasheet, available at <http://bit.ly/1dFEHhx>, to fully understand the GPIO designations.

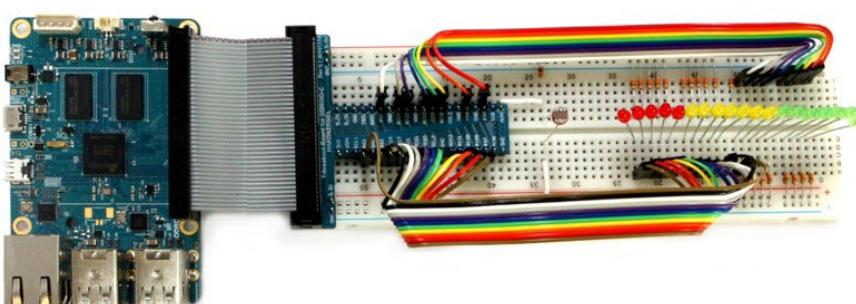
Chapter 4

ODROID-C1 40pin Layout											
										Power Pin	
										Special Function	
										GPIO/Special Function	
WiringPi GPIO#	Export GPIO#	ODROID-C PIN	Label	HEADER	Label	ODROID-C PIN	Export GPIO#	WiringPi GPIO#			
			3V3	1	2	5V0					
		I2CA_SDA	SDA1	3	4	5V0					
		I2CA_SCL	SCL1	5	6	GND					
7	83	GPIOY.BIT3	#83	7	8	TXD1	TXD_B		113		
			GND	9	10	RXD1	RXD_B		114		
0	88	GPIOY.BIT8	#88	11	12	#87	GPIOY.BIT7		87	1	
2	116	GPIOX.BIT19	#116	13	14	GND					
3	115	GPIOX.BIT18	#115	15	16	#104	GPIOX.BIT7		104	4	
			3V3	17	18	#102	GPIOX.BIT5		102	5	
12	107	MOSI	GPIOX.BIT10	MOSI	19	20	GND				
13	106	MISO	GPIOX.BIT9	MISO	21	22	#103	GPIOX.BIT6		103	6
14	105	SCLK	GPIOX.BIT8	SCLK	23	24	CE0	GPIOX.BIT20	CE0	117	10
			GND	25	26	#118	GPIOX.BIT21		118	11	
		I2CB_SDA	SDA2	27	28	SCL2	I2CB_SCL				
21	101	GPIOX.BIT4	#101	29	30	GND					
22	100	GPIOX.BIT3	#100	31	32	#99	GPIOX.BIT2		99	26	
23	108	GPIOX.BIT11	#108	33	34	GND					
24	97	GPIOX.BIT0	#97	35	36	#98	GPIOX.BIT1		98	27	
		ADC.AIN1	AIN1	37	38	1V8	1V8				
			GND	39	40	AIN0	ADC.AIN0				

40-pin layout with GPIO mappings for the Wiring Pi library

The above image illustrates simplified information on layout and the designation of the 40 pins. Use the data sheet and this information with caution when applying voltages to the pins, to ensure compliance and prevent possible device damage.

The WiringPi library has been ported to work with C1. The chart above illustrates the GPIO pin numbering translation scheme used in the WiringPi library.



Tinkering project example

The image above shows a project example using the C1 and the tinkering kit. Details can be found in the related wiki at <http://bit.ly/1NsrIU9>. You can wire up the project to reflect the example.

Prior to trying out the example, you will need to build and install the WiringPi library by running the following commands in a Terminal window:

Chapter 4

```
$ cd ~  
$ git clone https://github.com/hardkernel/WiringPi  
$ cd WiringPi  
$ ./build
```

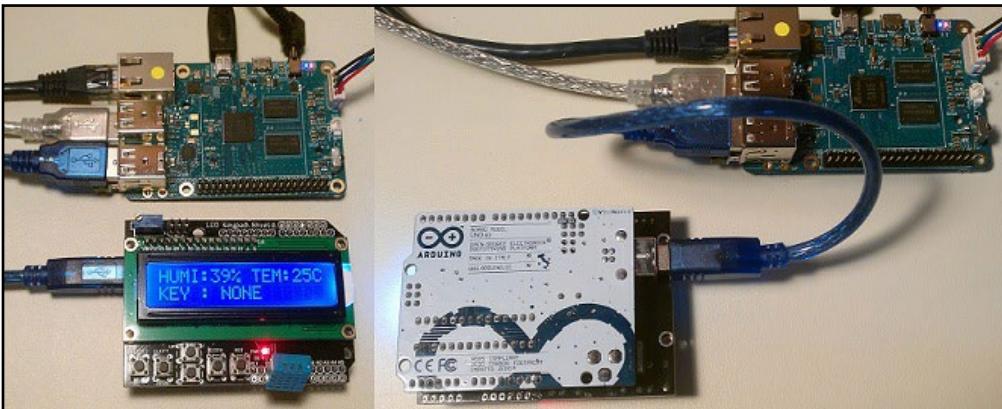
Then, download the C-language sample source code example-led.c from the previously mentioned Wiki, to a working directory such as tkit-example, then build and run it using the following commands:

```
$ cd ~ && mkdir tkit-example/ && cd tkit-example/  
$ gcc -o example-led example-led.c -lWiringPi -lWiringPiDev -lpthread  
$ sudo ./example-led
```

Note that GPIO access requires root user privilege. Move your finger closer to the light sensor and you should see the LEDs light up.

ODUINO ONE

The ODROID-compatible Arduino called the ODUINO ONE includes the following components, all packaged into one experimentation unit, as shown in the image below.



The ODUINO ONE unit

- **Arduino Uno R3 + a shield containing**
- **one 16x2 LCD display**
- **one DHT11 one-wire humidity / Temperature sensor**
- **one keypad (four directional keys)**

The ONE package can be powered by the C1 using the USB Std A/Std B cable. Connect the cable to the ODUINO ONE and the C1. The ONE goes through the power-up process and displays the ambient humidity and temperature. It can also display the key that gets clicked on the keypad. Some additional information on the ONE

Chapter 4

package can be obtained using the following commands:

```
$ ls -lsa /dev/ttyA*
0 crw-rw---- 1 root dialout 166, 0 May 24 21:05 /dev/ttyACM0

$ lsusb
...
Bus 001 Device 009: ID 2341:0043 Arduino SA Uno R3 (CDC ACM)
...
$ lsusb -D /dev/bus/usb/001/009
Device: ID 2341:0043 Arduino SA Uno R3 (CDC ACM)
Couldn't open device, some information will be missing
Device Descriptor:
  bLength          18
  bDescriptorType    1
  bcdUSB           1.10
  bDeviceClass       2 Communications
  bDeviceSubClass     0
  bDeviceProtocol     0
  bMaxPacketSize0      8
  idVendor         0x2341 Arduino SA
  idProduct         0x0043 Uno R3 (CDC ACM)
  bcdDevice          0.01
  iManufacturer        1
  iProduct            2
  iSerial              220
  bNumConfigurations     1
```

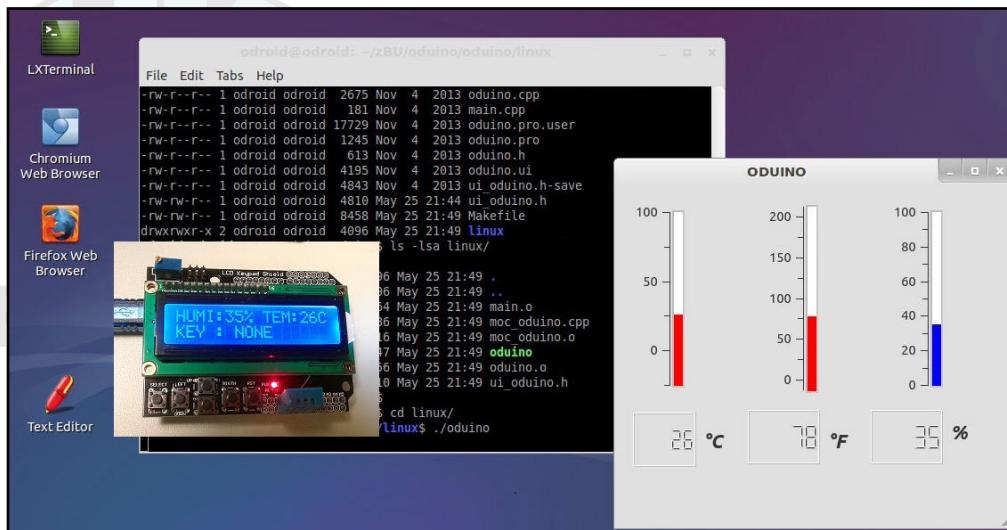
The activity on the ONE package also gets transmitted to the ODROID-C1. Hardkernel has provided C sample source code to display this information. You will need to install some pre-requisite software components first:

```
$ sudo apt-get install qt4-default qt4-designer libqwt-dev
```

You can then download the sample (Qt based) source code to the C1:

```
$ cd ~
$ wget -c http://dn.odroid.com/ODUINO/source/Ubuntu_Qt/oduino.zip
$ unzip oduino.zip
$ cd oduino
$ mv ui_oduino.h ui_oduino.h-save
$ uic oduin.ui > ui_oduino.h
$ qmake
$ make
```

Chapter 4



ODUINO ONE data captured by application on C1

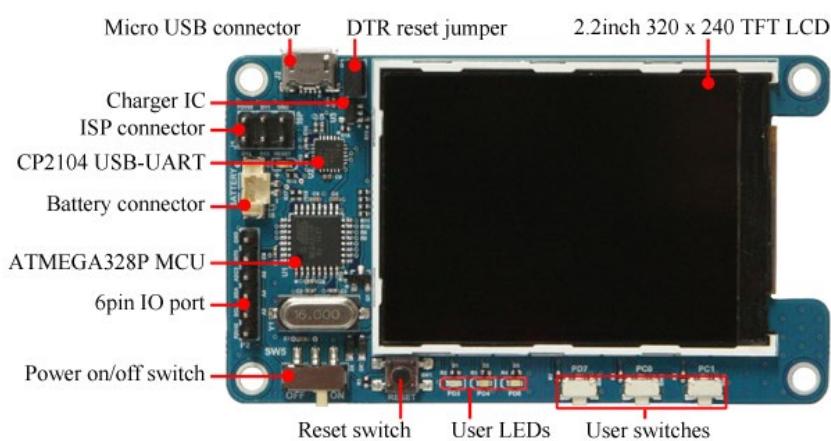
You can launch the freshly built application using the following commands:

```
$ cd Linux/  
$ ./oduino
```

The application user-interface should look like the above image.

ODROID-SHOW2

Hardkernel offers an Arduino compatible 2.2" 240×320 TFT-LCD display, called ODROID-SHOW2, that can be used with the C1, a PC or other Hardkernel boards. The LCD display can be used to display simple messages and hardware status, and the user switches mounted at the bottom of the board allow simple user interaction and input.

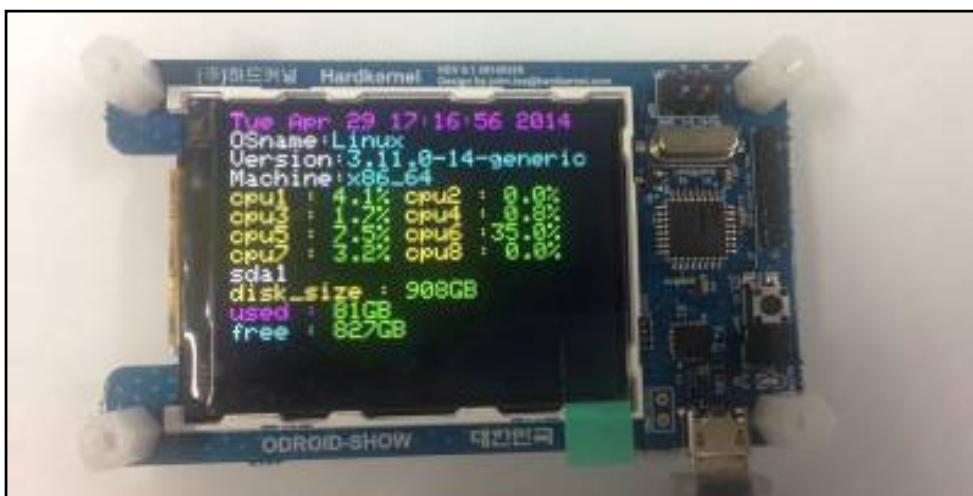


ODROID-SHOW2 annotated image

After attaching the SHOW to the C1, use the following commands to get details on it:

Chapter 4

```
$ lsusb
...
Bus 001 Device 005: ID 10c4:ea60 Cygnal Integrated Products, Inc.
CP210x UART Bridge / myAVR mySmartUSB light
...
$ sudo lsusb -D /dev/bus/usb/001/005
[sudo] password for odroid:
Device: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART
Bridge / myAVR mySmartUSB light
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         2.00
  bDeviceClass      0 (Defined at Interface level)
  bDeviceSubClass    0
  bDeviceProtocol     0
  bMaxPacketSize0    64
  idVendor        0x10c4 Cygnal Integrated Products, Inc.
  idProduct        0xea60 CP210x UART Bridge / myAVR mySmartUSB
light
  bcdDevice        1.00
  iManufacturer      1 Silicon Labs
  iProduct           2 CP2104 USB to UART Bridge Controller
  iSerial            3 00875559
  bNumConfigurations 1
...
$ ls /dev/ttyUSB*
/dev/ttyUSBO
```



ODROID-SHOW example project output

Obtain the SHOW sample source code and build a sample using the commands:

```
$ cd ~
$ git clone https://github.com/hardkernel/ODROID-SHOW
$ cd ODROID-SHOW/example/Linux
$ gcc -o status status.c
$ sudo ./status
```

Chapter 4

You should see the LCD display similar to the following image:



ODROID-SHOW showing the sample project startup screen

Additional details, such as, burning new firmware for the SHOW2 (or SHOW1) and sample applications can be found at <http://bit.ly/1toe7P1>.

One important aspect of the SHOW2 (or SHOW1) board, is the presence of a Data Terminal Ready (DTR) reset jumper/switch. Its role is important during burning of new firmware. The following precautions need to be followed, regarding the DTR jumper:

- ***It must be installed when you upload the firmware.***
- ***It must NOT be installed during normal usage mode.***

16x2 LCD I/O Shield

Hardkernel offers an I/O shield with a, 16 chars by 2 lines, white on blue, LCD display, tactile switches and I/O pins. Its use requires the WiringPi library, whose installation was discussed earlier. After installing the shield on the C1, the sample source can be obtained from <http://bit.ly/1fbtF1E>.

Copy the sample code using the following commands:

```
$ cd ~ && mkdir lcdio && cd lcdio  
$ mv ~/Downloads/lcd_cpuInfo.c .
```

Fix an issue with the sample code, where the temperature is displayed wrongly. Open the C source file in an editor and modify the following line, then save the changes:

```
sprintf(lcdFb[0], "TEMP %d *C", atoi(cpuTemp));
```

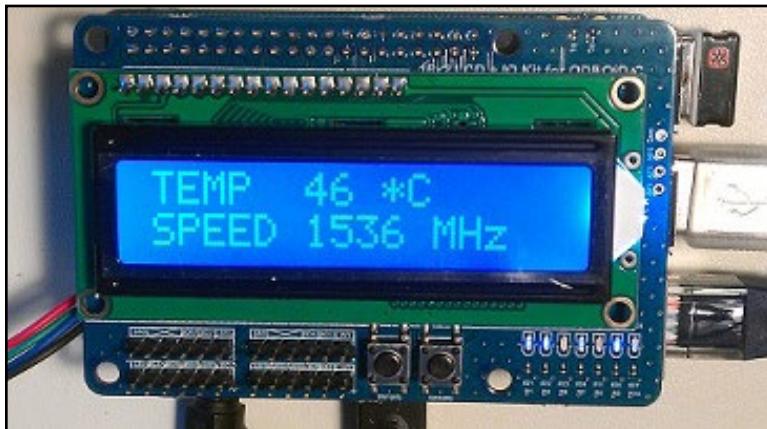
Chapter 4

to

```
sprintf(lcdFb[0], "TEMP %d *C", atoi(cpuTemp)/1000);
```

Then, build and launch the application using the following commands, which should show a display similar to the image below:

```
$ gcc -o lcd_cpuInfo lcd_cpuInfo.c -lwiringPi -lwiringPiDev -lp-thread  
$ sudo ./lcd_cpuInfo
```



16x2 LCD – I/O shield showing some CPU data

Weather Board

The Weather-board shown below is an add-on developed by Hardkernel, to be used with either the SHOW or 16x2 LCD device. Below are the steps needed to get it to work with the C1 and ODROID-SHOW.



Hardkernel's Weather Board peripheral

Chapter 4

Install the Device Tree Compiler (DTC) using the command:

```
$ sudo apt-get install device-tree-compiler
```

Then, enable the sensors on the device tree using the commands:

```
$ sudo fdtput -c /media/boot/meson8b_odroidc.dtb /i2c@c1108500/bmp085@77
$ sudo fdtput -c /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si702x@40
$ sudo fdtput -c /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si1132@60

$ sudo fdtput -t s /media/boot/meson8b_odroidc.dtb /i2c@c1108500/bmp085@77 compatible "bosch,bmp085"
$ sudo fdtput -t s /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si702x@40 compatible "silab,si702x"
$ sudo fdtput -t s /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si1132@60 compatible "silab,sil1132"

$ sudo fdtput -t x /media/boot/meson8b_odroidc.dtb /i2c@c1108500/bmp085@77 reg 0x77
$ sudo fdtput -t x /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si702x@40 reg 0x40
$ sudo fdtput -t x /media/boot/meson8b_odroidc.dtb /i2c@c1108500/si1132@60 reg 0x60
```

Enable the i2c module to auto-load on boot-up:

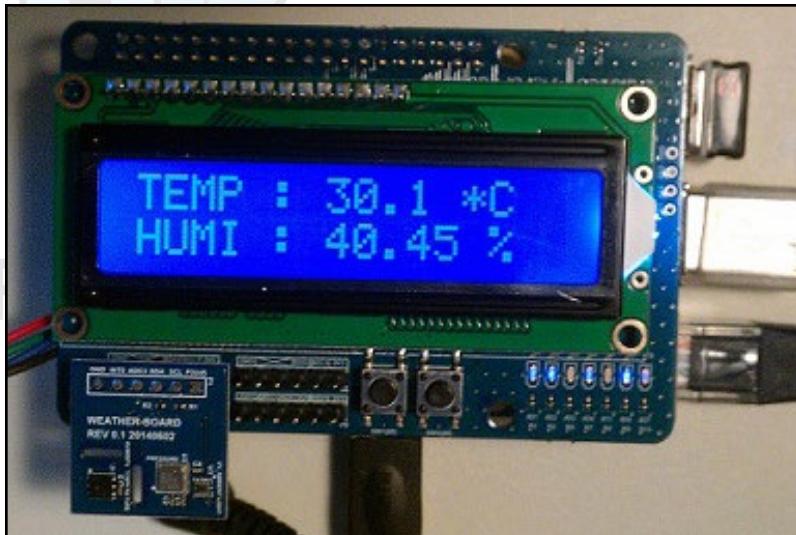
```
$ sudo echo aml_i2c >> /etc/modules
```

Ensure the WiringPi is installed as described earlier. Power down the system. Attach the weather-board to 16x2 display, then power up the whole system. Next, download the wb_si702x.c sample source file from <http://bit.ly/1Hles6U>, then build and launch it:

```
$ cd ~ && mkdir wb && cd wb
$ mv ~/Downloads/wb_si702x.c .
$ gcc -o wb_si702x wb_si702x.c -lWiringPi -lWiringPiDev -lpthread
$ ./wb_si702x.c
```

You should see the LCD module display the current ambient temperature and humidity, as shown in the following image.

Chapter 4



LCD displaying weather information

3.2inch TFT+Touchscreen Shield

Considering the size and form-factor of the C1, which is ideal for portable applications, it is only natural to expect the support for a small touchscreen. Hardkernel offers one such 3.2" 320x240 pixel TFT LCD screen with a resistive touch display, as shown here:



TFT touchscreen setup

This display is in the form of a shield that fits on to some of the the GPIO pins of the C1. To activate the display, start with the upgrade commands below:

```
$ sudo apt-get autoremove && sudo apt-get update && \
  sudo apt-get upgrade
$ sudo apt-get dist-upgrade && sudo apt-get install linux-image-c1
```

Then insert the modules using, with one command per line:

```
$ sudo modprobe spicc
$ sudo modprobe fbtft_device name=odroidc_tft32 rotate=270 \
```

Chapter 4

```
gpios=reset:116,dc:115 speed=32000000 cs=0
```

You can achieve the same, by updating the /etc/modules file to include the following configuration, which should be placed on two lines:

```
spicc  
fbtft_device name=odroidc_tft32 rotate=270 gpios=reset:116,dc:115  
speed=32000000 cs=0
```

To run X windows while using the display, update the X11 related configuration:

```
$ cd /usr/share/X11/xorg.conf.d/  
$ sudo touch 99-odroidc-tftlcd.conf  
$ sudo medit 99-odroidc-tftlcd.conf
```

Add the following contents to this new file, then save the changes:

```
Section "Device"  
    Identifier      "C1 fbdev"  
    Driver          "fbdev"  
    Option          "fbdev"  "/dev/fb2"  
EndSection
```

Then disable the Xorg option on your C1, using the ODROID Utility:

```
$ sudo odroid-utility
```

Then, use the up / down cursor keys to select the Xorg On/Off menu option and hit <ENTER>, then select Disable Xorg. Update the kernel launch statement using the command:

```
$ sudo con2fbmap 1 2
```

Alternatively, add the following kernel argument to file /media/boot/boot.ini and save the changes:

```
fbcon=map:22
```

Enable booting to the desktop using the following change to /etc/rc.local file and save the changes:

```
startx &
```

Shutdown the system, remove any attached HDMI monitor, then reboot the system to see the desktop on the new display. The display

Chapter 4

needs to be calibrated. To do so, run the following command (one line):

```
$ sudo TSLIB_FBDEVICE=/dev/fb2 \
TSLIB_TSDEVICE=/dev/input/touchscreen ts_calibrate
```

You will be presented with directions on the display. Touch each of the points to be calibrated. Then run the following command on a single line, then reboot the system:

```
$ sudo TSLIB_FBDEVICE=/dev/fb2 \
TSLIB_TSDEVICE=/dev/input/touchscreen ts_test
```

Note that only a limited number of desktop icon shortcuts can be displayed on the small 3.2" display. Some applications may not be designed to optimize their user interface to fit the small screen. As a result, some parts of the user interface may be hidden from full view.

The AmeriDroid team has also developed a software utility to install and use the TFT display. To use their utility, follow the steps below:

- Disconnect your C1 from power and HDMI and connect the screen to C1's GPIO header.
- Boot the C1 and expand the root partition (if not already done) using ODROID-Utility.
- Connect your C1 to the Internet.
- Open a console window (uxterm, xterm, or similar) and type:

```
$ sudo su
```

- Enter your superuser password (default password is odroid) and type:

```
$ wget http://respectech.com/odroid/c1-touch.sh
$ chmod 755 c1-touch.sh
$ ./c1-touch.sh
```

Type “a” and hit <ENTER> to run all steps automatically, then reboot.



DC plug cable assembly 2.5mm



DC plug with SmartPower

Chapter 4

This accessory, shown in the image below, is typically used to provide power to the C1 using a special purpose power supply. These power supplies could include bench-top models or Hardkernel's own SmartPower. Ensure that the power supply provides clean power, rated at 5V, 2.0+ Amps. Use the cable colors to ensure polarities match at either end.



DC plug cable assembly



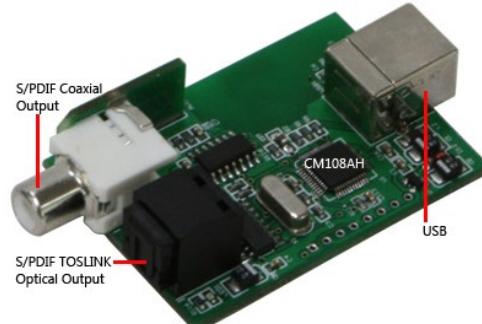
DC plug cable usage

USB-DC Plug Cable 2.5x0.8mm

This accessory is typically used to provide power to the C1 using a power supply that has an USB port. These power supplies could include units that convert AC power to DC power transformers or battery packs. The image below shows both the USB-DC plug cable and a typical LiPo battery pack. Ensure that the power supply provides clean power, rated at: 5V, 2.0+ Amps.



USB-DC plug cable



USB-DC plug cable usage

USB-SPDIF

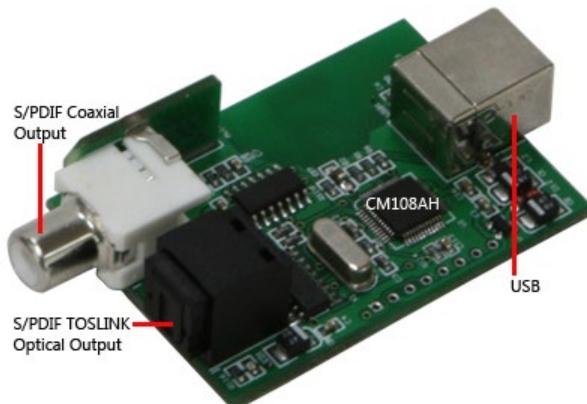
Hardkernel offers a C1-compatible S/PDIF (Sony/Philips Digital Interface Format) kit with a USB interface. This allows you to hook up the audio from the C1 via USB to an amplifier (A/V receiver), when the HDMI audio out option is not used. The images below show the kit and device internals.

Chapter 4

HAI



USB-SPDIF kit



USB-SPDIF device internals

If you use the latest Lubuntu image, you should have XBMC 13 Gotham (or newer) version of the open-source media player pre-installed. Note that the pass-through audio function does not work in older versions of XBMC. Attach the cable provided in the kit to the device on one end and the C1's USB port on the other end. Reboot the system. You can check to see if the device is detected using the commands:

```
$ lsusb
...
Bus 001 Device 009: ID 0d8c:013c C-Media Electronics, Inc. CM108
Audio Controller
...
$ sudo lsusb -D /dev/bus/usb/001/009
[sudo] password for odroid:
Device: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Con-
troller
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         1.10
  bDeviceClass      0 (Defined at Interface level)
  bDeviceSubClass    0
  bDeviceProtocol     0
  bMaxPacketSize0      8
  idVendor        0x0d8c C-Media Electronics, Inc.
  idProduct        0x013c CM108 Audio Controller
  bcdDevice        1.00
  iManufacturer      1 C-Media Electronics Inc.
  iProduct          2 USB PnP Sound Device
  iSerial            0
  bNumConfigurations   1
...
```

Now that the device is verified to be recognized by the system,

Chapter 4

configure it by first launching the PulseAudio Volume Control config option from the desktop. Then, in the Configuration tab, select the Digital Stereo Output profile, and select the Output Devices tab and update the configuration for the CM108 device as shown in the images below. Note that TrueHD or DTS-MA pass-through is unsupported.

Launch the Kodi Media Center application from the desktop. You should see the welcome screen. Select the main System option, then select the System menu under the Confluence tab. Select the Audio Output menu item, and click on the Audio Output device selection on the top. You will be presented with all possible options. Select the option corresponding to the CM108 device, as shown in the screenshot below. Next, unmute the system using your C1-compatible remote or the keyboard.

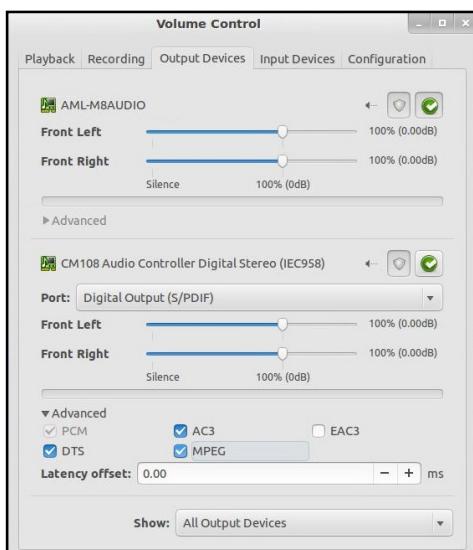
Download an mp3 file to a directory accessible by your account. Go back to the Home screen of the Kodi application. Select the Music option and click on the Music sub-option. Browse the directory that has the mp3 file, select the mp3 file and play it.



PulseAudio Volume Control

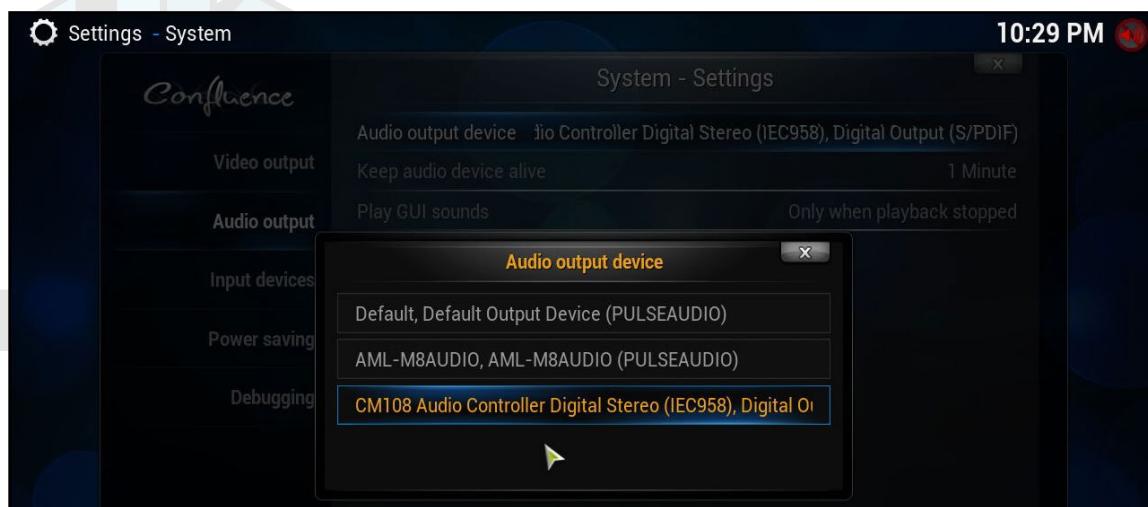


PulseAudio profile



PulseAudio output device config

Chapter 4



Kodi audio output device configuration

USB-CAM 720P

Hardkernel offers a USB-CAM rated @720p up to 30 fps (figure 4.43), which is certified to work with the C1. However, quite a few of the plug-and-plug USB webcams compatible with Microsoft Windows 7+, can also be expected to work with C1. The VF0610 Live! Cam Socialize HD (figure 4.44) is one such compatible model, details of which are shown below:

```
$ lsusb
...
Bus 001 Device 005: ID 041e:4080 Creative Technology, Ltd
...
```

Additional details can be found using the following command:

```
$ sudo lsusb -D /dev/bus/usb/001/005
Device: ID 041e:4080 Creative Technology, Ltd
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         2.00
  bDeviceClass      239 Miscellaneous Device
  bDeviceSubClass     2 ?
  bDeviceProtocol     1 Interface Association
  bMaxPacketSize0      64
  idVendor        0x041e Creative Technology, Ltd
  idProduct        0x4080
  bcdDevice         0.21
  iManufacturer       1 Creative Technology Ltd.
  iProduct           4 VF0610 Live! Cam Socialize HD
  iSerial            3 091006_A_01170
  bNumConfigurations    1
Configuration Descriptor:
  bLength          9
```

Chapter 4

bDescriptorType 2
wTotalLength 1087
bNumInterfaces 4
bConfigurationValue 1
iConfiguration 0
bmAttributes 0x80
(Bus Powered)
MaxPower 250mA
Interface Association:
bLength 8
bDescriptorType 11
bFirstInterface 0
bInterfaceCount 2
bFunctionClass 14 Video
bFunctionSubClass 3 Video Interface Collection
bFunctionProtocol 0
iFunction 4 VF0610 Live! Cam Socialize HD

...



USB-CAM 720p



VF0610 USB-CAM 720p

Attach the USB-CAM to an available USB port on the C, and wait for a few moments. Then, run the above commands to ensure that the device and its capabilities are detected accurately. Start Camorama with the camera pointed to any well-lit object. An image of that object should be visible in the display. The image below shows such a screen displaying the view seen through the attached camera:



USB-CAM 720p usage via Camorama application

Chapter 4

UPS2

There are many situations when an Uninterrupted Power Supply (UPS) is essential to ensure that the C1 is functioning well, even with loss of main power. Hardkernel has developed a 3000mAh UPS kit as shown below. This LiPo battery kit, when used in conjunction with the C1 charges the battery pack. The second image shows the fully assembled kit mounted on a C1.



UPS2 kit



UPS2 assembled

The UPS is designed that during its use, if the main power source fails, it takes over, presuming it is fully charged. After the battery has drained to a low level, it signals the C1 through a digital output pin, triggering a graceful shutdown. Since the cache/file operations are likely to be flushed out before the power is turned off, the risk of data loss is low.

USB GPS Module

Hardkernel produces a 5V 0.1A GPS receiver with an USB interface that supports the standard National Marine Electronics Association (NMEA) GPS protocol, as shown below.



USB GPS module

The device uses a Ublox 6010 chipset, with support already built

Chapter 4

into the official Hardkernel Lubuntu image. One should be able to attach the device to an available USB port on the C1 and start using it right away:

```
$ lsusb  
...  
Bus 001 Device 005: ID 1546:01a6 U-Blox AG  
...  
$ sudo ls -lsa /dev/ttyA*  
0 crw-rw---- 1 root dialout 166, 0 Dec 31 1979 /dev/ttyACM0
```

Now that the relevant TTY port is obtained, you can check to see if the GPS receiver is doing its job, using the following command:

```
$ sudo cat /dev/ttyACM0 | grep GPRMC  
$GPRMC,161053.00,A,3719.54074,N,12201.49867,W,0.079,,110415,,  
,A*65  
$GPRMC,161054.00,A,3719.54074,N,12201.49867,W,0.085,,110415,,  
,A*65  
$GPRMC,161055.00,A,3719.54074,N,12201.49867,W,0.024,,110415,,  
,A*66
```

The RMC - NMEA has its own version of essential GPS pvt (position, velocity, time) data, which represents the following information:

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W  
*6A
```

RMC	Recommended Minimum sentence C
123519	Fix taken at 12:35:19 UTC
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
022.4	Speed over the ground in knots
084.4	Track angle in degrees True
230394	Date - 23rd of March 1994
003.1,W	Magnetic Variation
*6A	The checksum data, always begins with *

To test the higher level functionality of the GPS dongle, we can use the services of gpsd, a service daemon, that monitors one or more GPS modules and makes the pcv (position, course, velocity) data available via the TCP port 2947 of the host system.

Install gpsd and relevant utilities using the following command:

Chapter 4

```
$ sudo apt-get install gpsd gpsd-clients foxtrotgps
```

Then, configure gpsd using the following command and options, then reboot:

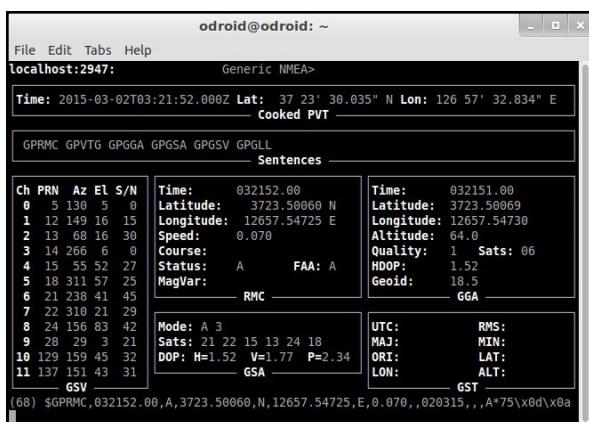
```
$ sudo dpkg-reconfigure gpsd
```

- Q. Start gpsd automatically?
<Yes>
- Q. Should gpsd handle attached USB GPS receivers automatically?
<Yes>
- C. Device the GPS receiver is attached to:
`/dev/ttyACM0`
<OK>
- c. Options to gpsd:
<blank>
- c. gpsd control socket path:
`/var/run/gpsd.sock`

Run the gpsmon command:

```
$ gpsmon
```

You should see a screen like that shown in the screenshot below. The \$GPRMC lines will scroll at the bottom of the screen, which indicates that the application is actually receiving the GPS information from the gpsd daemon.



```
odroid@odroid: ~
File Edit Tabs Help
localhost:2947: Generic NMEA>
Time: 2015-03-02T03:21:52.000Z Lat: 37 23' 30.035" N Lon: 126 57' 32.834" E
          Cooked PVT
GPRMC GPTVG GPGGA GPGSA GPGSV GPGLL
Sentences
Ch PRN Az El S/N Time: 032152.00 Time: 032151.00
0 5 130 5 0 Latitude: 3723.50069 Latitude: 3723.50069
1 12 149 16 15 Longitude: 12657.54725 E Longitude: 12657.54730
2 13 68 16 30 Speed: 0.070 Altitude: 64.0
3 14 266 6 0 Course: 155.5 Sats: 06
4 15 55 52 27 Status: A FAA: A HDOP: 1.52
5 18 311 57 23 MagVar: RMC Geoid: 18.5
6 21 238 41 45 Mode: A 3 GGA
7 22 310 21 29 Sats: 21 22 15 13 24 18
8 24 156 83 42 UTC: RMS:
9 28 29 3 21 MAJ: MIN:
10 129 159 45 32 ORI: LAT:
11 137 151 43 31 DOP: H=1.52 V=1.77 P=2.34 LON: ALT:
      GSA      GST
(68) $GPRMC,032152.00,A,3723.50069,N,12657.54725,E,0.070,,020315,,,A*75\x0d\x0a
```

Gpsmon display

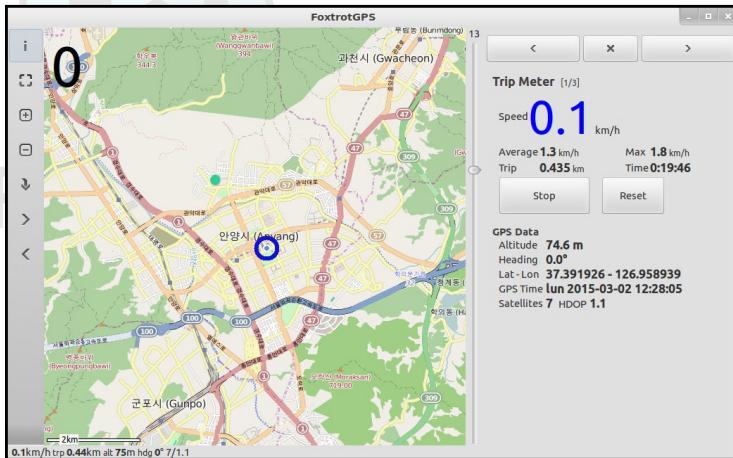
From a terminal window, launch the foxtrotgps application using the following command:

```
$ foxtrotgps
```

Note that the display of a live map requires the presence of a working internet connection. If you wish to see a real-time map while

Chapter 4

you are driving in a vehicle with this setup, you will need to use your smartphone as a hot-spot and have the setup talk to it via WiFi.



Foxtrotgps display

Conclusion

The ODROID-C1 is compatible with many types of hardware gadgets, and nearly any USB sensor may be used as long as it has a Linux driver available. The ones sold by Hardkernel at <http://bit.ly/1fbE91d> have the advantage of having pre-configured drivers included with the official Hardkernel disk images. We hope you enjoy tinkering and building your own projects using some of the techniques described here.

Additional Resources

ODROID Forums: <http://forum.odroid.com>

ODROID-C1 Wiki: <http://bit.ly/1KRK0GV>

Android images: <http://bit.ly/1FRHJK1>

Linux images: <http://bit.ly/1dFLsQQ>

Improved Win32 DiskImager: <http://bit.ly/1lYQ7MF>

Hardkernel store: <http://bit.ly/1fbE91d>

ODROID-C1+

The ODROID-C1+, released in August 2015, offers several new features and improvements over the original ODROID-C1:

- **HDMI connector was changed to Type A from Type D**

The Type A connection allows a standard HDMI cable to be used with any compliant HDMI monitor without the need for a Type D adapter or cable.

- **Expanded microSD card compatibility**

The original C1 model worked optimally with a specific set of microSD cards, and using a non-compliant card would sometimes result in a slow-booting or non-booting image. The C1+ is compatible with a wider array of microSD cards.

- **Improved CEC functionality**

The CEC hardware has been updated to operate regardless of the presence of an RTC backup battery.

- **USB OTG power path**

A small hardware modification could be performed on the original C1 model to allow it to be powered via the USB OTG port. This modification is now standard with the C1+ model by enabling the J8 jumper.

- **I2S signals exposed for external audio DAC**

The I2S signals can be easily accessed on the C1+ model in order to permit the use of an external audio Digital-To-Analog Converter (DAC).

