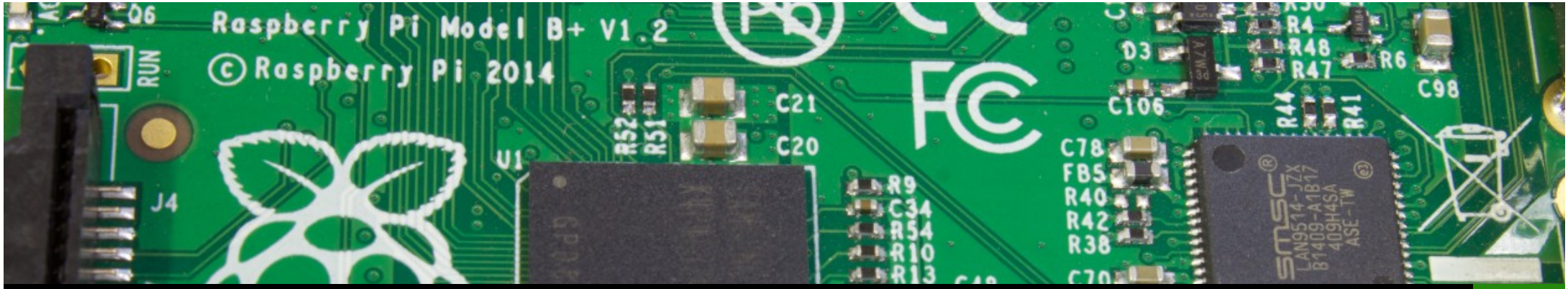

Andrew's blog

ABOUT ME

UNCATEGORISED

RASPBERRY PI ZERO – PROGRAMMING OVER USB!

24TH DECEMBER 2015 GBAMAN 8 COMMENTS

Full credit for the initial documentation for this goes to DaveB off the Raspberry Pi Forums, see his post here.

An acknowledgement also to Lady Ada of Adafruit who was working on documenting in parallel to myself on Christmas eve. She has taken a different approach which requires a UART serial cable. Her guide can be found here.

What is this?

The Raspberry Pi Zero is a very cool little computer. At £4, is pretty amazing for the price. But one thing many Pi users have wanted to be able to do for a long time is program their Raspberry Pi from another computer, **using only a single USB cable!**

The Raspberry Pi Zero is able to do this, hardware wise anyway, although a lack of software support was holding up it ever happening. Until now!

So what can it do?

Using the Linux USB Gadget modules, we can get the Pi Zero to emulate a whole host of USB devices including

- Virtual Serial – So can get a serial connection into the Pi, similar to using the UART pins. You can use Putty (on Windows) or Screen (on Mac and Linux).
- Virtual Ethernet – You can get your Pi Zero to appear as a USB Ethernet modem. With a little configuration, you can then get full SSH, VNC, FTP etc.
- Mass storage device – You can get the Pi Zero to appear as a flash drive, allowing you to for example, copy files over and have the Pi run these files (useful for robotics for example)
- Virtual MIDI – The Pi Zero could appear as a virtual MIDI instrument.
- Virtual Audio – The Pi Zero could appear as a virtual headphone output or microphone input.
- Virtual Human Interface Device (HID) – The Pi Zero could appear

as a virtual HID, for example a keyboard or mouse. So when you plug it into your computer, it could start typing!

You can also combine a few of the above (up to 3 at a time) using the g_multi module, although Windows and Mac have difficulty handling it then.

Examples

Where is this useful? Lets take an example of some Raspberry Pi robots in a classroom. Although you could be using Wifi for each robot, remembering addresses, unreliability with wifi etc all make wifi a bit of a rubbish answer.
With this, the student simply plugs in the robot and perhaps drops their script onto the flash drive that appears. When they unplug it, the robot runs the script, dumping the results of the script back onto the mass storage device, ready to be read when it is plugged in next.

Another example, lets say you don't have access to a screen to use with your Raspberry Pi Zero, in for example, a school. You could still let your students play with GPIO or Linux by simply using the serial module (with Putty or Screen), or the virtual ethernet module to allow them SSH access.

How to I set it up on my Raspberry Pi?

I have thrown together a guide over on Github Gists, including links to the downloadable precompiled kernels.

Raspberry Pi Zero OTG Mode

Simple guide for setting up OTG modes on the Raspberry Pi Zero - By Andrew Mulholland (gbaman).

The Raspberry Pi Zero (and model A and A+) support USB On The Go, given the processor is connected directly to the USB port, unlike on the B, B+ or Pi 2 B, which goes via a USB hub. Because of this, if setup to, the Pi can act as a USB slave instead, providing virtual serial (a terminal), virtual ethernet, virtual mass storage device (pendrive) or even other virtual devices like HID, MIDI, or act as a virtual webcam!
Below are a serial of precompiled kernels with instructions for each. It is recommended to go for the Modular kernel below, as it has support for all the Linux Gadget USB modules. It also has full

standard USB support, where as some of the non modular ones do not.

Note - If wanting to use the SD card on a Raspberry Pi 2, comment (add a hash in front of) the `device_tree=` line on the config.txt file on the boot partition.

Modular (all modules) - Recommended

Kernels - https://dl.dropboxusercontent.com/u/1122948/temp/PiOTG-Test/PiZeroCombined.tar.gz

Modules included

- Serial (g_serial)
- Ethernet (g_ether)
- Mass storage (g_mass_storage)
- MIDI (g_midi)
- Audio (g_audio)
- Mass storage and Serial (g_acm_ms)
- Ethernet and Serial (g_cdc)
- Multi (g_multi) - Allows you to configure 2 from Ethernet,

Mass storage and Serial

Note This section includes standard USB support as well, so can use normal USB devices (like a keyboard) throughout. It should auto switch when plugged into another computer.

1. First, flash Jessie (full, not lite) onto a blank microSD card.
2. Copy the tar.gz file onto the boot partition of the image.
3. Start it up, at this stage should be fine with any Pi, including Pi2.
4. Do normal first boot stuff (resize image etc with `sudo raspi-config`).
5. Run `tar xvzfC /boot/PiZeroCombined.tar.gz /tmp/`.
6. Then run `sudo cp -R /tmp/PiZeroCombined/fat32/* /boot/` (to copy the kernel and device tree stuff)
7. Then run `sudo cp -R /tmp/PiZeroCombined/ext4/lib/* /lib/` (to copy the kernel modules themselves)
8. At this point, you have 2 options, you can either set the module you want to run on startup, or dynamically load and unload them.
9. To set a module to load on startup, edit the modules file with `sudo nano /etc/modules` and add the module name you want to start (e.g. g_ether) to the end on a new line.

Also, make sure to the check the "Using the modules" section before rebooting.

10. Then, reboot the Raspberry Pi using `sudo reboot`.
11. To manually load a kernel module, after rebooting use `sudo modprobe moduleName`. To unload a module, use `sudo rmmod moduleName`.

Using the modules

- **g_serial** - To use the standard serial module, you need to tell the Pi to forward the serial console to it with `sudo systemctl enable getty@ttyGS0.service`, then you can connect to the device via Putty or Screen.
- **g_ether** - Using virtual ethernet, you should simply be able to ssh into the address of your Raspberry Pi. To do this, there is a little extra configuration required though. There is a few ways we could set up the point to point networking. The proper way would be to set up a DHCP server on one of the ends. A far simplier was though is just to give the Raspberry Pi a fixed IP address. To do this, you will need to run `sudo echo -e "interface usb0 \nstatic ip_address=169.254.64.64" >> /etc/dhcpcd.conf`. You can then

access the Raspberry Pi Zero by connecting to `169.254.64.64`, or by using `raspberrypi.local` if your computer has Bonjour installed (Mac and most Linux OSs including Raspbian). Note this method does not support adding a fixed address to the cmdline.txt file. For that, you have to use the Ethernet only kernel below.

- **g_mass_storage** - To have your Pi Zero appear as a mass storage device (flash drive), first create a mini filesystem in a file on your Pi with `sudo dd if=/dev/zero of=/piusb.bin bs=512 count=2880` and set it up as a fat32 filesystem with `sudo mkdosfs /piusb.bin`. Then, when enabling it, add `file=/piusb.bin stall=0` onto the end, for example `sudo modprobe g_mass_storage  file=/piusb.bin stall=0`.

In theory, most USB devices should work alongside these kernels, to switch to USB OTG mode, simply don't use an OTG adapter cable and use a standard USB cable to plug your Pi Zero into another computer, it should auto switch.
If you have any auto switching issues, try any of the other sets of kernels below.

Note - Using these kernels below will disable standard USB on the

Pi Zero with that SD card, it is recommended to also have a USB UART serial adapter on hand in case you get locked out due to lack of keyboard.

Older kernels

USB Virtual Ethernet - No standard USB support (only Gadget)

Ethernet kernels download - https://dl.dropboxusercontent.com/u/1122948/temp/PiOTG-Test/PiZeroEthernet.tar.gz

1. First, flash Jessie (full, not lite) onto a blank microSD card.
2. Copy the tar.gz file onto the boot partition of the image.
3. Start it up, at this stage should be fine with any Pi, including Pi2.
4. Do normal first boot stuff (resize image etc with `sudo raspi-config`).

5. Run `tar xvzfC /boot/PiZeroEthernet.tar.gz /tmp/`.
6. Then run `sudo cp -R /tmp/PiZeroEthernet/fat32/* /boot/` (to copy the kernel and device tree stuff)
7. Then run `sudo cp -R /tmp/PiZeroEthernet/ext4/lib/* /lib/` (to copy the kernel modules themselves)
8. Next, you will need to manually asign a fixed IP to the Pi, you can do this by editing the cmdline.txt file (using `sudo nano /boot/cmdline.txt`) and add on the end `ip=169.254.64.64:::255.255.0.0`. This will allow you to connect the Pi using the address `169.254.64.64` from your computer.
9. Finally, reboot your Pi. It will appear on your computer as a virtual network card, make sure you also assign a fixed IP address on the PC side. (On Mac OS, a much easier way is go into settings, sharing and enable internet sharing from wifi etc to the Gadget device. Using this, you don't even have to have a fixed IP on each side as it will get one via DHCP, will be 192.168.2.3 usually)

USB Virtual Serial - No standard USB support (only Gadget)

Ethernet kernels download - https://dl.dropboxusercontent.com/u/1122948/temp/PiOTG-Test/PiZeroSerial.tar.gz

1. First, flash Jessie (full, not lite) onto a blank microSD card.
2. Copy the tar.gz file onto the boot partition of the image.
3. Start it up, at this stage should be fine with any Pi, including Pi2.
4. Do normal first boot stuff (resize image etc with `sudo raspi-config`).
5. Run `tar xvzfC /boot/PiZeroSerial.tar.gz /tmp/` .
6. Then run `sudo cp -R /tmp/PiZeroSerial/fat32/* /boot/` (to copy the kernel and device tree stuff)
7. Then run `sudo cp -R /tmp/PiZeroSerial/ext4/lib/* /lib/` (to copy the kernel modules themselves)
8. Finally, to enable the serial service, run `sudo systemctl enable getty@ttyGS0.service` .
9. Everything is now configured, shutdown the Pi `sudo shutdown now` and insert the card into your Pi Zero, plug in the USB data port into your computer and it should appear as a Gadget serial device.

PREVIOUS POST

Destination space at W5

8 THOUGHTS ON “RASPBERRY PI ZERO – PROGRAMMING OVER USB!”

Alan Mc

25TH DECEMBER 2015 AT 8:02 AM

It’s like opening the last door on an Advent calendar !

Thanks to all for this Christmas present which will make our favourite nano-computer even more accessible to everyone =o)

Bravo to all who’ve worked on both the idea and the technical side for PiZero USB OTG use : DaveB, Andrew M, Simon W, LadyAda and of course the RPFoundation for bringing us the Hardware in the first place !

Joyeux Noël et bonnes fêtes!

REPLY

Pingback: PiZero to Windows 10 USB Networking – Cymplecy (Simplesi)

Pingback: Walkthru of getting networking between PC using PiZero via USB – Cymplecy (Simplesi)

Jarle Teigland

26TH DECEMBER 2015 AT 10:11 AM

This is just fantastic work Andrew – huge potential , I've had a go at this, mind you with a limited 'skillset'. Bearing in mind it's still Christmas – would it be possible for you to do a walkthrough on setting it up for a Mac ?? Experiencing a few problems connecting – not sure what I'm doing wrong if at all ?? Apple is notoriously difficult to deal with in terms of 3rd party access / setups as it is so 'locked down' ;(

REPLY

mobluse

26TH DECEMBER 2015 AT 5:12 PM

It seems as if the last tar.gz-file is the same as the first. I have a Raspberry Pi Model A (and not Zero) and I then need files that are hard-coded to gadget mode, because the OTG-ID-pin is connected to ground, i.e. always host.

REPLY

Lewis Cowles (@LewisCowles1)

27TH DECEMBER 2015 AT 11:04 PM

I'd love to see where the sources and patches are being stored so I could compile myself, is this possible, or is this binary-only?

REPLY

★ gbaman

28TH DECEMBER 2015 AT 6:11 PM

There is very little extra code added, simply makes use of the Linux Gadget drivers and a small tweak to the device tree files

for the Zero.
There is now a pull request sitting ready to be merged with the 4.4.x kernel branch with all the changes – https://github.com/raspberrypi/linux/pull/1239

REPLY

Pingback: Program your Raspberry Pi Zero over USB – Raspberry Pi Pod

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Search …

RECENT POSTS

Raspberry Pi Zero – Programming over USB!

Destination space at W5

#MyFestivalOfLight 2015 competition entry

Light painting at the Festival of Light 2015

Mozilla festival 2015 is coming!

FOLLOW ME ON TWITTER

My Tweets

RECENT COMMENTS

Program your Raspberry Pi Zero over USB – Raspberry Pi Pod on Raspberry Pi Zero – Programming over USB!

gbaman on Raspberry Pi Zero – Programming over USB!

Lewis Cowles (@LewisCowles1) on Raspberry Pi Zero – Programming over USB!

mobluse on Raspberry Pi Zero – Programming over USB!

Jarle Teigland on Raspberry Pi Zero – Programming over USB!

ARCHIVES

December 2015

October 2015

September 2015

July 2015

March 2015

January 2015

December 2014

November 2014

October 2014

September 2014

May 2014

September 2013

May 2013

April 2013

March 2013

February 2013

CATEGORIES

Camera board

Computer Science education

Dots boards

Pibot

Raspberry Pi

Uncategorised

Uncategorized

META

Log in

Entries RSS

Comments RSS

WordPress.org

Proudly powered by WordPress