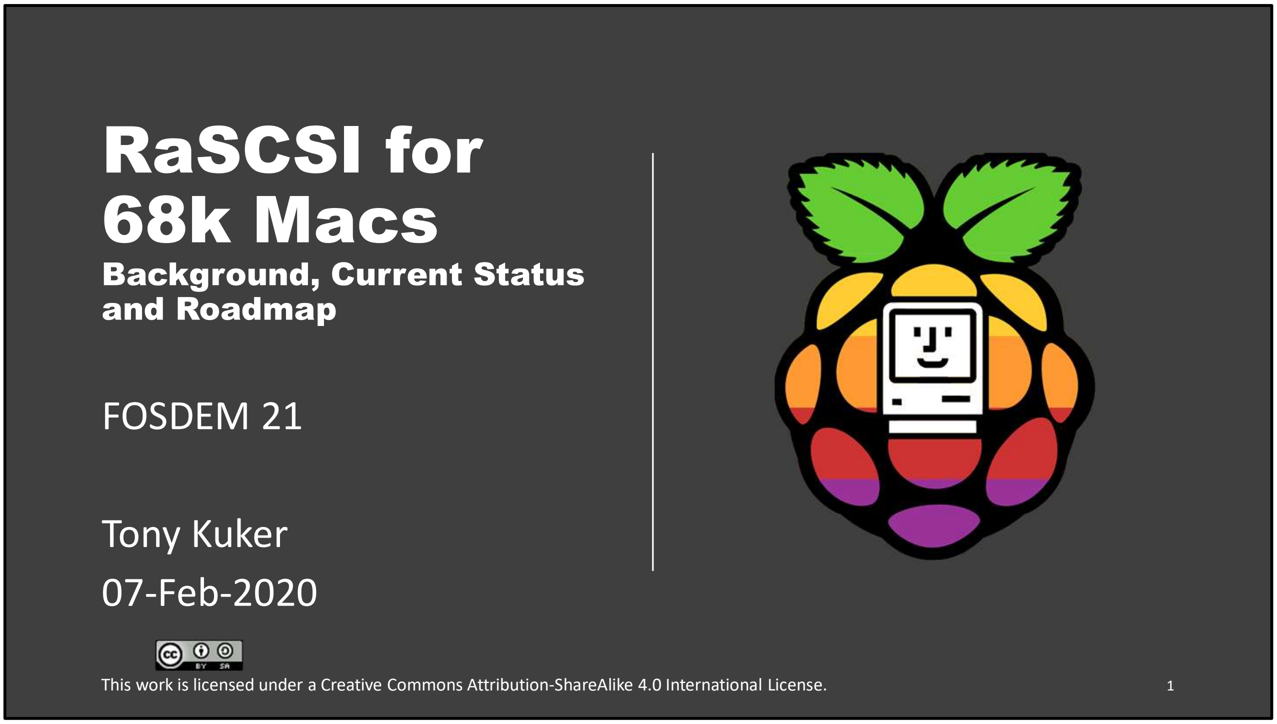


RaSCSI for 68k Macs

Background, Current Status and Roadmap

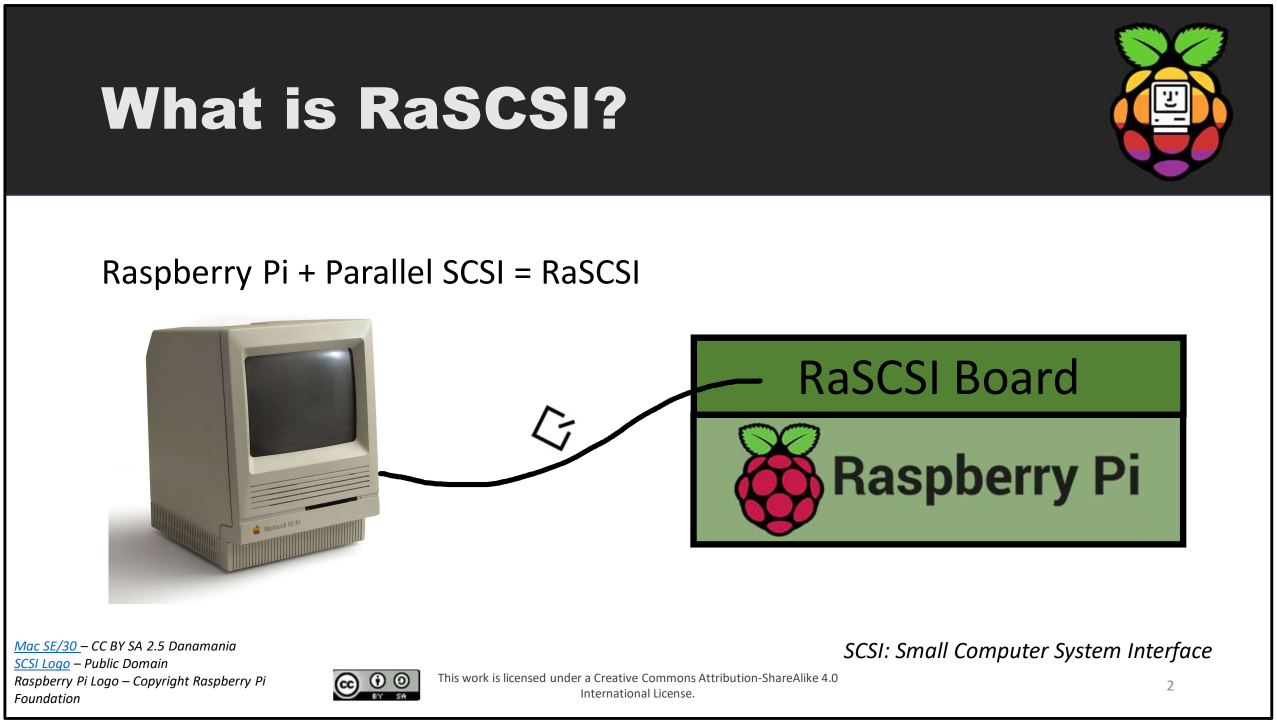
FOSDEM 21

Tony Kuker

07-Feb-2020

Introduction

- Tony Kuker
- Apple user since the 1980’s
- Day job as avionics systems / software engineer
- Been a fan of Rasperry Pi’s since they first appeared on the market
- Picked up working RaSCSI during the pandemic
    - Perfect marriage of interests



What is RaSCSI?

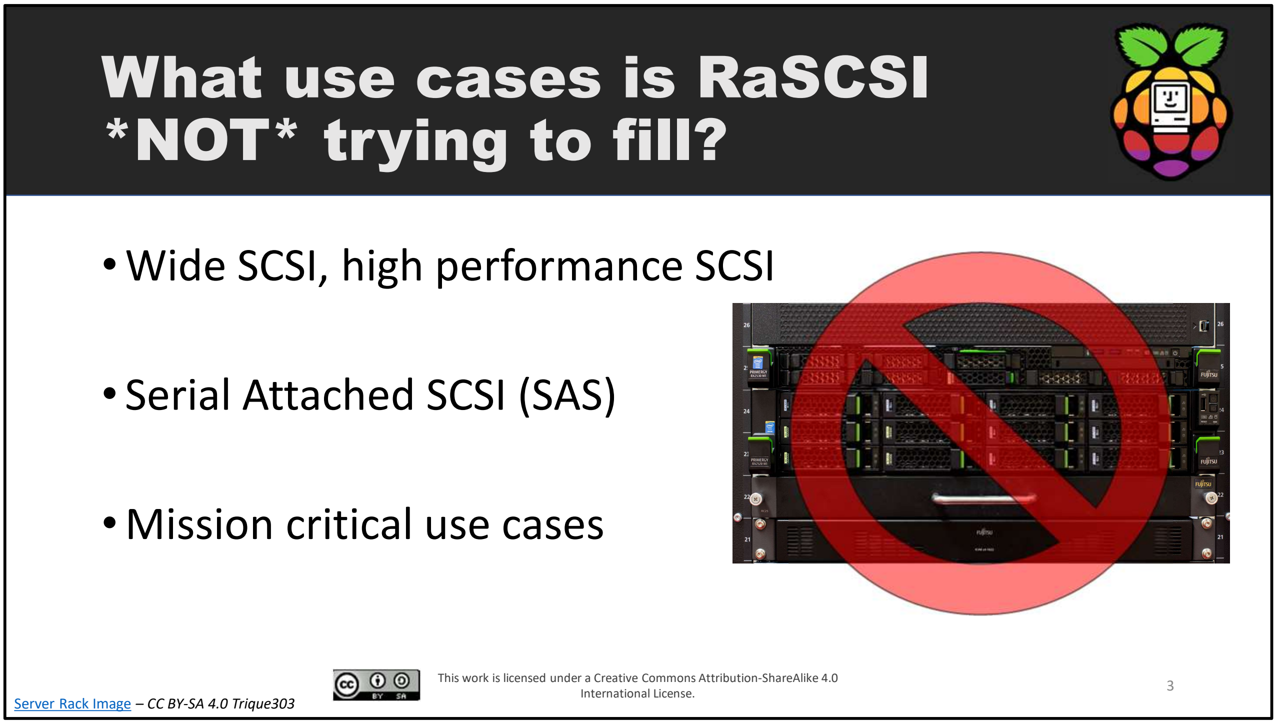
Raspberry Pi + Parallel SCSI = RaSCSI

Mac SE/30 – CC BY SA 2.5 Danamania
SCSI Logo – Public Domain
Raspberry Pi Logo – Copyright Raspberry Pi Foundation

SCSI: Small Computer System Interface

RaSCSI is:

- Interface board that allows software on the Raspberry Pi to read/write signals on a SCSI 1 or SCSI 2 bus
- Software service that runs on the Raspberry Pi that manages the communication
- “Bit bangs” the SCSI data on the bus
- SCSI isn’t extremely timing dependent
  - Request & Acknowledge signals allow for Raspberry Pi to communicate reliably
  - Based upon standard Linux distribution, so definitely NOT real-time

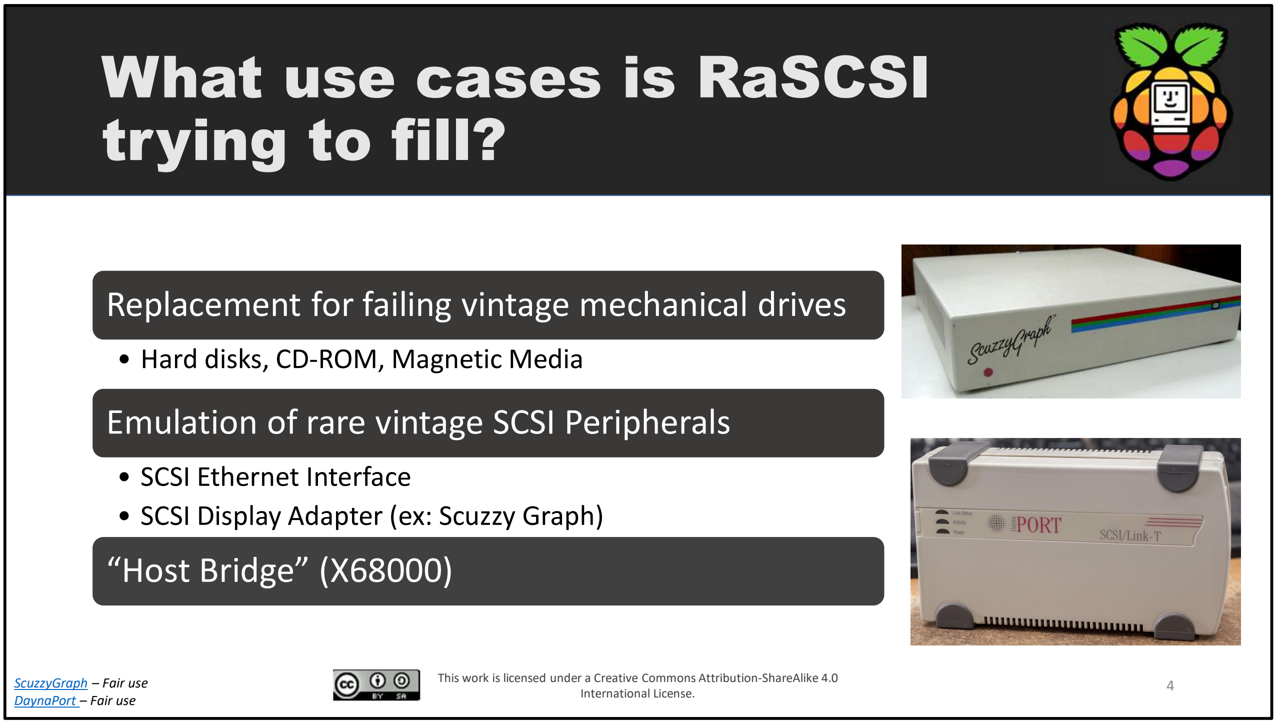


What use cases is RaSCSI *NOT* trying to fill?

- Wide SCSI, high performance SCSI
- Serial Attached SCSI (SAS)
- Mission critical use cases

Server Rack Image – *CC BY-SA 4.0 Trique303*

- Built upon standard Linux Raspberry Pi OS
- Performance is OK
  - Old SCSI devices relied on Z80 class processors
  - Gigahertz plus Raspberry Pi still does better
- Current RaSCSI does not do Wide SCSI
  - Not enough GPIO
- No Serial Attached SCSI
- Not recommended for mission critical use cases
  - Its open source
  - It’s a hobby project



What use cases is RaSCSI trying to fill?

Replacement for failing vintage mechanical drives

- Hard disks, CD-ROM, Magnetic Media

Emulation of rare vintage SCSI Peripherals

- SCSI Ethernet Interface
- SCSI Display Adapter (ex: Scuzzy Graph)

"Host Bridge" (X68000)

ScuzzyGraph – *Fair use*
DaynaPort – *Fair use*

- RaSCSI is not intended to be everything to everyone
- Use cases include:
    - Replacement for failing vintage mechanical drives on ***vintage*** computers
        - Hard Disk, CD-Rom, Magneto-Optical drives
    - Emulating rare/unique vintage SCSI devices
        - SCSI Ethernet interfaces – made for a while for devices with SCSI
            - DaynaPort SCSI/Link-T, NuvoLink, Asante
            - DaynaPort also available on the AtariST "FreeMint" OS, using Roger Burrows driver
            - Roger did a fantastic job figuring out most of the DaynaPort functionality
            - Big benefit of using Raspberry Pi – You have a full Linux network stack and flexible file system management
            - Microcontroller based solutions are more deterministic, but don't provide the swiss army knife for network configuration
        - Display adapters
            - Allowed a black and white Mac to use an external color display
            - Scuzzy Graphs are like unicorns – they are rumored to exist, but very very rare
            - Someday, hope to find one to emulate

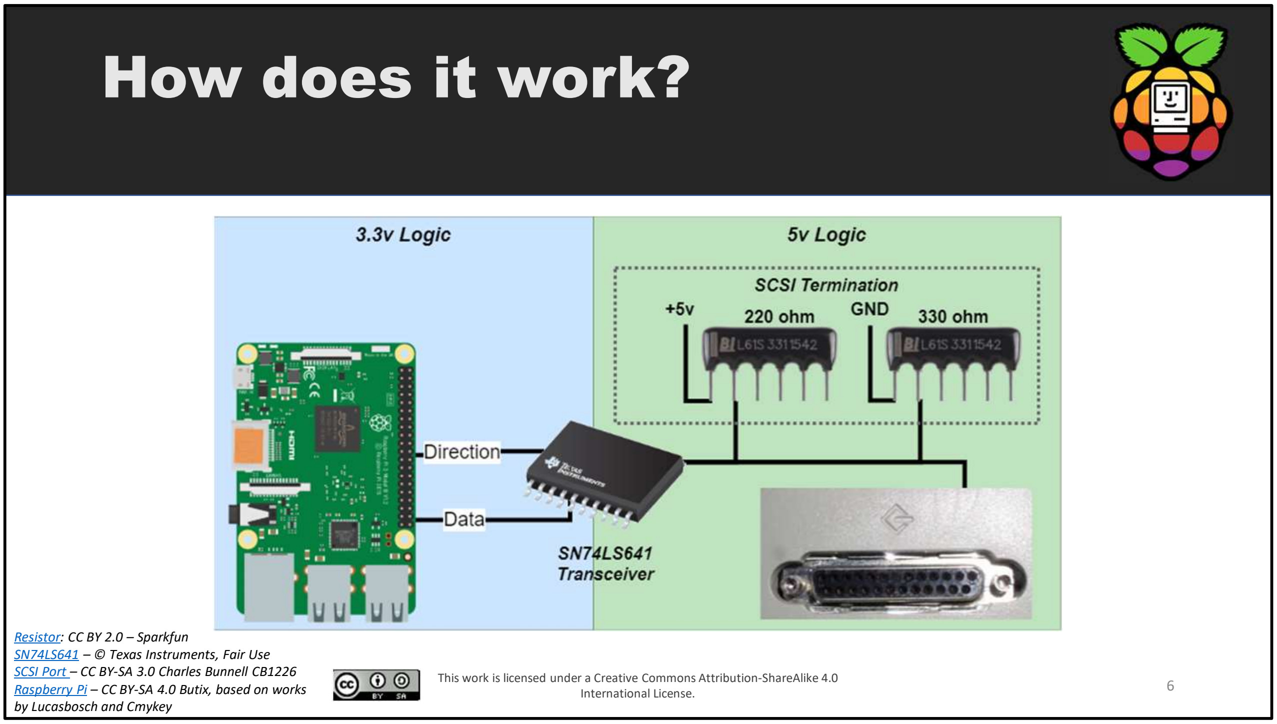
- “Host Bridge” for the Sharp X68000
    - Requires special driver written by Gimons
    - Currently only available for the Sharp X68000

Where has it successfully been used?

https://github.com/akuker/RASCSI/wiki/Compatibility

Mac Images licensed under CC-SA by Danamania
X68000 image licensed under CC BY-SA by Manupkp
AKAI Sampler – Public Domain

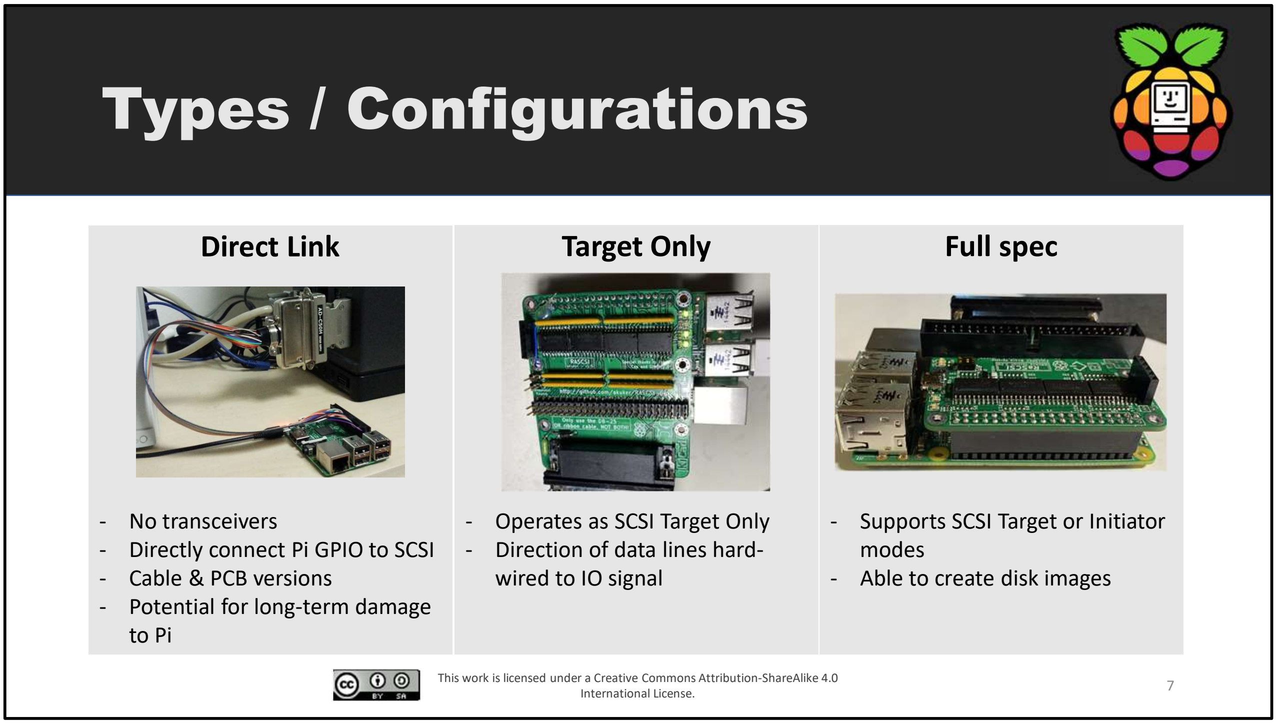
- Where has RaSCSI been used?
  - Originally developed for the Sharp X68000
    - 68kmla version is untested with Sharp X68000. One report that it does NOT work.
  - Many, many, many 68k and PowerPC Macintoshes
    - Mac Plus up to Power Mac G3
      - Mac Plus can not boot from RaSCSI, due to design of Mac Plus
    - Several SE/30 users
  - Akai Samplers
  - Ubuntu Intel i3 w/ PCI SCSI card
  - List of tested systems on the wiki



How does it work?

Resistor: CC BY 2.0 – Sparkfun
SN74LS641 – © Texas Instruments, Fair Use
SCSI Port – CC BY-SA 3.0 Charles Bunnell CB1226
Raspberry Pi – CC BY-SA 4.0 Butix, based on works by Lucasbosch and Cmykey

- Basic requirement for SCSI is that each end of the bus must have termination
  - Two options – Active or Passive
  - RaSCSI uses passive. 220ohm 5v pull-up and 330 ohm pull down resistor on each signal line
  - Around 3v on each signal while idle
- RaSCSI is built around the SN74LS641 bus transceiver from TI
  - Open collector or tri-state outputs, based upon the “direction” of the transceiver
  - When direction is output
    - Low output - grounds the signal and must be able to sink at least 46mA of current
    - High output – open
  - When direction is input, RaSCSI can read the current state of the signal
- RaSCSI GPIO is 3.3v logic



Types / Configurations

Direct Link

- No transceivers
- Directly connect Pi GPIO to SCSI
- Cable & PCB versions
- Potential for long-term damage to Pi

Target Only

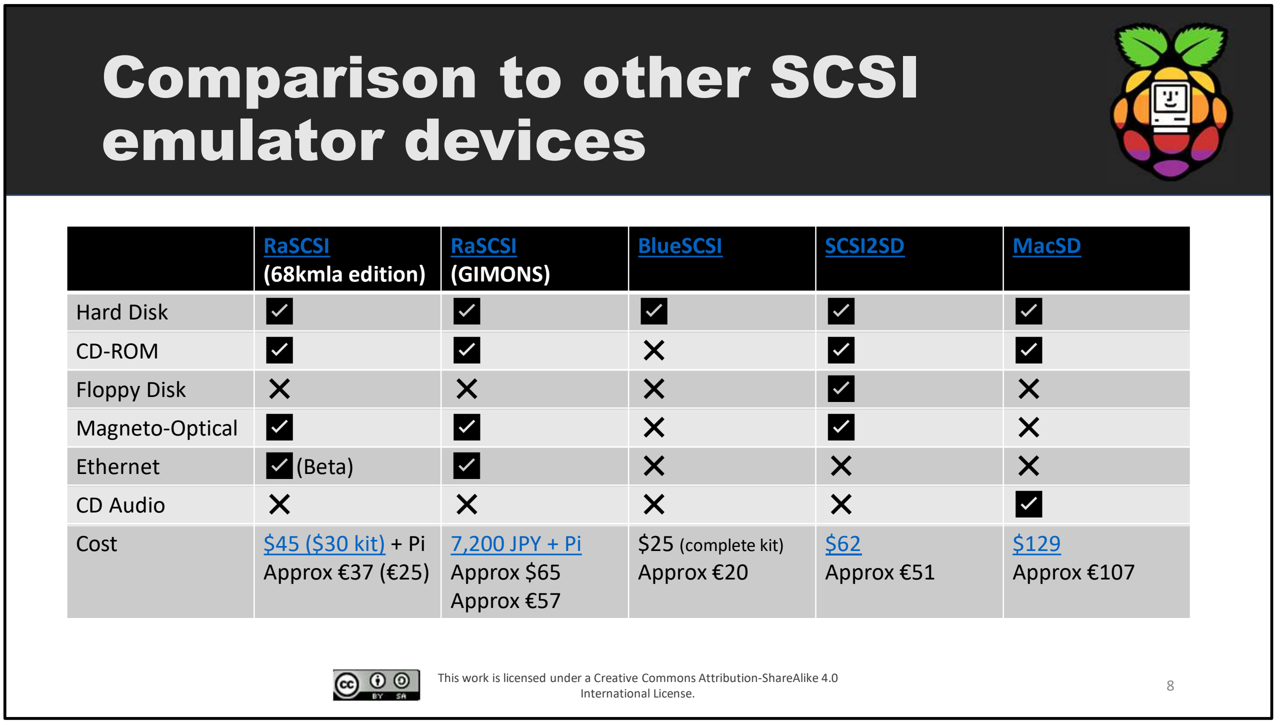
- Operates as SCSI Target Only
- Direction of data lines hard-wired to IO signal

Full spec

- Supports SCSI Target or Initiator modes
- Able to create disk images

- Direct Link
  - Easiest/simplest way to do connect RaSCSI
  - No transceivers – just connect the Raspberry Pi GPIO
  - Circuit boards are available, or just hard-wire a connector
  - Voltages of the signal/data lines are typically 0v or 3v, so this should work with Raspberry Pi GPIO
  - However, SCSI Spec requires 48 mA of current sink. Raspberry Pi is designed for ~18mA
  - Potential for damage to the Raspberry Pi, since using it outside of its design limits
- Target Only
  - Uses the 74LS641 transceivers
  - Direction of the transceivers is directly tied to the IO SCSI signal
  - Can only act as a “Target”
    - Can NOT initiate transactions
    - The Initiator needs to ask for data
- Full Spec
  - Similar to Target Only, but with more flexible transceiver direction control
  - Allows RaSCSI to “initiate” transactions
  - RaSCSI can act as a “host” and read drives

    - Most RaSCSIs are Full Spec – should say on the board
- ***Different software is needed, depending upon which type you have***
    - RaSCSI prints this out when it starts up

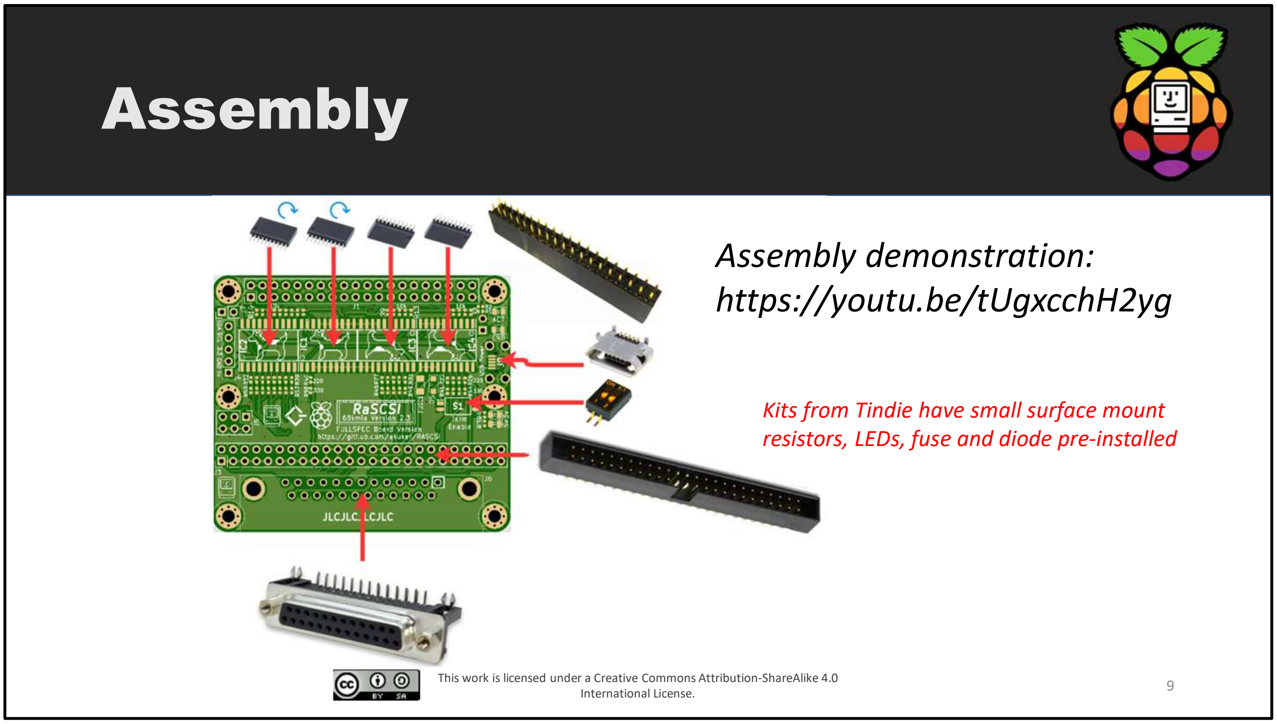


Comparison to other SCSI emulator devices

	RaSCSI (68kmla edition)	RaSCSI (GIMONS)	BlueSCSI	SCSI2SD	MacSD
Hard Disk	☑	☑	☑	☑	☑
CD-ROM	☑	☑	✕	☑	☑
Floppy Disk	✕	✕	✕	☑	✕
Magneto-Optical	☑	☑	✕	☑	✕
Ethernet	☑ (Beta)	☑	✕	✕	✕
CD Audio	✕	✕	✕	✕	☑
Cost	$45 ($30 kit) + Pi Approx €37 (€25)	7,200 JPY + Pi Approx $65 Approx €57	$25 (complete kit) Approx €20	$62 Approx €51	$129 Approx €107

- Different SCSI emulators have different benefits
- RaSCSI
  - Energized developer community
  - Fully open source
  - Full Linux software stack available to add new features
- BlueSCSI – fork of ArdSCSino-stm32
  - Very low cost solution to replace failing hard disks
  - Fully open source
  - Collaboration with the RaSCSI community
  - If you need 20 of them, BlueSCSI is your best option
- SCSI2SD – Proven, stable, mature product
  - Been around for years
  - No current path to support Ethernet
  - Older versions were open source
  - SCSI2SD v6 has great performance, but closed source
- MacSD – relatively new
  - Supports CD Audio
  - Easy setup
  - Great documentation
  - Active developers adding new features.

- Closed source



Assembly

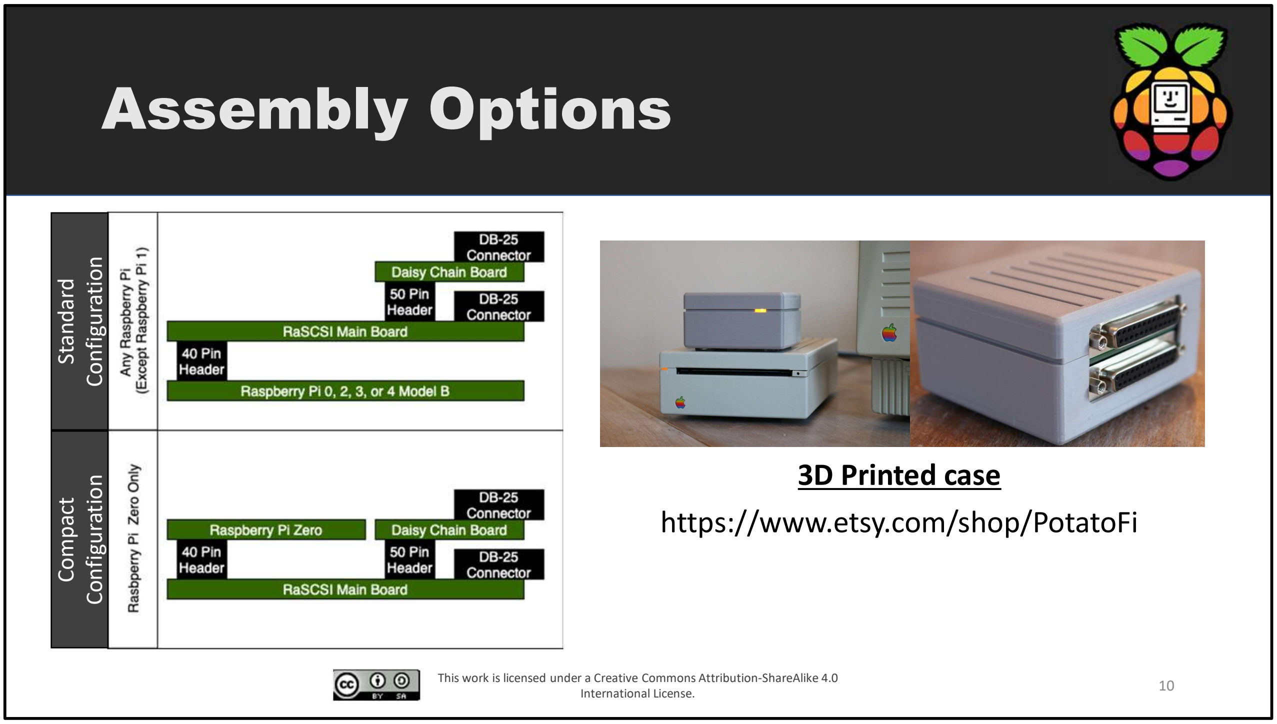
Assembly demonstration: https://youtu.be/tUgxcchH2yg

Kits from Tindie have small surface mount resistors, LEDs, fuse and diode pre-installed

9

- RaSCSI available pre-built or in kit form
- From Tindie, the 0402 resistors, LEDs fuse and diode are all pre-populated
  - Basically, whatever SMT parts JLCPCB has available
- Through-hole parts, connectors and transceivers are included with the kit
- Allows for flexibility in how you assemble everything
- PotatoFi has a video walking you through the process of assembling them
- Notice that IC1 and 2 are reversed from IC3 and 4

- Everything is fully open source – you can have your own PCBs made
  - Would not recommend manually soldering the 0402 resistors (unless you’re brave)
  - Could also modify board to use through-hole or just skip them completely and not have termination
  - Would need external terminator



Assembly Options

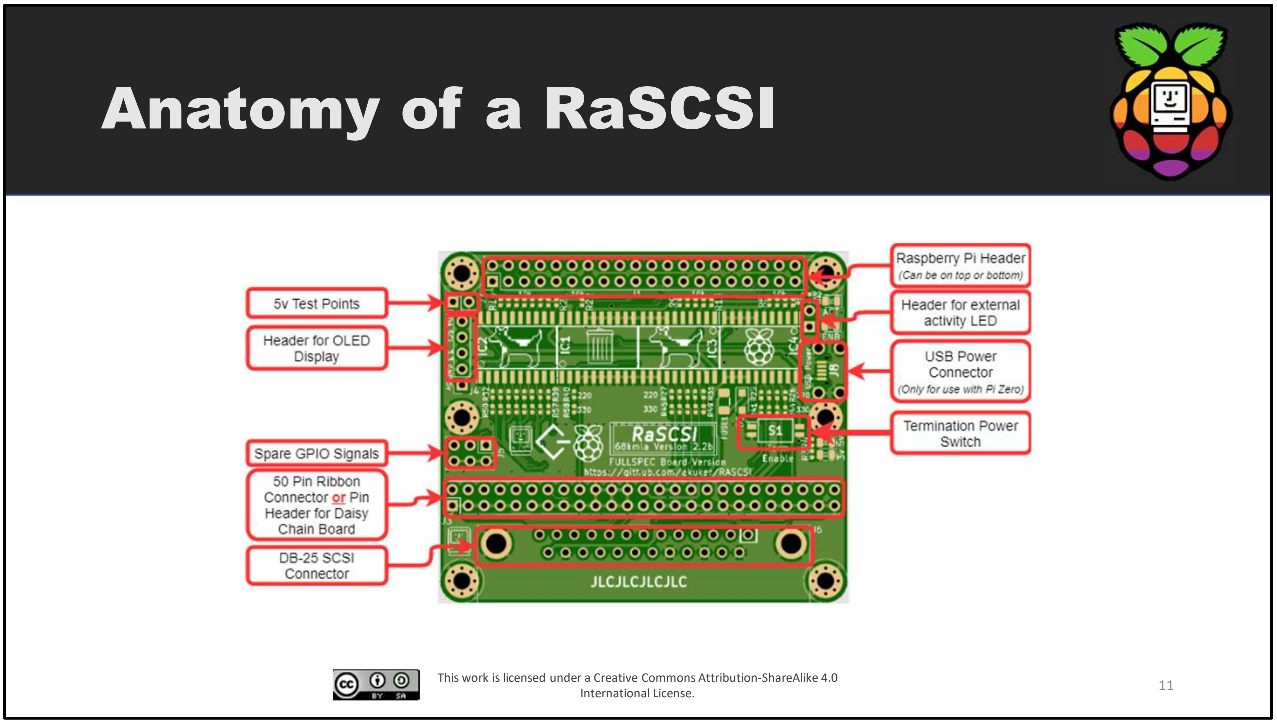
Standard Configuration
Any Raspberry Pi (Except Raspberry Pi 1)
DB-25 Connector
Daisy Chain Board
50 Pin Header
DB-25 Connector
RaSCSI Main Board
40 Pin Header
Raspberry Pi 0, 2, 3, or 4 Model B

Compact Configuration
Rasbperry Pi Zero Only
DB-25 Connector
Raspberry Pi Zero
Daisy Chain Board
40 Pin Header
50 Pin Header
DB-25 Connector
RaSCSI Main Board

3D Printed case

https://www.etsy.com/shop/PotatoFi

- With the kit, if you have a Raspberry Pi Zero without the header pre-soldered, you can install it above the RaSCSI board
- This allows for compact size but…
    - Prevents access to HDMI and power connector – Can use connector on RaSCSI to power everything
    - Will not work with other Raspberry Pis (without getting creative)
- Pre-build boards are all assembled in “Standard” configuration
- PotatoFi has created a 3D-Printable enclosure for the Compact configuration
    - You can download the model and print it yourself
    - OR, available on Etsy for a great price – he even includes an LED

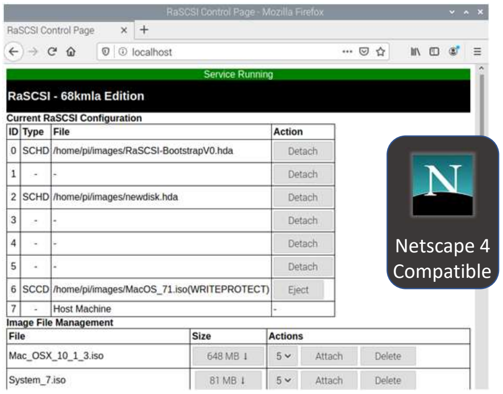


Anatomy of a RaSCSI

- The RaSCSI board has a lot crammed onto a small board
- At the top – header for connecting to the Raspberry Pi
    - Standard Raspberry Pi GPIO pinout
    - Should work with other DIY boards with compatible pinouts
- Header of OLED display
    - Optional piece for showing which drives are installed
- 50 pin ribbon connector
    - Can be used inside the computer as an internal drive
    - Or, can have the daisy-chain add-on board connected to it
- 25 pin SCSI connector
    - Standard SCSI connector used on most Macintosh desktop computers
- Termination Power Switch
    - Allows you to turn on/off termination power
    - Both switches should be turned ON or OFF
- USB Power Connector
    - For compact configuration, this is needed to power the Raspberry Pi
    - The Pi Zero's connector can be blocked by the Daisy Chain board
    - Not needed for standard configuration
- External activity LED header

How to Control RaSCSI

Web Interface



Service Running

RaSCSI - 68kmla Edition

Current RaSCSI Configuration

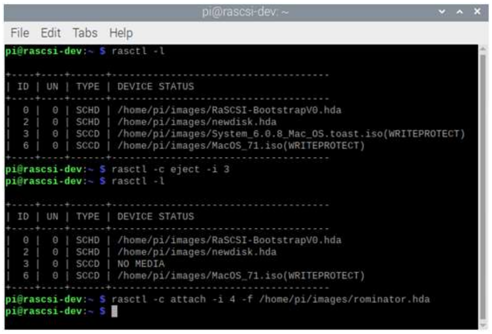
ID	Type	File	Action
0	SCHD	/home/pi/images/RaSCSI-BootstrapV0.hda	Detach
1	-	-	Detach
2	SCHD	/home/pi/images/newdisk.hda	Detach
3	-	-	Detach
4	-	-	Detach
5	-	-	Detach
6	SCCD	/home/pi/images/MacOS_71.iso(WRITEPROTECT)	Eject
7	-	Host Machine	-

Image File Management

File	Size	Actions
Mac_OSX_10_1_3.iso	648 MB ⇩	5 ⌄ Attach Delete
System_7.iso	81 MB ⇩	5 ⌄ Attach Delete

Netscape 4 Compatible

Command Line Utility



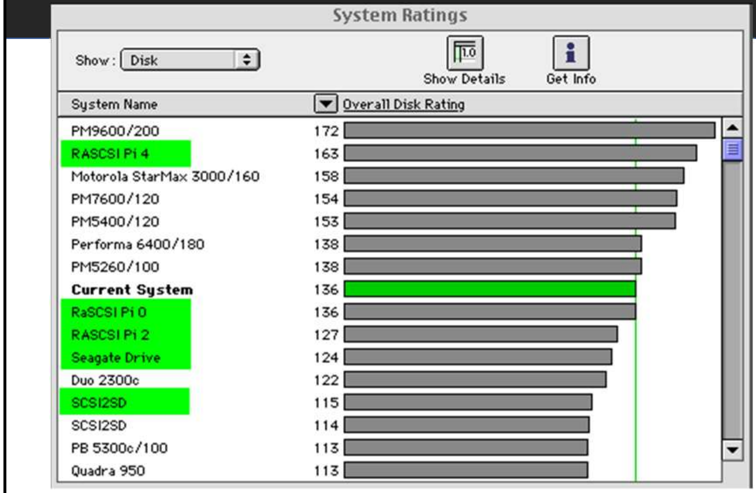
```
pi@rascsi-dev:~ $ rasctl -l

+----+----+------+-------------------------------------
| ID | UN | TYPE | DEVICE STATUS
+----+----+------+-------------------------------------
|  0 |  0 | SCHD | /home/pi/images/RaSCSI-BootstrapV0.hda
|  2 |  0 | SCHD | /home/pi/images/newdisk.hda
|  3 |  0 | SCCD | /home/pi/images/System_6.0.8_Mac_OS.toast.iso(WRITEPROTECT)
|  6 |  0 | SCCD | /home/pi/images/MacOS_71.iso(WRITEPROTECT)
+----+----+------+-------------------------------------
pi@rascsi-dev:~ $ rasctl -c eject -i 3
pi@rascsi-dev:~ $ rasctl -l

+----+----+------+-------------------------------------
| ID | UN | TYPE | DEVICE STATUS
+----+----+------+-------------------------------------
|  0 |  0 | SCHD | /home/pi/images/RaSCSI-BootstrapV0.hda
|  2 |  0 | SCHD | /home/pi/images/newdisk.hda
|  3 |  0 | SCCD | NO MEDIA
|  6 |  0 | SCCD | /home/pi/images/MacOS_71.iso(WRITEPROTECT)
+----+----+------+-------------------------------------
pi@rascsi-dev:~ $ rasctl -c attach -i 4 -f /home/pi/images/rominator.hda
pi@rascsi-dev:~ $ █
```

- RaSCSI can be controlled/configured two different ways – Web interface or command line
- Web interface is compatible with Netscape 4
  - Allows you to control directly from the device
  - Also works with newer browsers
  - You can upload disk images directly, download them, attach them
  - Allows for directly downloading a file from an abandonware site and copying it to an “ISO” that can be read by the Mac
- Rasctl is the command line utility that allows you to do the same functions as the web interface
  - Behind the scenes, the web interface is still using rasctl
  - Separate application – does not require root access
  - When you run it, it connects to a socket that is opened by the RaSCSI service
  - Rasctl sends the command and receives the response
- Unfortunately, no network security in place right now
  - Anyone that can access the Pi can change its SCSI configuration
  - Only recommended for use on closed networks

Benchmarks



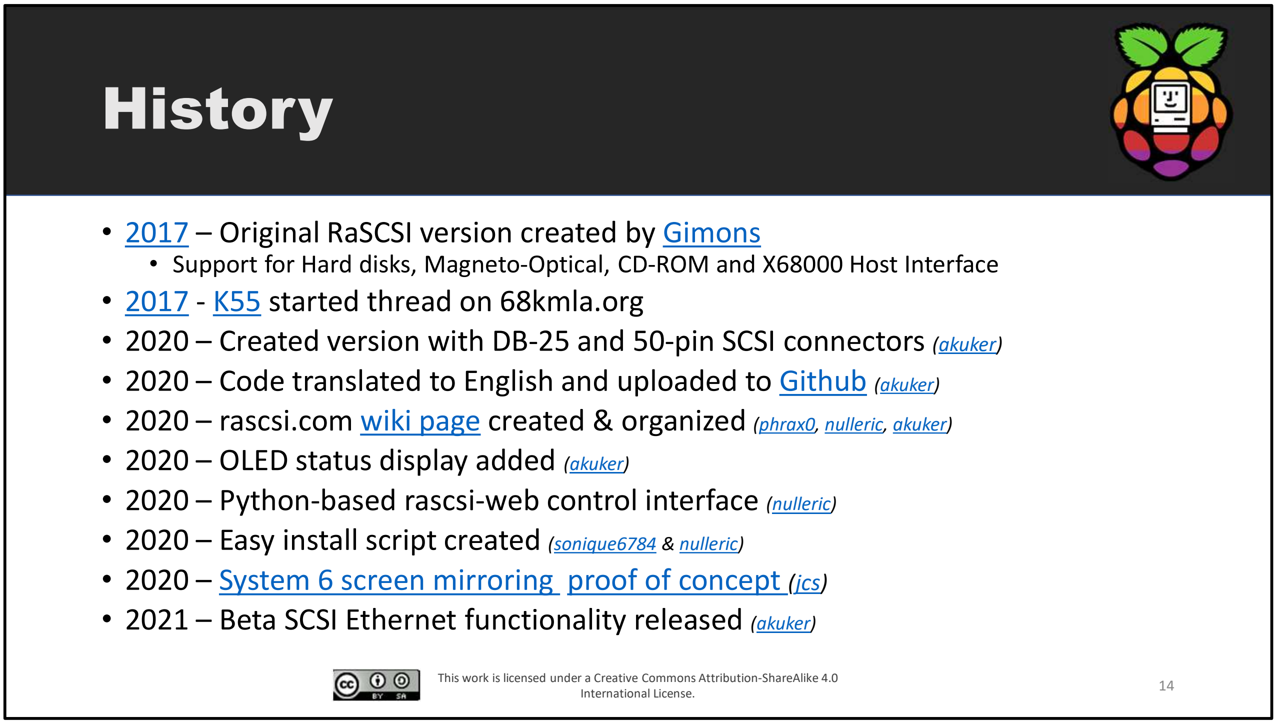
System Name	Overall Disk Rating
PM9600/200	172
RASCSI Pi 4	163
Motorola StarMax 3000/160	158
PM7600/120	154
PM5400/120	153
Performa 6400/180	138
PM5260/100	138
Current System	136
RaSCSI Pi 0	136
RASCSI Pi 2	127
Seagate Drive	124
Duo 2300c	122
SCSI2SD	115
SCSI2SD	114
PB 5300c/100	113
Quadra 950	113

Test platform:

Macintosh Quadra 840av

- 68040 Processor at 40MHz
- 128MB RAM
- 1MB VRAM
- Seagate ST3600N 500MB HD w/stock Apple firmware
- MacOS 8.1
- Drive cache configured at 128KB
- Norton System Info 3.5

- Different versions of Raspberry Pis do seem to have an impact on performance
  - Norton System Info was used to test different Raspberry Pi models, along with SCSI2SD v5 (because that’s all I have on hand)
  - Raspberry Pi 4 currently has the highest benchmarks
  - Even Raspberry Pi 0 had better performance than the stock Seagate spinning disk
  - Tests with SCSI2SD v6 were much higher. @nulleric recorded a score of 247 on a PowerMac G3 with a SCSI2SD v6.
    - The higher score was likely influenced by a much faster computer, as well as a much faster SCSI device
  - On slower machines with 68030 and slower processors, the SCSI device isn’t going to be the bottleneck
    - If you’re using a Macintosh SE, a Raspberry Pi Zero is plenty fast

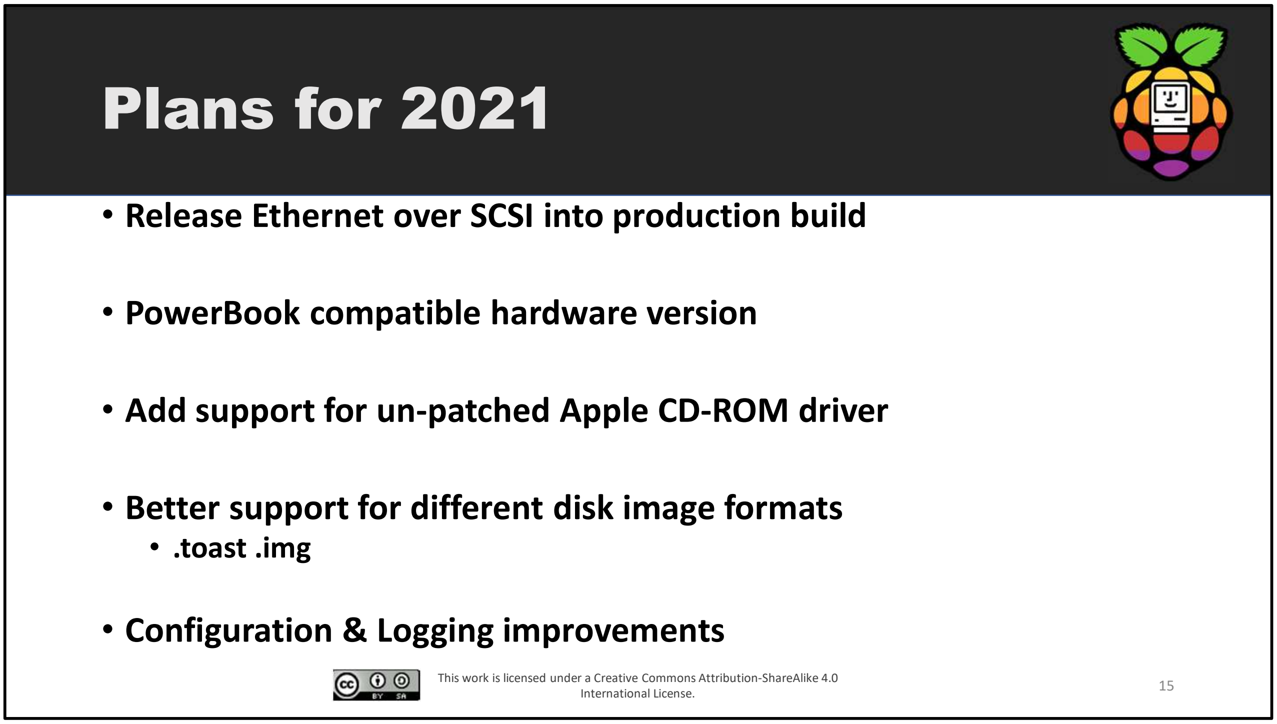


History

- 2017 – Original RaSCSI version created by Gimons
  - Support for Hard disks, Magneto-Optical, CD-ROM and X68000 Host Interface
- 2017 - K55 started thread on 68kmla.org
- 2020 – Created version with DB-25 and 50-pin SCSI connectors *(akuker)*
- 2020 – Code translated to English and uploaded to Github *(akuker)*
- 2020 – rascsi.com wiki page created & organized *(phrax0, nulleric, akuker)*
- 2020 – OLED status display added *(akuker)*
- 2020 – Python-based rascsi-web control interface *(nulleric)*
- 2020 – Easy install script created *(sonique6784 & nulleric)*
- 2020 – System 6 screen mirroring proof of concept *(jcs)*
- 2021 – Beta SCSI Ethernet functionality released *(akuker)*

- All credit for the hard work goes to Gimons. Original architect of the project
  - Published in 2017, maybe earlier?
  - Schematics and source code made available on his website
  - Originally intended for the Sharp X68000 computers
  - Developed the “host bridge” functionality
    - Allows X68000 to directly access Raspberry Pi’s file system
    - Also allows Ethernet access for the X68000
- Also in 2017, K55 started thread on 68kmla.org
  - Built a couple boards and provide that this could be used with vintage Macs
- In 2020 – I jumped in
  - Laid out board with DB-25 and 50 pin header connectors for more flexibility
  - Made daisy chain daughter board
  - “Forked” the code on Github
    - Gimons had released source code as zip files on his site
    - Initial version based on 1.47
  - Translated the C++ code comments to English – with lots of help from Google Translate
  - Simple python script to show the currently installed SCSI devices on a cheap OLED display
  - Translated ZTTO’s rascsi-php web interface to English

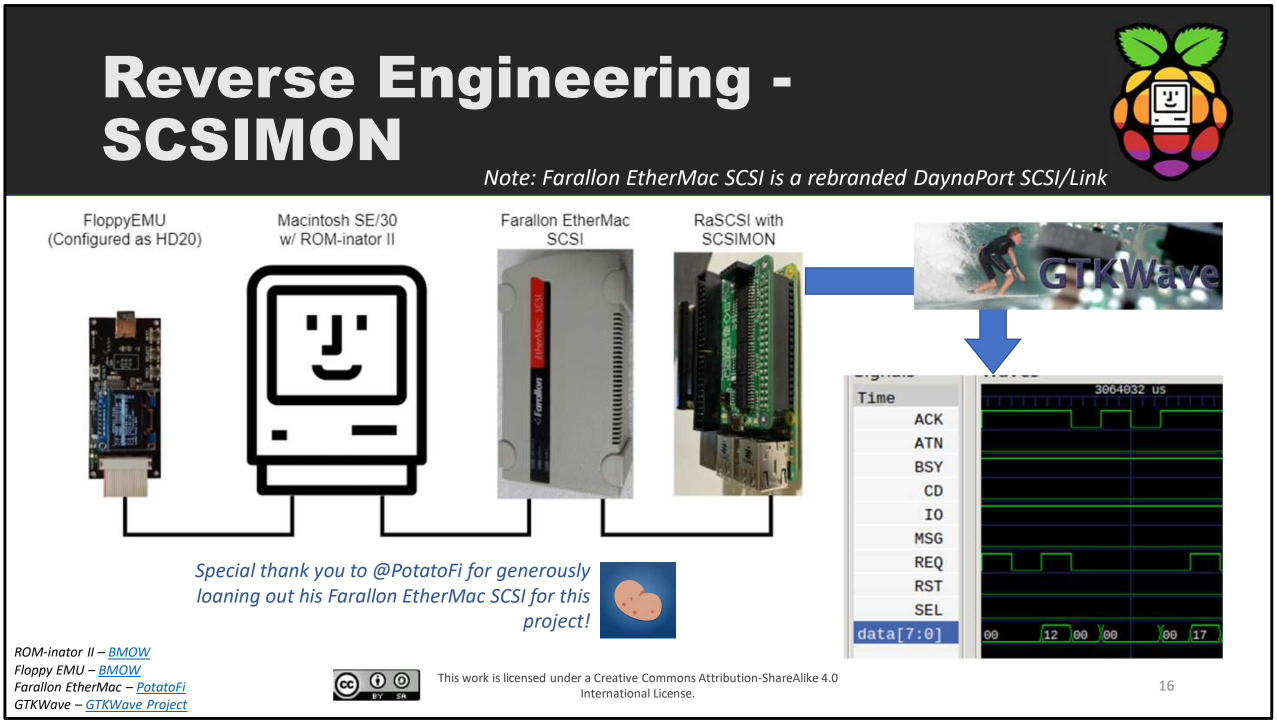
- Phrax and Nulleric joined in
    - Pulled together some fantastic documentation on github
- Nulleric created a Python-based web interface to control the configuration of RaSCSI
    - This allows direct control of the drives from the Mac
    - You can open Netscape 4, connect to the Raspberry Pi to:
        - Insert/eject CDs
        - Attach/detach drives
        - Check the status of the RaSCSI software
    - You don't need a separate device/interface to control RaSCSI
- October 2020 - Sonique and Nulleric created an easy install script. You can set up a new RaSCSI device with only a couple commands
- October of 2020 Joshua Stein created a screen mirroring proof of concept
- 2021 - DaynaPort SCSI/Link emulation released in beta form



Plans for 2021

- **Release Ethernet over SCSI into production build**
- **PowerBook compatible hardware version**
- **Add support for un-patched Apple CD-ROM driver**
- **Better support for different disk image formats**
  - **.toast .img**
- **Configuration & Logging improvements**

- 2021 – Keep the momentum going!
- Clean up DaynaPort SCSI/Link emulation and improve documentation
  - Create setup tools/instructions for wired and wireless networking
  - Pull software into main release build
- Improved regression test suite
- Release version with high density 40 pin connector for PowerBooks
  - 3D printable mounting mechanism
- Add zip drive emulation – I love my zip drive
- Allow using standard Apple CD driver
  - Right now, you need to use a patched version
- Support more image formats
  - Toast, Apple IMG files
- Improved configuration and logging
  - Method to save existing configuration so that it restores after a Pi reboot
- Move to a GitFlow branching model and establish version numbering



Reverse Engineering - SCSIMON

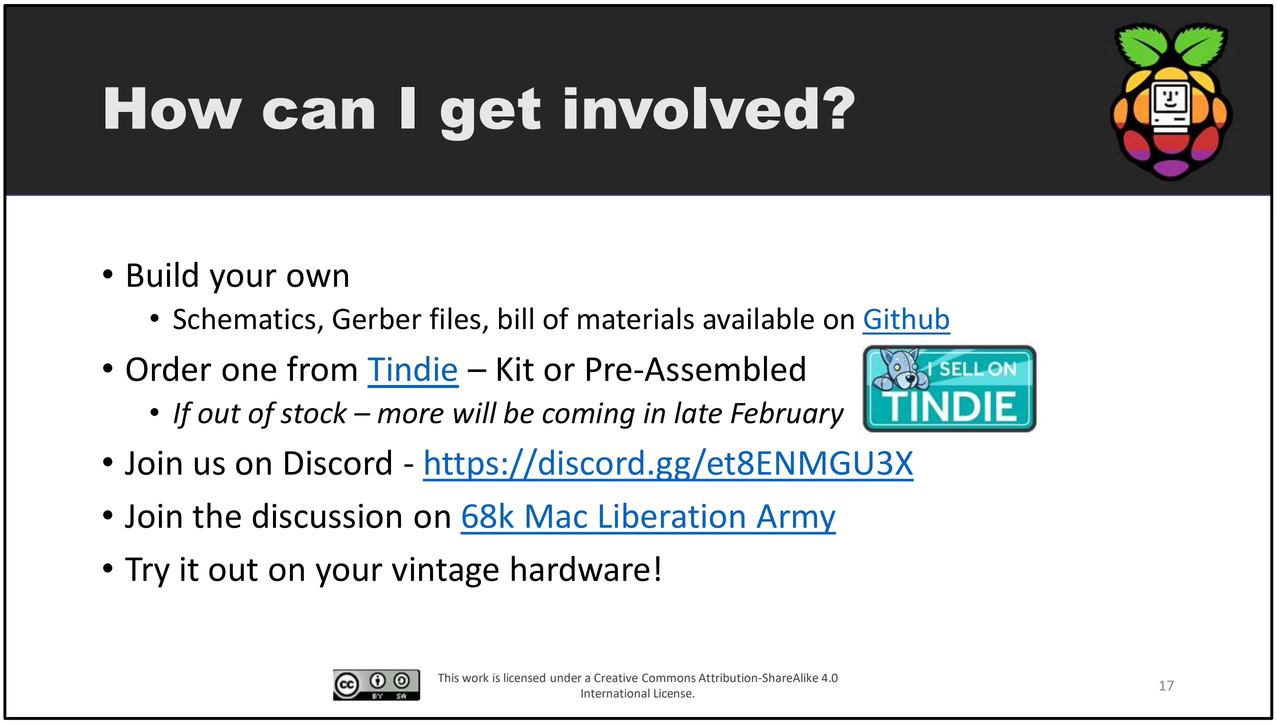
Note: Farallon EtherMac SCSI is a rebranded DaynaPort SCSI/Link

Special thank you to @PotatoFi for generously loaning out his Farallon EtherMac SCSI for this project!

ROM-inator II – BMOW
Floppy EMU – BMOW
Farallon EtherMac – PotatoFi
GTKWave – GTKWave Project

- Implementing Ethernet emulation required a way to monitor the SCSI bus traffic
- Budget didn’t allow for a fancy logic analyzer
    - It did allow for a second RaSCSI
- Developed SCSIMON tool
    - Command line tool that monitors the SCSI bus – read only
    - While its running, caches the data in memory
    - As soon as you stop, dumps the data in to a “VCD” file – value change dump
    - VCD can be opened with GTKWave to analyze the data
    - Went through a few iterations, but got down to ~80ns accuracy on the bus using a Raspberry Pi 4
- Another issue – booting from a SCSI drive introduced A LOT of extraneous traffic on the SCSI bus
    - Solution – Floppy EMU
    - Floppy EMU is able to emulate a HD20 hard disk, which connect via the floppy drive port
    - For the SE/30, this only works if you replace the stock ROM with ROM-inator
    - Able to boot from Floppy Emu with only the DaynaPort on the SCSI bus
        - Note: Farallon EtherMac is just rebranded DayanPort SCSI Link. EtherMac firmware even reports itself as a DaynaPort SCSI Link
    - No extraneous traffic!

- Using this setup, was able to capture the DaynaPort's traffic and compare it to RaSCSI's emulated DaynaPort traffic
- 2005 Roger Burrows provided excellent documentation of how this interface works
    - Was written from the perspective of the driver
    - Missing some critical details about how the hardware side was implemented
    - Once I had an actual device (Thanks PotatoFi), I was able to capture a lot of data and fill in the missing pieces



How can I get involved?

- Build your own
  - Schematics, Gerber files, bill of materials available on Github
- Order one from Tindie – Kit or Pre-Assembled
  - *If out of stock – more will be coming in late February*
- Join us on Discord - https://discord.gg/et8ENMGU3X
- Join the discussion on 68k Mac Liberation Army
- Try it out on your vintage hardware!

I SELL ON TINDIE

17

- To get involved
  - All of the design files are on Github
    - You can have your own boards fabricated using the existing design
    - Make updates to the design and have custom boards made
    - Please share improvements with the community
  - Order from Tindie
    - Kit version is only partially assembled, but includes everything you need to get started
      - You provide your own solder and tools
    - Assembled version is ready to go
    - International shipping – if your country is missing, let me know. I'll add it
    - Price structure is intended to get RaSCSI into the hands of as many people as possible.
  - Hang out on discord
    - Support channel to get help
    - Developers channel to talk low level techie stuff
    - Off-topic channel to talk about whatever is on your mind
    - Channels for other SCSI-related projects – BlueSCSI, 68net, scuznet (Nuvolink emulator)
  - Help us test RaSCSI on different hardware

- There are a lot of vintage platforms that support SCSI
- We'd love to learn how well RaSCSI works with Nextstation, SPARCstation, AtariST, Amiga, Apple II, etc