Case Study: Building an Out Of The Box Raspberry Pi Modular Synthesizer

LAC 2014 /// ZKM
Jürgen Reuter
reuter_j@web.de
The Idea

Headless Synth Modules

Network

Control Host
(Synth GUI & Configuration)
It's appealing since...

• ... it scales (distributed modular architecture),
• ... it looks cool (see all modules & wiring),
• ... you can touch all modules (WYSIWYG),
• ... you can turn, say, a low pass filter into a reverb effect just by replacing the SD card,
• ... bugs have only local impact.
Audio Relevant Sockets

- 10 / 100 Mbps Ethernet
- 2x USB 2.0
- 3.5mm Audio
- RCA Video
- HDMI
- GPIO / I2S
Audio Relevant Sockets

- 10 / 100 Mbps Ethernet
- 2x USB 2.0
- HDMI
- 3.5mm Audio
- RCA Video
- GPIO / I2S

Audio + Control Data ✔
High Bandwidth ✔
Bi-directional ✔
Digital ✔
Out-Of-Box Connectable ✔
Basic Setup

- Tested with NOOBS Raspian Wheezy (Debian 3.10.24+ #614 PREEMPT armv6l kernel)
- Detailed software installation guide → see paper
- 16GB class 4 SD card as HDD
Streaming Software Requirements

• Available for ARMv6
• Optimized for Low Latency
• Headless Use
• Candidates:
  – Network Audio System (NAS)
  – PulseAudio
  – JACK with netJACK
Streaming Software Requirements

• Available for ARMv6 ✓
• Optimized for Low Latency ✓
• Headless Use ✓

• Candidates:
  – Network Audio System (NAS)
  – PulseAudio
  – JACK with netJACK ✓
Module Software

• In this study: simple clients derived from JACK's `simple_client.c`

• Goal: Use existing plug-ins, e.g. LV²

• Missing: JACK client for remote export of plug-in lines via netJACK
Evaluation Setup

VCO

Master JACK Server

ALSA

n

n

n

n
CPU Load Against # of Channels

CPU Load on Master JACK Server

LAC 2014 @ ZKM
May 1, 2014
Raspberry Pi Modular Synthesizer
Displaying Routes with `qjackctl`
Displaying Routes with \texttt{qjackctl}

![Diagram showing qjackctl interface with routes and server names highlighted.](image)
Displaying Routes with `qjackctl`

![Diagram showing connections in qjackctl](image)

Remote Client Name ✗
Displaying Routes with \texttt{qjackctl}

>2 channels possible

\textbf{Connections - JACK Audio Connection Kit}

- **Readable Clients / Output Ports**
  - \texttt{low-pass}
    - from\_slave\_1
    - from\_slave\_2
    - from\_slave\_3
    - from\_slave\_4
  - \texttt{system}
    - capture\_1
    - capture\_2

- **Writable Clients / Input Ports**
  - \texttt{low-pass}
    - to\_slave\_1
    - to\_slave\_2
    - to\_slave\_3
  - \texttt{system}
    - playback\_1
    - playback\_2
Module Identification

• Either single SD card with all possible modules
  – Need module selection by software / hardware switch
    → comfortable, but not out-of-the-box

• Or dedicated SD cards with single module
  – Out-of-the-box, but need many SD cards

• Or identification via additional shield
  – Require additional minimal hardware, but can be used
to provide dedicated input knobs / sliders
Conclusions

- Linux on RPi *almost* ready for out-of-the-box modular synthesizer
- Severest bottleneck: all audio routing via central control host → require remote routes
- Missing remote version of *qjackctl* for setting up remote routes
- Missing JACK client wrapper for e.g. *LV²* plug-ins
- Need to speed up boot time
- Medium-scale number of open bugs (crashes, hangs, …)
Questions?