Grabbing audio and video on a farm

Krzysztof Opasiak
Samsung R&D Institute Poland
Agenda

- What is board farm for?
- Typical setups
- HDMIPi as PoC
- Future work
- Summary
- Q & A
What is board farm for?
Automated testing systems

- Manual testing is very time consuming
- It does not scale
- Fortunately most of tests can be automated
- There are two options to run them
  - Emulator
  - Physical board
• Some development targets have to be kept secret
• Having as many boards as developers is often not feasible
• This makes the board a shared resource
• But doing this in traditional way is at least annoying
Remote access to HW

• Samsung (like many others) is scattered around the world
• Not every board is available in every office (or home)
• Sometimes issues are very hardware dependent
• Buying or shipping the board may often take days if not weeks
Board Farm challenges

• Every board is unique like a snow flake
• It’s hard to provide unified access to it (but possible)
• Maintenance
• Stability
• Scalability
Typical setups
LAVA

- Linaro Automated Validation Architecture
- yaml job description
- Board interfaces
  - Power relay
  - Serial
  - USB/Ethernet
- Typical test consists of
  - Downloading image
  - Flashing it to the DUT
  - Booting
  - Running test script
  - Collecting results

Source: [10]
Lab in a box

- PC case + PDU
- PC [(M), (D)]
- ACME
- Serials
- USB hub/controller
- Ethernet switch

Source: [9]
Test manager

Farm manager

Board controller

WELES

BORUTA

MuxPi

https://github.com/SamsungSLAV
MuxPi

• Power Control
• SD-Mux
• DyPer
• Serial
• USB/Serial switch
• Power measurement
• Display
• 2 RGB diodes
• Extendability
• EDID injector

Source: [6]
• In order to identify the display EDID is required
• Without it you cannot make a screenshot
• Fortunately it can be injected via I2C
HDMIPi as PoC
Why not dedicated HW?

- There are HDMI grabbers available on the market
- But they cost around $300
- And we need tens of those devices not just one
- And still you need to somehow stream the video over the network
- Most of them require USB 3.0
- Many of them cause high CPU usage
Lenkeng LKV373a HDMI Extender

- HDMI to Ethernet RX/TX
- HDMI Video and Audio converted to MPEG stream
- Streaming using multicast UDP
- A lot of investigation on this by danman[1]
- Cost?
Lenkeng LKV373a HDMI Extender

- HDMI to Ethernet RX/TX
- HDMI Video and Audio converted to MPEG stream
- Streaming using multicast UDP
- A lot of investigation on this by danman[1]
- Cost?

$30
Wiring
• LKV streams MPEG stream using UDP multicast
• Default IP: 192.168.1.238
• Default multicast group: 239.255.42.42:5004
• Stream can be captured using VLC, ffmpeg and many others
• VLC params really matter
• VLC version matters even more
• NanoPi kernel version also matters
Audio and Video - solution

- LKV is connected to USB network card in MuxPi
- NanoPi has to reroute the stream and at least change the SRC
- Solution #1:

  ```
  # socat UDP-RECV:5004,ip-add-membership=239.255.42.42:192.168.1.1 \
  UDP-SENDTO:192.168.0.1:5004
  ```

Issues: High CPU load + high delay
Issue: Unable to change destination (anyone?)
Audio and Video - solution

- LKV is connected to USB network card in MuxPi
- NanoPi has to reroute the stream and at least change the SRC
- Solution #1:

  ```
  # socat UDP-RECV:5004,ip-add-membership=239.255.42.42:192.168.1.1 \
  UDP-SENDTO:192.168.0.1:5004
  ```

Issues: High CPU load + high delay
Audio and Video - solution

• LKV is connected to USB network card in MuxPi
• NanoPi has to reroute the stream and at least change the SRC
• Solution #1:

```bash
# socat UDP-RECV:5004,ip-add-membership=239.255.42.42:192.168.1.1 \ 
   UDP-SENDTO:192.168.0.1:5004
```

Issues: High CPU load + high delay

• Solution #2:

```bash
# smcroute -d
# smcroute -a eth1 192.168.1.238 239.255.42.42 eth0
# iptables -t nat -A POSTROUTING -p udp -d 239.255.42.42 \ 
   --to-source 192.168.0.13
# conntrack -F
```
Audio and Video - solution

- LKV is connected to USB network card in MuxPi
- NanoPi has to reroute the stream and at least change the SRC
- Solution #1:

  ```
  # socat UDP-RECV:5004,ip-add-membership=239.255.42.42:192.168.1.1 \
  UDP-SENDTO:192.168.0.1:5004
  ```

  **Issues:** High CPU load + high delay

- Solution #2:

  ```
  # smcroute -d
  # smcroute -a eth1 192.168.1.238 239.255.42.42 eth0
  # iptables -t nat -A POSTROUTING -p udp -d 239.255.42.42 \ 
  -j SNAT --to-source 192.168.0.13
  # conntrack -F
  ```

  **Issue:** Unable to change destination (anyone?)
Audio and Video - versions

- VLC version used: v3.0.3
- Params:
  - Network caching: 200
  - Clock jitter: 0
- NanoPi kernel: 4.11.2-github.com-friendlyarm-linux-43baabc1f+
- smcroute: 2.0.0 build 150506
HID - Human Interface Device

- NanoPi is equipped with UDC
- So it can act as any USB device[8]
- Linux Kernel provides HID protocol implementation
- So we can easily act as a Keyboard and Mouse device

```bash
cd /sys/kernel/config/usb_gadget
mkdir g1

cd g1

echo 0x1a2c > idVendor
echo 0x0c21 > idProduct

mkdir strings/0x409
echo "MuxPi" > strings/0x409/manufacturer
echo "Kbd+Touch" > strings/0x409/product
echo "001" > strings/0x409/serialnumber

mkdir functions/hid.kbd

cd functions/hid.kbd

echo 0 > subclass
echo 1 > protocol
echo 8 > report_length

cat /root/usb_hid/kbd_desc > report_desc
cd -

mkdir functions/hid.touch

cd functions/hid.touch

echo 0 > subclass
echo 0 > protocol
echo 8 > report_length

cat /root/usb_hid/touch_desc > report_desc
cd -

mkdir configs/c.1

ln -s functions/hid.kbd configs/c.1
ln -s functions/hid.touch configs/c.1

echo `ls -1 /sys/class/udc` > UDC
```
Keyboard

- Write report descriptors to `/dev/hidg0`
- Report descriptor format:

```
01000000 0x00 0x38 0x00 0x00 0x00 0x00 0x00
```
Mouse

• Write report descriptors to /dev/hidg1
• Not really a mouse but a touchscreen
• Some math to calculate the position
• Report descriptor format:

```
Number of fingers

0x01 0x42 0x03 0x32 0x00 0x32

Finger ID State

X Y
```
Putting all into one app

- App written using Qt library
- QSsh for commands execution
- QSsh for HID fwd
- QVlc for display

https://github.com/kopasiak/muxpi
Issues

- Lack of mouse tracking
- In-kernel multicast to unicast conversion
- Delay
- Video quality
- Stability
Future work
LKV373a Issues

- HDMI extender works (usually)
- Hard to tweak stream parameters
- Sound is getting out of sync
- Unable to set compression params
- Stability issues
- The Black Box
TimVideos - HDMI2USB

- OSS project for recording conferences
- Used for recording various events in Australia
- HDMI2USB[2]
  - Open Source
  - Open Hardware (Numato Opsis[7])
  - Multi input
  - Multi output
  - HDMI and DP grabber

- Cost:
TimVideos - HDMI2USB

- OSS project for recording conferences
- Used for recording various events in Australia
- HDMI2USB[2]
  - Open Source
  - Open Hardware (Numato Opsis[7])
  - Multi input
  - Multi output
  - HDMI and DP grabber

- Cost: $430
BTW - MiGen + LiteX

- Allows to program FPGA in Python
- Makes the code modular
- Makes the code really portable
- Makes the code really Open Source!
- Permissive two-clause BSD license
- LiteX[4] Cores Ecosystem
TODO list

• Get access to Numato board or any other dev board
• Reduce the FPGA bitstream:
  • Only single HDMI input
  • Only Ethernet output
• Find cheaper FPGA to handle this
• Redesign the board
Summary
Summary

• Creating a board farm is challenging
• Multi DUT vs Single DUT approach
• HDMI Extender can be useful for grabbing audio and video
• But it has some issues…
• Some collaboration with TimVideos could make a great result
• It’s easy to simulate keyboard and mouse interaction
Q & A
Thank you!

Krzysztof Opasiak
Samsung R&D Institute Poland

+48 605 125 174
k.opasiak@samsung.com


