Lava Multipurpose Probes (Lava-Lmp)

Specification

Andy Green <andy.green@linaro.org>

0.1	5 Dec 2012	Initial version
0.2	7 Dec 2012	• SD Mux • mounting considerations • clarify multi connector set boards only have one connector set fitted on a given board and set the SCEN bus accordingly

Introduction

This document describes a cheap, flexible hardware probe architecture for extending Lava tests via a “test buddy” to cover tests that are currently impossible for Lava.

Problem to be solved

Vendor boards that need Lava tests have many interfaces that cannot be exercised currently. It seriously restricts test coverage once the tests that can be done “from inside” the SoC have been implemented.

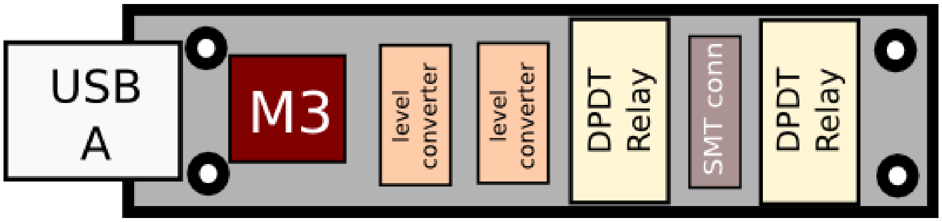
The kinds of test that are out of scope for lava at the moment include

Selecting boot source, eg, from USB, Sata, eMMC, SD
Ethernet Hotplug correctness
HDMI Hotplug correctness
USB Device Hotplug correctness
USB Host Hotplug correctness
USB OTG mode switching correctness
USB Host / OTG Host power provision correctness
SATA / eSATA Hotplug correctness
Audio socket hotplug correctness
LED operation
Switch operation
SoC Package accurate temperature verification
Scanned keypad operation simulation
Ambient light sensor
External SoC package heating
Proximity sensor
Capacitive touchscreen operation emulation
SD Card multiplexing for boot test

Lava-Lmp Solution

A small USB Cortex-M3 based “USB stick” device is defined with a selection of hardware interfaces needed to service all of the requirements above.

The rationale for this is USB will be compatible with all test hosts; it is extensible by hubs; the hubs can be placed physically near the DUT; power is provided to the devices in a clean way; you can have as many as you need next to any DUT.



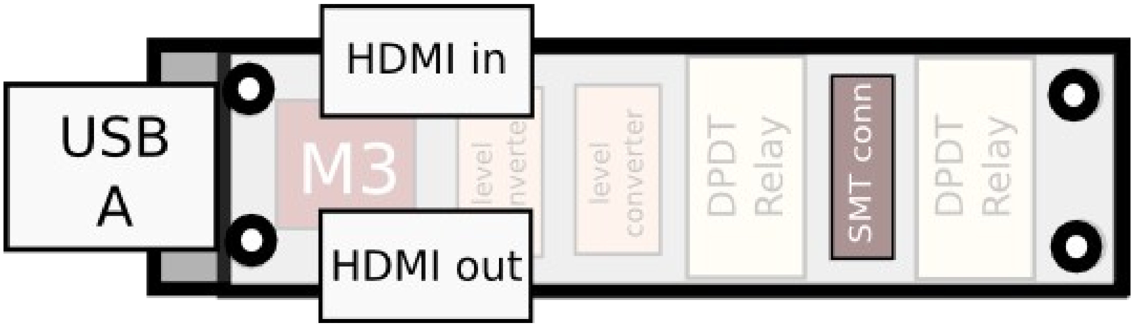
The selected IO appears at a single SMT connector.

The selection of IO provided is the following

<table>
<tr><th>IO</th><th>Capability</th><th colspan="2">Provides testing for</th></tr>
<tr><td>2 x 2-channel relay switching
(ATE relay 50M operations lifetime minimum)</td><td>Allows all kinds of “interruption” disconnect / hotplug testing without having to take care about protocol compatibility for muxing or levels</td><td colspan="2">• HDMI hotplug / EDID source select
• USB host / device / otg hotplug
• Audio hotplug
• Ethernet hotplug
• SATA / E-Sata hotplug</td></tr>
<tr><td>8 x level-converted output, individually tristateable</td><td>Drive 8 x logical signal at any voltage reference</td><td>• Boot source selection
• Switch actuation simulation
• Ambient light sensor (open collector)</td><td rowspan="2">• Scanned keypad testing up to 8x8
• I2C device emulation
• HDMI EDID emulation
• I2C temperature sensor</td></tr>
<tr><td>8 x level-converted input</td><td>Sense 8 x logical signal at any voltage reference</td><td>• LED test
• Confirm state of any static signal</td></tr>
</table>

In addition to this “base” board additional “scenario” boards are defined.

These mount on top of the base board and provide appropriate connectors for the test activity. Here is an example scenario board for HDMI test (covering HDMI hotplug control, and ability to either pass through EDID or provide a synthetic EDID from the test host instead)



The scenario boards typically do not need additional components, just appropriate connectors hooked up to the provided signals from the base board.

Scenario Board identfication

A 3-bit bus is provided on the SMT Connector to identify which scenario the board is configured for. These may be tied high, low or left floating for a total of 26 possible scenario boards + all-floating reserved to detect no scenario board present.

Software interface to PC

The base board always presents itself as a ttyACM serial device and uses human-readable control and status protocol, dependent on the scenario board fitted. It has a unique USB serial number.

Base board firmware

There is only one base board firmware which supports all scenarios. The correct scenario software support in the Cortex M3 firmware is selected when it boots, from querying the scenario board ID it is fitted with. No state regarding scenario selection is required to be sent by the PC or appears at the USB configuration level.

Base Board IO Connector signals

A 40-pin SMT connector provides the following signals. Note Relay 1 + 2 are controlled by one signal from the Cortex M3 and Relay 3 + 4 are controlled by a second signal independently.

Row	A	B
1	Relay 1 NO	Relay 1 COM
2	Relay 1 NC	0V
3	Relay 2 NO	Relay 2 COM
4	Relay 2 NC	0V
5	OUT0	OUT1
6	OUT2	OUT3
7	OUT4	OUT5
8	OUT6	OUT7
9	+5V	Vref for ~~In and Out~~ Out signals
10	IN0	IN1
11	IN2	IN3
12	IN4	IN5
13	IN6	IN7
14	0V	SCEN1
15	ADC IN 1	ADC IN 2

16	0V	SCEN2
17	Relay 3 NO	Relay 3 COM
18	Relay 3 NC	SCEN3
19	Relay 4 NO	Relay 4 COM
20	Relay 4 NC	Vref for In signals~~0V~~

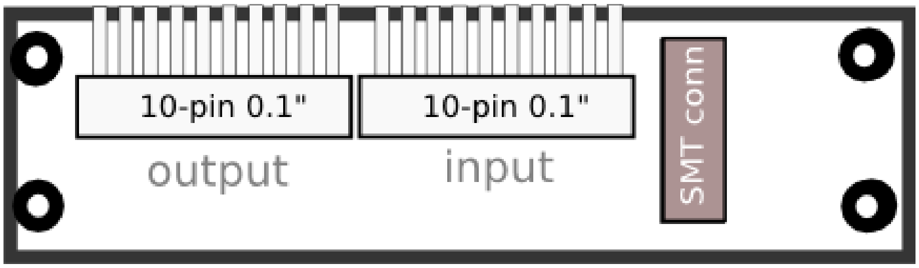
Mounting considerations

Physical mounting in a safe and repeatable way around the DUT is expected to be an issue. Threaded right-angle mounts will be provided at both the “thin ends” to allow mounting the cards in either a cage or single stiff drilled mounting backplane at either or both ends.

Scenario boards

Scenario: Digital IO

This provides 8 level-converted 3-state outputs suitable for controlling pulled-up or pulled-down wired boot control signals, and 8 level-converted inputs suitable for checking the state of signals



Pin	Output connector signal	Input connector signal
1	Vref (wire to board supply to be used for a high level output, eg, 1.8V)	Vref (wire to board supply the input signals are referenced to, eg, 1.8V)
2	OUT0	IN0
3	OUT1	IN1
4	OUT2	IN2
5	OUT3	IN3
6	OUT4	IN4
7	OUT5	IN5
8	OUT6	IN6
9	OUT7	IN7

10	0V	0V

Scenarios useful for

- Boot source selection
- Switch actuation simulation
- LED state confirmation
- Scanned keypress simulation (practically will need custom scenario board with matching flat cable socket)

Host Software interface

- WXXXX10XX
  - W = write new levels
  - 8 characters defining the 8 IO levels, 0 = 0V, 1=Vref, X=floating
- R

  00001001

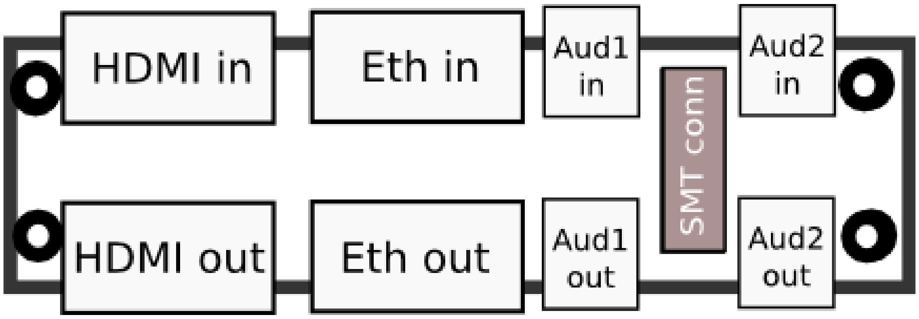
  - *R = return input states*

Scenario: Common Hotplug

This board provides hotplug test support for the following interfaces:

- Ethernet (no explicit plug detect signal, RX_/- and TX+/- must be disconnected)
- HDMI (Edid and hotplug signal controllable)
- 2 x 3.5mm Stereo audio (explicit and implicit plug detect: independently controlled)

Only one kind of interface can be used at a time on a single board. If you need to control them all three base + scenario boards will be needed. Only one connector set is fitted to a given board and the SCEN[1..3] bus set accordingly.



For implicit (impedence-based) audio hotplug sensing, just pass the audio connectors through the jacks and both stereo audio signals will be interrupted under software control. If explicit plug detect (switch in the jack) is used, interrupt the plug detect signal on the board and pass it through one of the audio jacks on the scenario board so it can be interrupted under software control.

Scenarios useful for

- HDMI hotplug
- HDMI EDID emulation
- Ethernet hotplug test
- Audio jack presence detect x 2

Host Software interface

- W00
  - W = write new relay levels
  - For each relay pair, 0 = off / “normal”, 1 = actuated
- EDID=<hex bytes until CR>
  - Set the emulated EDID for use with HDM when the second relay pair is actuated

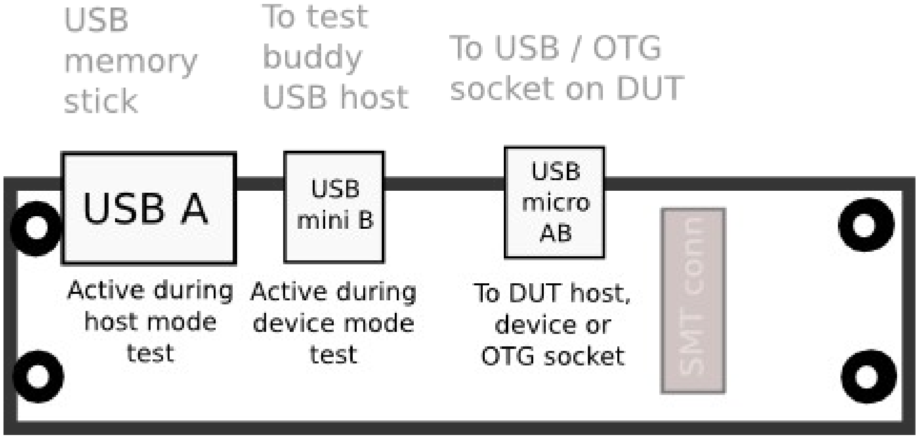
Scenario: USB2 hotplug

This provides test support for all kinds of USB-related hotplug. When plugged into a DUT USB OTG socket, it can simulate

- Disconnect
- Connection to USB device plugged into the A socket, eg, USB memory stick
- Connection to USB host plugged into the mini B socket, eg, test host buddy

It can also provide the correct SENSE pin level for the OTG scenario being tested, simulating the correct cable plugged in.

Additionally it monitors the voltage provided by the DUT in the host mode.


USB memory stick
To test buddy USB host
To USB / OTG socket on DUT
USB A
USB mini B
USB micro AB
SMT conn
Active during host mode test
Active during device mode test
To DUT host, device or OTG socket

Scenarios useful for

- USB Host hotplug and functionality confirm
- USB Host voltage monitoring
- USB Device hotplug
- USB OTG mode sensing by SENSE pin
- USB OTG role switching

Host software interface

- M_
  - ○ _ is H or D for Host or Device, or X for disconnected
- V

  5.13
  - ○ Report minimum voltage seen since last V

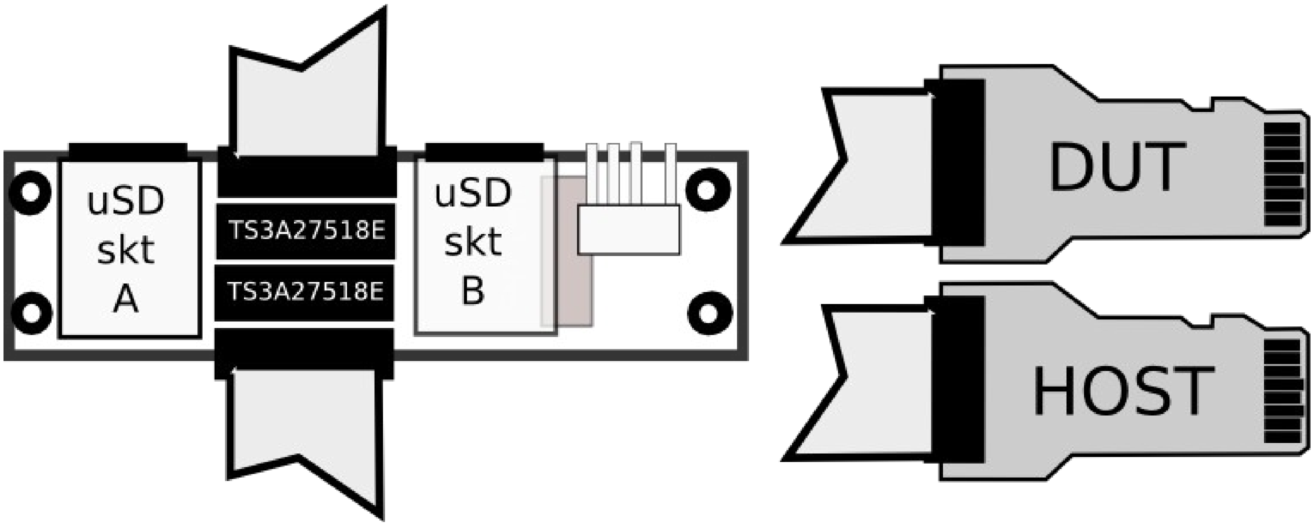
Scenario: SD Card mux

Description

This allows the host and Cortex M3 chip to control which of two Micro SD cards, A and B, are seen by the DUT at boot time or optionally the host at any time.

One or two 10-pin flex cable connect the scenario board to a microSD PCB probes. These uSD footprint probes can be used with a uSD to fullsize SD adapter as needed.

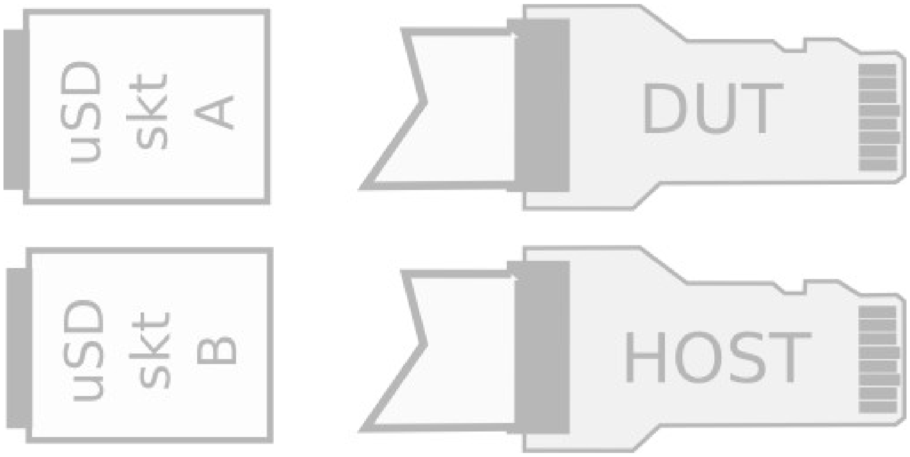
The main board relays are used to separately disconnect DUT power, and disconnect DUT SD Card presence detect if appropriately wired.



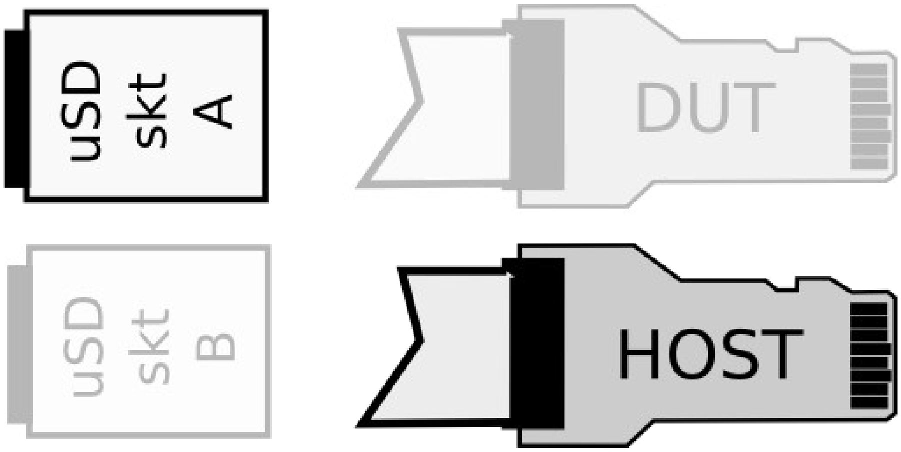
Mux mappings

There are five mappings this scenario card can perform:

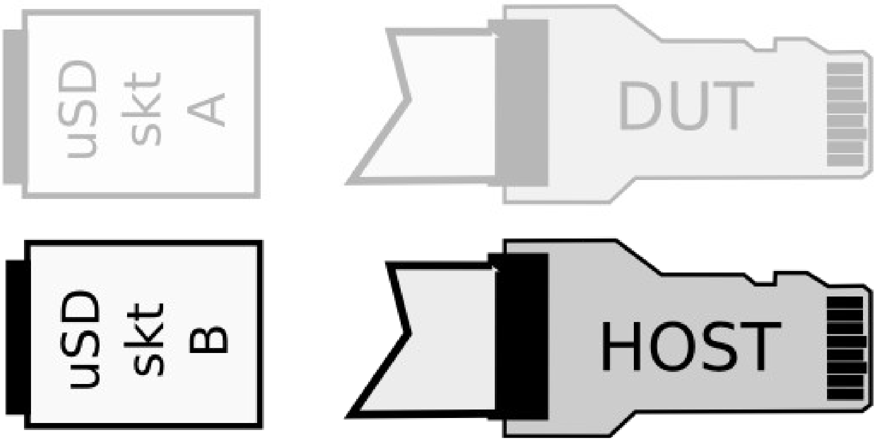
1) Disconnected. DUT SD Card presence detect relay is opened. This is useful during DUT SD Card hotplug test action.



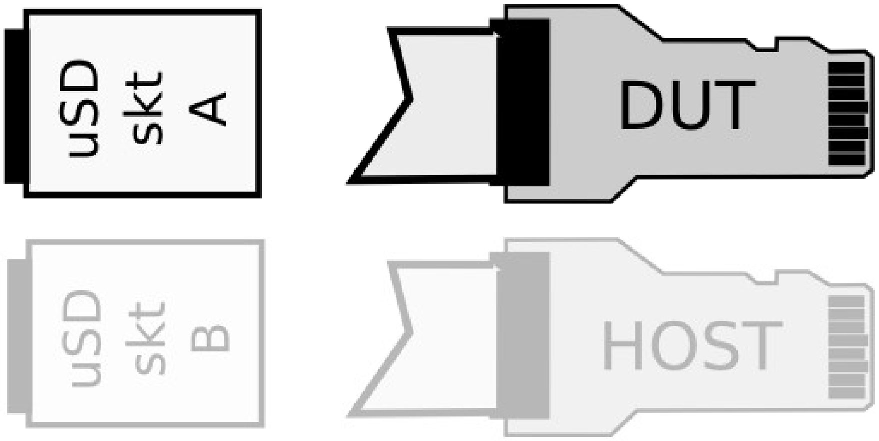
2) Host access to SD Card A; DUT disconnected. DUT SD Card presence detect relay is opened.



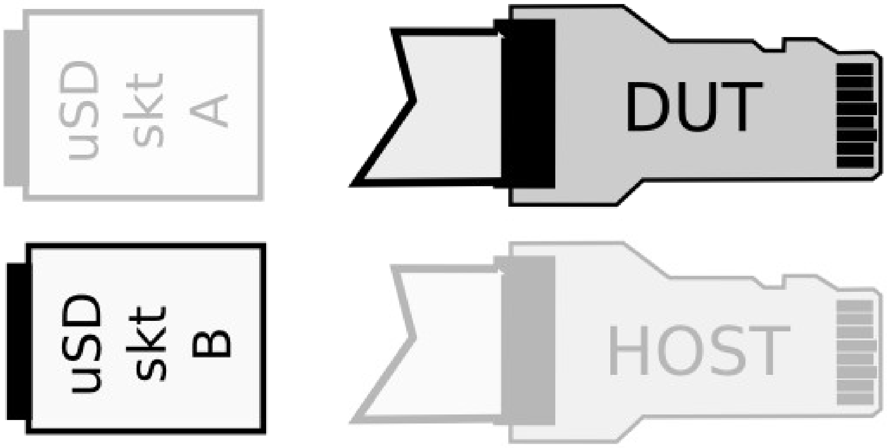
3) Host access to SD Card B; DUT disconnected. DUT SD Card presence detect relay is opened.



4) DUT access to SD Card A; Host disconnected. DUT SD Card presence detect relay is connected.



5) DUT access to SD Card B: Host disconnected.   DUT SD Card presence detect relay is connected.



Disconnect control header

1	DUT main power disconnect A	Host can request disconnection to force DUT power cycle
2	DUT main power disconnect B	
3	DUT SD Card Presence Detect disconnect A	These are connected while either uSD card is visible at the DUT, and disconnected otherwise
4	DUT SD Card Presence Detect disconnect A	

The idea is that by inserting these in series the DUT power and SD Card detect switch path, the host / mux device can control DUT power and card detect state appropriately.

Dynamic mux switching

For DUT performing uSD boot, this allows Lava to continue normally from a known-good boot / kernel / rootfs on uSD A no matter what the state of the test rootfs on uSD B.

In order to optionally allow access to uSD B even after booting from uSDA, the microcontroller can watch for periods when the uSD is not in use and after a programmable count and duration automatically swap to uSD B. This should be set to coincide with the end of bootloader uSD accesses and before Linux probes the card.

This leaves the DUT booted with a known-good environment and initrd, but with uSD active, and allows operation without a host connected.

Scenarios useful for

- USD card A can contain a known-good boot, kernel and initramfs. At the start of test preparation, the DUT is always booted into uSD A environment
- If a SD card <-> USB bridge is also provided, then the host can have write access to both uSD cards as well

Host software interface

- M<n>
    - Select mux mode 1..5
- P<0 | 1>
    - 0 = disconnect DUT power
    - 1 = connect DUT power
- I<ms>
    - Set number of ms with no bus activity before considered idle
- C<n>
    - Count of idle events before we automatically swap back to uSD B (0 = stay on uSD A)
- D<0 | 1>
    - Enable / disable dynamic DUT card switching once after next DUT power connection, using idle rules above