Linux Automation GmbH

LXA USB-Mux

Typical Applications

The **USB-Mux** switches the USB 2.0 connection of an Embedded Linux Device Under Test (DUT) between another device (e.g. a USB storage device) and the USB host computer. The USB-Mux is controlled from the host using an Open Source software tool.

Typical Applications:

- Embedded Linux operating system development
- Continuous testing of software on embedded hardware
- Remote development enablement

Typical Use Cases

- **Booting Embedded CPUs using ROM code**
  Modern embedded CPUs feature a “boot from ROM code” mode which often allows to boot a system from USB.
  The USB-Mux can be used to switch a DUT’s USB port to the host computer on demand.
- **Testing of updates from USB devices**
  The USB-Mux allows to prepare an USB storage device with an image from a host computer and afterwards switching it’s connection to the DUT.
- **Plug / Unplug Testing for the DUT’s USB-Stack**
  The USB-Mux allows to remotely plug and unplug an USB device from the DUT.
- **Testing power fault handling of USB devices**
  The USB-Mux allows to remotely cut the power to an USB device allowing to testing it’s ability to handle power faults.
- **Testing and Quality Assurance**
  The USB-Mux is a building block for testing and quality assurance of Embedded Linux devices if these need to be booted from USB.
- **Remote Development and Resource Sharing**
  With remote control over the DUT there is no need to have it on your desk. Remote development and sharing of scarce prototypes is possible.

USB-Mux (Host side)

USB-Mux (DUT side)

Interfaces

- **Host Port**
  Connects the USB-Mux to the test server.
  (USB 2.0, Type B socket)
- **Device Port**
  Connects the USB-Mux to an USB device that can be either switched to the test server or the DUT.
  (USB 2.0, Type A socket)
- **DUT Port**
  Connects the USB-Mux to the DUT. This connection can be either switched to the test server or the device port.
  This port features an ID pin that can be controlled from the test server.
  (USB 2.0, 2.54 mm / 0.1” *DuPont* compatible connector)

Additional Features

- **Internal USB Hub**
  Thanks to the internal USB hub all USB devices appear on different ports facing the test server. This makes it easy to differentiate between, for example, DUT and device using the USB path.
- **DUT Power lock**
  This hardware switch allows to disable the connection between the host and DUT port. This helps preventing accidental connection of two host controllers to the same USB bus.
- **Unique Serial Numbers**
  Every USB-Mux is identified by a unique serial number. This simplifies operation of multiple devices on the same test server.

Open Source Control Software for Linux

- Python-based control software
- Available from GitHub and pypi
- Controls the connections made inside the USB-Mux
- Controlled via the command line
- Can be used as a library

- Already integrated into labgrid

LAB GRID

One of our focuses for the design of our software and hardware is the integration of labgrid.

Linux Automation GmbH, Steuerwalder Straße 21, 31137 Hildesheim, Germany
+49 51 21 / 20 69 17 140 shop.linux-automation.com
info@linux-automation.com linux-automation.com

umx-M02-R01, 2022-12-08

Technical Data

USB Standard	USB 2.0
USB Hub	Microchip USB2514
Microcontroller	STM32F04
Input voltage	4.5 .. 5.5 V from host port
Current draw	< 100 mA from host port
Power Supply Switch On-Resistance	90 mΩ typ. < 200 mΩ worst case (for each possible path)
Power Supply Switch Max. Current	1.5 A continuous (for each possible path)
Data Switch Bandwidth	1.22 GHz (-3 dB)
Data Switch On-Resistance	< 10 Ω
ID-Pin On-Resistance	<= 100 mΩ worst case
ID-Pin Max. Current	0.5 A continuous
DUT Connector Pinout	• + 5V (input or output) • USB D- (input or output) • USB D+ (input or output) • ID-Pin (open drain) • GND (common to all connections) • Shield (connected to GND)
DUT Connector	Amphenol 78208-106HLF

System Requirements

- Linux host with Python
- USB 2.0 host port

Customization Services

In case the USB-Mux does not fully fit your needs we provide customized hardware and software solutions based on our existing ecosystem.

Integration and Test Development Services

With our partner Pengutronix we provide comprehensive services: We can help with integration of our USB-Mux into your existing remote-control or Labgrid environment and can get your Embedded Linux testing activities up to speed.

Optional Accessories

- Micro-USB DUT connection cable
- USB-C DUT connection cable
- USB-A DUT connection cable

Further Links

- **Handbook**

https://linux-automation.com/umx-M01

- **Control Software**

https://github.com/linux-automation/usbmuxctl

- **Product Page**

https://www.linux-automation.com/de/products/usb-mux.html

Specification is based on the most recent prototype and can change as development progresses. This datasheet is subject to change without notice.