mipi®
DEVCON
VIRTUAL EVENT

MIPI ALLIANCE
DEVELOPERS
CONFERENCE
22-23
SEPTEMBER
2020

The Story Behind the MIPI I3C HCI℠ Driver for Linux

Nicolas Pitre
BayLibre

MOBILE & BEYOND

MIPI.ORG/DEVCON

The Story Behind the MIPI I3C HCI℠ Driver for Linux

What is Linux?

What does "Linux" refer to?

- A complete system (distribution)
- An Operating System (OS)
- A kernel

The Story Behind the MIPI I3C HCI℠ Driver for Linux

What is Linux?

Linux development is:

- Organically grown, not "designed"
- Driven by code contributors
- Not dictated by anyone in particular
- Peer reviewed in the open

The Story Behind the MIPI I3C HCI℠ Driver for Linux

Why Linux?

***Linux* is a compelling choice**

- Large developer community
- Best looked-after network stack
- Extensive storage options
- Scalable from embedded IoT device to supercomputers
- All-around very popular and widely used

Providing MIPI I3C HCI support for Linux is a good strategy

The Story Behind the MIPI I3C HCI℠ Driver for Linux

What is Linux?

***Linux* is an untraditional choice:**

- Freely obtainable
- Freely modifiable

BUT

- Modifications to GPL-licensed source code must be made available

People contribute their changes back to main Linux:

- To simplify their GPL obligations
- To lower their maintenance cost
- To benefit from Linux's distribution channels

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The MIPI I3C Host Controller Interface

Standardized interface between software and hardware allows for:

- Common software driver
- Reduced software effort
- Sharing the cost of non-differentiating development

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The MIPI I3C Host Controller Interface

The I3C HCI specification is:

- Comprehensive
- Extensive
- Versioned

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The MIPI I3C Host Controller Interface

Comparison between existing drivers in their current state

Table 1. Lines of code for various I3C Host Controller Drivers

Driver	# of lines
Synopsys DesignWare I3C	1219
Silvaco I3C master	1349
Cadence I3C master	1689
MIPI I3C HCI	4093

The MIPI I3C HCI design covers a lot more ground

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The MIPI I3C Host Controller Interface

Yet, support is still incomplete.

Some of the unsupported MIPI I3C HCI features:

- Auto-commands
- Scheduled commands
- Master role handoff

Important changes to the Linux I3C subsystem is required before those features can be accommodated

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

There are 3 layers:

- Host Controller (hardware) drivers
- The I3C subsystem (common) layer
- Device (or function) drivers

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

Host Controller (hardware) drivers:

- Discovers and initializes controller hardware
- Implements hardware abstraction for the I3C subsystem
- Performs low-level requests from the I3C subsystem
- Signals hardware events to the I3C subsystem

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

The I3C subsystem (common) layer:

- Manages registration of host controller and function drivers
- Collects info about discovered device and configures them
- Connects device instances with appropriate function driver
- Provides a library of helpers for common driver tasks

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

The device (or function) driver:

- Concerns itself with functionality of a single device (camera, storage, etc.)
- Is host controller agnostic
- Gets involved only when matched to an actual device instance by the I3C subsystem

Only driver currently in mainline Linux:

- STMicroelectronics LSM6DSx 6-axis IMU sensors

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

Function driver interface currently provided by the I3C subsystem:

- Register/Unregister for a device
- Perform private read and/or write data transfers
- Register for receiving IBI's

The Story Behind the MIPI I3C HCI℠ Driver for Linux

The Linux I3C subsystem structure

Needed extensions to the Linux I3C subsystem:

- IBI interface extension for auto-commands
- Interface for scheduled commands
- Allow for time stamped IBI's
- Device private CCC's
- Bus mastership handoff handling

Involvement from interested parties is required

The Story Behind the MIPI I3C HCI℠ Driver for Linux

Linux development model peculiarities

Common wisdom with regards to Linux development:

- There are no stable API's in the kernel
- A new API must have at least one in-tree user
- Never over-engineer an API, because…
- There are no stable API's in the kernel

Those are more reasons why people contribute their code to the main Linux source code

The Story Behind the MIPI I3C HCI℠ Driver for Linux

Conclusion

Considerations for future work:

- The MIPI I3C HCI driver's core structure is done
- Further HCI support require I3C subsystem extensions
- New I3C subsystem extensions require example usage from function drivers
- More function drivers are sorely needed

The Story Behind the MIPI I3C HCI℠ Driver for Linux

Questions?

Nicolas Pitre
npitre@baylibre.com
www.baylibre.com

Baylibre


mipi®
DEVCON
VIRTUAL EVENT
THANK YOU
MOBILE & BEYOND
MIPI ALLIANCE
DEVELOPERS
CONFERENCE
22-23
SEPTEMBER
2020
MIPI.ORG/DEVCON