

A network diagram background consisting of a teal-to-green gradient. It features a complex web of white lines connecting various colored nodes (white, orange, red, purple, blue). The background is also filled with a pattern of small, faint icons related to mobile technology, such as smartphones, Wi-Fi symbols, and speech bubbles.

Bootstrapping MIPI CCSSM Software Support with CCS Tools

Sakari Ailus

Intel Corporation &
Member of MIPI Camera
Working Group

Presentation Agenda

About MIPI Alliance

Peter Lefkin

MIPI Alliance Managing Director

Presentation of MIPI CCS Tools

Sakari Ailus

*Software Engineer, Intel Corporation
and Member of MIPI Camera Working Group*

Demonstration of MIPI CCS Tools and Linux CCS Driver

Questions and Answers





Introduction to MIPI Alliance

Peter Lefkin

Managing Director, MIPI Alliance

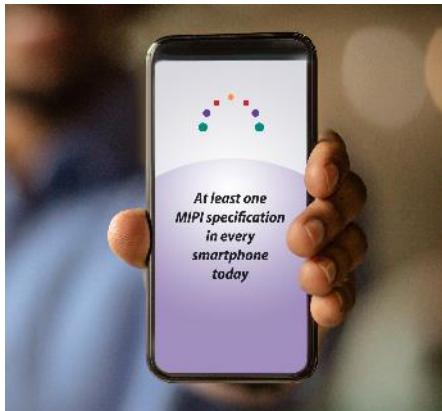
About MIPI Alliance



2003 THE CELL PHONE MARKET



IN 2003 MIPI ALLIANCE WAS FORMED TO STANDARDIZE CAMERA AND DISPLAY INTERFACES



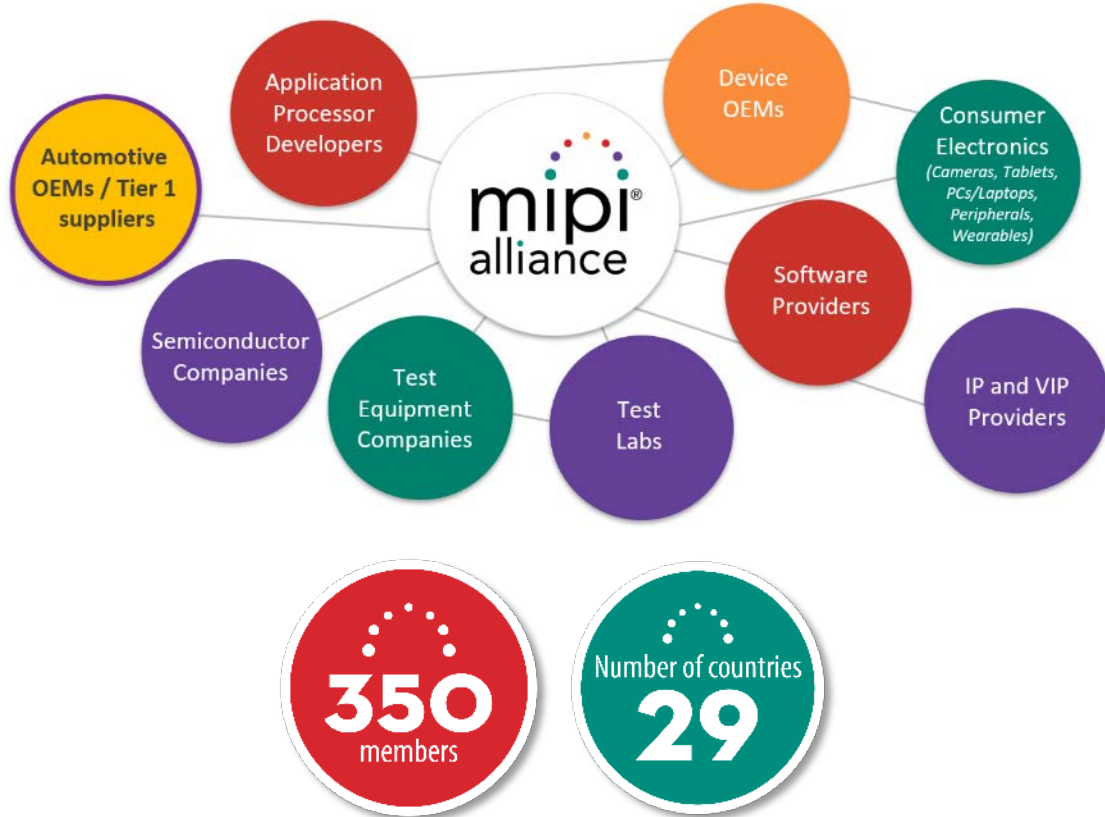
At least one MIPI specification in every smartphone today

2020

MIPI ALLIANCE HAS DEVELOPED ROUGHLY 50 SPECIFICATIONS COVERING THE FULL RANGE OF INTERFACE APPLICATIONS NEEDED FOR MOBILE DEVICES



TODAY'S MIPI MEMBER ECOSYSTEM



Board and Contributor Members

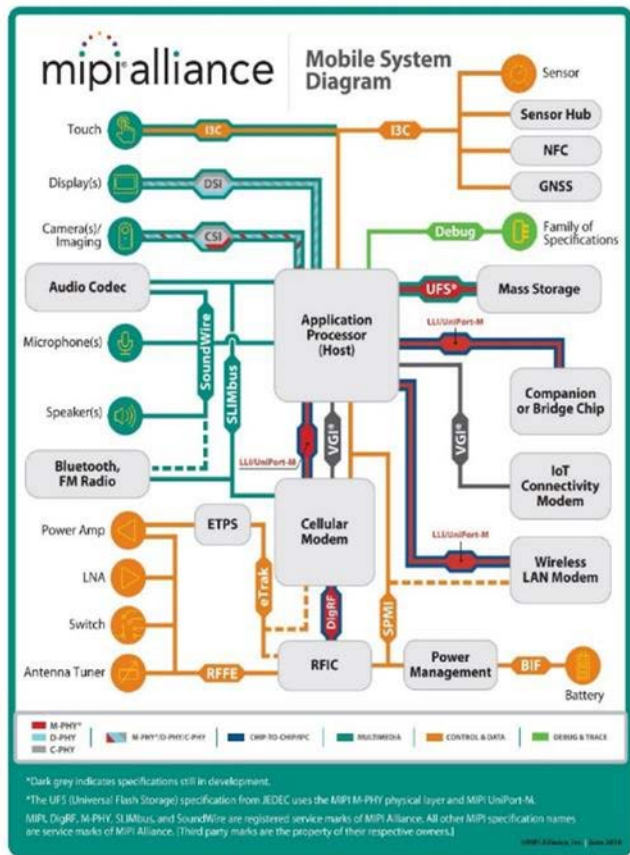
Contributor Members



Board Members



MIPI Specifications Leveraged Beyond Mobile



50
Number of
current
specifications

Fundamentally, usage rights are granted to members royalty free for implementation of MIPI specifications from all MIPI members



Bootstrapping MIPI CCSSM Software Support with CCS Tools

Sakari Ailus
Intel Corporation &
Member of MIPI Camera
Working Group

MIPI Camera Command Set (CCS)



- MIPI CCS standardizes the control interface of raw camera sensors
- Capability and limit enumeration
 - Low-end to high-end
- Extensible



Generic driver



- A common control interface enables writing a common driver for all compliant sensors:
 - No driver changes needed to support new camera sensors
 - No need for per-sensor driver maintenance effort

Limit and capability information



- Limit and capability information has traditionally resided in hardware registers, but:
 - Some values may be known only at a late phase of sensor development
 - Some values may need updating after the hardware has been manufactured
 - Register based interface is not a good fit for describing some aspects of the sensors
- And what to do with Manufacturer Specific Registers (MSRs)?

MIPI CCS v1.1: CCS static data



- CCS static data addresses these problems by moving the limit and capability information out of the sensor's registers
- The block-based format includes support for:
 - Read-only registers (capabilities and limits)
 - Manufacturer Specific Registers (MSRs)
 - Frame format descriptor
 - Conditional functionality
- Extensible

But...



```
0000000 0802 0300 0000 e207 1404 1a03 1117 4000
0000010 7a4a 8000 3248 0200 1081 0000 3a01 0800
0000020 0a00 0800 0a00 2105 3381 4255 8a13 5044
0000030 3412 7856 2353 3445 4553 0d88 aae e cc0a
0000040 42bb dd9a 80ff ffff 2003 051a 0901 1101
0000050 08ff 8302 4011 f000 7f3f 040a 4481 5650
0000060 4278 aa12 7fcc a204 e320 00e9
```

MIPI CCS Tools



- MIPI CCS Tools contain software to work with CCS static data binary format defined in MIPI CCS 1.1
 - YAML based, human-editable format of CCS static data
 - CCS static data YAML to binary format converter
 - CCS static data parser library
- Licensed under the 3-clause BSD license



MIPI CCS Tools: Static data YAML format



- Human editable
- Can be easily generated by other programs
- Functionally equivalent to CCS static data binary format

MIPI CCS Tools: Static data YAML format (example)

```
---
data-version:
  version-major: 3
  version-minor: 0
  date-year: 2018
  date-month: 04
  date-day: 20

sensor-read-only-regs:
  - mipi_ccs_version 0x11
  - data_pedestal 0x0040
  - analogue_gain_code_min 0x0080
  - analogue_gain_code_max 0x4832
  - analogue_gain_code_step 0x0002
...
```



MIPI CCS Tools: Static data parser library



- A library for parsing CCS static data binaries
 - Easily usable C data structure
- Facilitates driver implementation
- Common implementation avoids implementing new parsers for the same format
 - No need to try to adapt to bugs in different parsers

MIPI CCS and MIPI CCS Tools



- Available to non-members of MIPI Alliance and the open-source community
- Find out more:
 - <https://www.mipi.org/specifications/camera-command-set>
 - <https://www.mipi.org/code-projects/specification-software-code/mipi-ccs-tools>
- Get the Code: [Public Repository on Github](#)



Demonstration of MIPI CCS Tools and Linux CCS Driver

Demonstration



- Linux CCS driver
- Change CCS static data file (YAML) for a sensor
- Generate CCS static data binary
- Use the updated CCS static data binary in the driver

The background is a teal color with a dense pattern of small, light-colored icons representing various digital and communication concepts such as Wi-Fi, SMS, mobile phones, and social media. Overlaid on this is a network diagram consisting of several nodes (colored circles in orange, red, purple, and white) connected by thin white lines. The nodes are arranged in a roughly horizontal line across the top and middle of the page, with some lines extending downwards and outwards.

Questions & Answers