DTF DRIVER

(Sycamore Island)

Module Design Specification

Version: 1.0 Status: Draft

Date: 23-July-2020

Document Author: Anjana Joshi

Document List of Approvers:

Firmware Architect: Rahil Malhotra

Validation Architect: Mahaveer Addanki

Functional Safety Architect: Srikanth Kaniyanoor Srinivasan

Functional Safety Manager: Francesco Pingitore

Applicable Standard: IEC61508

Template Version: FuSa\_Documentation\_Word\_Template.docx v0.5

FuSa Lifecycle Phase: FS2

Intel Top Secret

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: [http://](http://www.intel.com/design/literature.htm)www.intel.com/design/literature.htm

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at h[ttp://www.intel.com/](http://www.intel.com/) or from the OEM or retailer.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2021, Intel Corporation. All rights reserved.

Contents

[1.0 Introduction 5](#_Toc514353440)

[1.1 Purpose 5](#_Toc514353441)

[1.2 Audience 5](#_Toc514353442)

[1.3 Acronyms and Terminology 5](#_Toc514353443)

[1.4 Reference Documents 6](#_Toc514353444)

[2.0 Document Scope 7](#_Toc514353445)

[2.1 Prerequisite Documents 7](#_Toc514353446)

[2.2 In Scope 7](#_Toc514353447)

[2.3 Out of Scope 7](#_Toc514353448)

[2.4 Known Gaps/Opens 7](#_Toc514353449)

[2.5 IEC61508 Fulfillment Matrix 8](#_Toc514353450)

[3.0 Global Design Description 9](#_Toc514353451)

[4.0 Functional Description](#_Toc514353451)

[4.1 Introduction](#_Toc514353452) 13

[4.2 Message Format](#_Toc514353453) 14

[4.3 SCI FW DTF Write 1](#_Toc514353454)5

4.4  [MipiSysT library 1](#_Toc514353454)6

4.5 [Configuration Paramters 1](#_Toc514353454)6

4.6 [Type Definitions 1](#_Toc514353454)6

4.7 [Exported API in DTF Driver 1](#_Toc514353454)7

4.8 [Debug Macro 1](#_Toc514353454)9

[5.0 Data Description](#_Toc514353457) 27

[Appendix A](#_Toc514353458) 28

Figures

[Figure 1. DTF Architecture](#_Toc514353459) 10

[Figure 2. Message Format](#_Toc514353459) 11

[Figure 3. SCI FW-DTF Flow](#_Toc514353459) 12

Tables

[Table 1. Terminology 5](#_Toc514353459)

[Table 2. Reference Documents 6](#_Toc514353460)

[Table 3. Open & Follow up Needs 7](#_Toc514353461)

[Table 4 IEC61508 Fulfillment Matrix 8](#_Toc514353462)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Author | Description |
| 07/18/18 | 0.30 | Anjana Joshi | Initial release for architectural and validation teams feedback |
| 02/26/19 | 0.50 | Anjana Joshi | Added mipi syst library details |
| 04/08/2019 | 0.80 | Anjana Joshi | Added additional details for mipi syst library and debug macro details |
| 05/23/2019 | 0.81 | Anjana Joshi | Added Wrapper API details and change of mipi API |
| 06/23/2020 | 1.0 | Anjana Joshi | Added minor changes |

# Introduction

## Purpose

This document describes the software architecture of DTF driver that will run on Sycamore Island (SCI) in Elkhart Lake.

## Audience

The audience for this document is as follows:

* Safety auditors and reviewers for SCI
* Product design and development team for Elkhart Lake platforms
* SCI firmware development and validation team
* FUSA validation team for SCI

## Acronyms and Terminology

Table 1. Terminology

| Term | Description |
| --- | --- |
| DTF | Debug Trace Fabric |
| ISI | Intel Safety Island |
| SCI | SCI stands for Sycamore Island |
| OSDL | OS Dependent Layer |
| OSIL | OS Independent Layer |

## Reference Documents

Table 2. Reference Documents

| Document | Document No./Location |
| --- | --- |
| *Sycamore Island Hardware Architecture Specification v0.8* | <https://sharepoint.amr.ith.intel.com/sites/FUSA_GlobalDomain/ISI/_layouts/WordViewer.aspx?id=/sites/FUSA_GlobalDomain/ISI/Safety%20Island%20WIP/Gen%200.6%20(EHL-MCC)/IP%20TR%20WG/SCI_HAS/Archive/SCI_HAS_0.8/SCI_HAS_v0.81.docx&Source=https%3A%2F%2Fsharepoint%2Eamr%2Eith%2Eintel%2Ecom%2Fsites%2FFUSA_GlobalDomain%2FISI%2FSafety%2520Island%2520WIP%2FForms%2FAllItems%2Easpx%3FRootFolder%3D%252Fsites%252Ffusa%255Fglobaldomain%252Fisi%252FSafety%2520Island%2520WIP%252FGen%25200%252E6%2520%2528EHL%252DMCC%2529%252FIP%2520TR%2520WG%252FSCI%255FHAS%252FArchive%252FSCI%255FHAS%255F0%252E8%26FolderCTID%3D0x0120005B0F9717B168964B8F65BC8E87C5640B%26InitialTabId%3DRibbon%252EDocument%26VisibilityContext%3DWSSTabPersistence&DefaultItemOpen=1> |
| *Sycamore Island Software Architecture Specification v1.0* | [~~https://sharepoint.amr.ith.intel.com/sites/FUSA\_GlobalDomain/ISI/Sycamore%20Island%20ISI%2006/Engineering/Specifications/Sycamore%20Island%20SAS%20v1.0.pdf~~](https://sharepoint.amr.ith.intel.com/sites/FUSA_GlobalDomain/ISI/Sycamore%20Island%20ISI%2006/Engineering/Specifications/Sycamore%20Island%20SAS%20v1.0.pdf) |
| *ISO 26262-2:2018* | Draft version available in FuSA Global SharePoint |
| *IEC61508* | Draft version available in FuSA Global SharePoint |
| *ThreadX User Guide* | <http://rtos.com/images/uploads/ThreadX_User_Guide_V5.pdf> |
| *Debug Trace Fabric HAS* |  |
| *DTF MAS 0.5* | <https://sharepoint.amr.ith.intel.com/sites/FUSA_GlobalDomain/ISI/Safety%20Island%20WIP/Gen%200.6%20(EHL-MCC)/IP%20TR%20WG/SCI_MAS/DTF%20MAS%200.5.docx> |
| *Mipi Sys T specification Document* | https://sharepoint.gar.ith.intel.com/sites/QSD/Shared%20Documents/Sycamore%20Island/SCI%20SW%20Module%20Design%20Spec/DTF%20Driver%20MDS/mipi\_SyS-T\_specification\_v1-0.pdf |

# Document Scope

## Prerequisite Documents

* SCI Software Architecture Specification documentation and SCI Hardware Architecture Specification documentation.
* mipi\_SyS-T specification documentation.

## In Scope

The DTF Driver Module Design Specification is intended to describe the implementation of the DTF driver and mipi SyS- T library layer. This description includes

* The DTF driver routines.
* Mipi Sys – T library routines.
* Functional behavior and internal design to the level of detail necessary for their implementation.

## Out of Scope

## Known Gaps/Opens

Table 3. Open & Follow up Needs

|  |  |
| --- | --- |
| **Open and follow-up needs** | **Notes** |
|  |  |
|  |  |
|  |  |

## IEC61508 Fulfillment Matrix

This section is not application to DTF driver and mipi syst library since it is QM component.

# Global Design Decisions

DTF Driver will have osil layer only as no thread x specific operations are needed. MipiSysT library will be the upper layer for DTF drive . DTF driver and MipiSyst Library are decided to be QM code since they will never be invoked in mission mode and will be used only in debug mode. Hence these are not required to adhere to MISRA C compliance and LDRA Compliance. However the code is aligned to LDRA as much as possible

# Functional Description

### 4.1 Introduction

* Debug Trace Fabric (DTF) is used to send trace information from a source to a destination to support post-silicon debug
* DTF Driver is used by any firmware module to send log messages over DTF interface to Host.

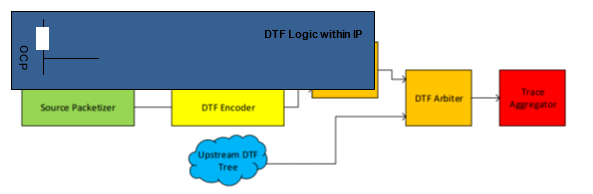


Figure 1: DTF Architecture

* Source Packetizer
  + Produces the debug data to be captured.
* Encoder
  + Takes the data produced by source packetizers and transfers it to the DTF backbone (a tree of arbiters)
* Arbiter
  + Takes two upstream inputs and produces one downstream output, forms the backbone
* Trace Aggregator
  + Sends the contents of the DTF and sends to the final destination
* SCI firmware uses DTF- source packetizer implemented by SCI hardware to push debug messages.
* The debug messages are transferred in a lossless fashion.

### 4.2 Message Format



Figure 2: Message Format

1. USER64 – D64 (1...N) – D64M is the desired output of the DTF port from SCI.
2. SCI FW will produce the message by initially pushing a USER64 packet, with SVEN Event Header information, by writing to USER32\_LO/HI\_REG.
3. Then, SCI will start pushing 32b data at a time, by writing to a DATA\_WITHOUT\_EOP register in SRCPKT.
4. Then, SCI FW will push the last 32b chunk data of the message, by writing to a DATA\_WITH\_EOP register.
   1. SRCPKT will append two chunks of 32b data, if possible, and push 64b data towards the SRCENC.
   2. If SCI FW writes the last chunk of the message on an odd # of 32b data, then SRCPKT will insert 32b zero padding and push it to the SRCENC.
5. SCI FW is responsible to embed the HH timer (in SCI case it should be ‘0’) into the message payload, as shown in the diagram above.
6. SRCENC will deliver the FW Debug message to the DTF fabric, in a lossless fashion.

### 4.3 SCI FW – DTF Flow

 Figure 3: SCI FW- DTF Flow

* SCI FW will push DW debug messages, during its code flow (under debug), into SCI local SRAM buffer DTFBUF of size 16DW.
* When DTFBUF is full, SCI FW will push the DTFBUF contents onto the DTF packetizer interface.
* SRCENC will deliver the FW Debug message to the DTF fabric, in a lossless fashion.
* Writing event header information to addresses USR64\_LO\_ADDR and USR64\_HI\_ADDR sends out USR64 message type to Encoder.
* Writing to DATA\_WITHOUT\_EOP address range sends out D64 message type to Encoder.
* Writing to DATA\_WITH\_EOP address in packetizer sends out D64M message type to Encode. The last 32bytes of data is written to this register. DATA\_WITH\_EOP indicates end of packet data.

##### 4.4 MipiSysT library

MIPI System Software - Trace is a common data format for transmitting software trace and debug information between a test system and a device, such as a system-on-chip (SoC) or platform. SyS-T provides a platform-independent general purpose trace protocol and software instrumentation. Library

MiPi SysT Message types 1. Build Messages 2. Short Messages 3. String Messages 4. Catalog Messages 5. Raw messages 6. Clock Messages

We will be using short message and raw message format for SCI.



Figure 4: Mipi Sys T Call Flow.

The above diagram shows the call flow of Mipi Sys T library. The SCI firmware module calls a DEBUG macro which is mapped to wrapper API for dtf call. Before calling dtf wrapper API configuration parameter Mission mode is checked. If mission mode is enabled dtf module will be disabled else it will call Sys – T API which maps to SyS – T library implementation. Mipi SyS –T Api will in turn call dtf driver API.

## Configuration Parameters

A configuration parameter will be added to know whether system is in mission mode. In Mission mode DTF is disabled.

## Type Definitions

#### Type definition of mipi\_syst\_handle

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax: | struct mipi\_syst\_handle | | |
| Type: | Struct | | |
| File: | mipi\_syst.h | | |
| Members | mipi\_syst\_header\* systh\_header | struct | Global state |
| mipi\_syst\_handle\_flags systh\_flags | struct | handle state |
| mipi\_syst\_msg\_tag systh\_tag | struct | tag flags |
| mipi\_syst\_guid systh\_guid; | struct | module guid |
| systh\_param\_count | mipi\_syst\_u32 | Number of parameters |
| systh\_param[6] | mipi\_syst\_u32 | Catalog msg paramters |
| mipi\_syst\_platform\_handle systh\_platform | struct | Platform specific data |
|  |  |  |  |
| Description: | This enum holds the type of event the interrupt handler receives | | |

#### Type definition of mipi\_syst\_handle\_flags

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax: | mipi\_syst\_handle\_flags | | |
| Type: | Struct | | |
| File: | mipi\_syst.h | | |
| Members | shf\_alloc: | mipi\_syst\_u32 | Set to 1 if heap allocated handle |
| Description: | This structure holds syst internal handle state flags | | |

#### Type definition of mipi\_syst\_msg\_tag

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax: | struct mipi\_syst\_msg\_tag | | |
| Type: | Struct | | |
| File: | mipi\_syst.h | | |
| Members | et\_type : 4 | mipi\_syst\_u32 | SyS-T message type ID |
| et\_severity : 3 | mipi\_syst\_u32 | severity level of message |
| et\_res7 : 1 | mipi\_syst\_u32 | reserved for future use |
| et\_location : 1 | mipi\_syst\_u32 | Indicate location information |
| et\_length : 1 | mipi\_syst\_u32 | indicate length field |
| et\_chksum : 1 | mipi\_syst\_u32 | indicate 32-bit CRC |
| et\_timestamp : 1 | mipi\_syst\_u32 | indicate 64-bit timestamp |
|  | et\_modunit : 11 | mipi\_syst\_u32 | unit for GUID or module:unit |
|  | et\_guid : 1; | mipi\_syst\_u32 | 128-bit GUID present |
|  | et\_subtype : 6 | mipi\_syst\_u32 | type dependent sub category |
|  | et\_res30 : 1; | mipi\_syst\_u32 | reserved for future use |
|  | et\_res31 : 1; | mipi\_syst\_u32 | reserved for future use |
| Description: | This enum holds the type of event the interrupt handler receives | | |

#### Type definition of mipi\_syst\_guid

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax: | struct mipi\_syst\_guid | | |
| Type: | Struct | | |
| File: | mipi\_syst.h | | |
| Members | U | union |  |
| mipi\_syst\_u8 b[16];  mipi\_syst\_u64 ll[2] | Union members | Message id |
| Description: | 128-bit GUID style message origin ID | | |
|  |  | | |

#### Type definition of mipi\_syst\_platform\_handle

|  |  |  |  |
| --- | --- | --- | --- |
| Syntax: | struct mipi\_syst\_platform\_handle | | |
| Type: | Struct | | |
| File: | Platform.h | | |
| Members | sph\_io\_count | mipi\_syst\_u32 | cnt io's, used for pretty printing |
| sph\_raw\_coun | mipi\_syst\_u32 | number of raw bytes in sph\_raw |
|  | sph\_raw[1024] | mipi\_syst\_u8 | buffer for printing |
| Description: | This structure connects the instrumentation API with the underlying SyS-T infrastructure. It plays a similar role to a FILE \* in traditional  C file IO. | | |

## Exported API in DTF driver

## 4.5.1 DTF OSIL Write

|  |  |  |  |
| --- | --- | --- | --- |
| Service name: | OSIL DTF Driver Write | | |
| Syntax: | UINT dtf\_raw\_write (void const \*payload, uint32\_t size) | | |
| Sync/Async: | Synchronous | | |
| Reentrant: | Reentrant | | |
| Parameters (in): | \*payload | Void | Pointer to payload |
|  | Size | uint32\_t | Size of the payload |
| Return value: | STATUS\_SUCCESS | | DTF write is successful |
|  | STATUS\_FIRMWARE(M\_DTF, E\_IN\_WRITE, E\_INVALID\_ARGUMENT | | DTF write faied |
| Description: | dtf\_write will be called by mipiSysT library to push debug messages to DTF Fabric. Osdl layer copies debug messages to SRAM buffer, OSIL layer reads contents of buffer and writes to DATA\_WITH/WITHOUT\_EOP register | | |
| Preconditions: |  | | |
| Timing Constraints: |  | | |
| Caveats: |  | | |
| Configuration: |  | | |
| Error Handling: |  | | |
| Implementation comments |  | | |
| Design Decisions |  | | |
| SAS traceability | Private Function | | |

## 4.5.2 Mipi Sys-T Printf

|  |  |  |  |
| --- | --- | --- | --- |
| Service name: | Mipi Sys T write printf string | | |
| Syntax: | Void mipi\_syst\_write\_printf\_string(struct mipi\_syst\_handle\* svh,  struct mipi\_syst\_msglocation\* loc,  enum mipi\_syst\_severity severity,  const char \*fmt,  uint32\_t status) | | |
| Sync/Async: | Synchronous | | |
| Reentrant: | Reentrant | | |
| Parameters (in): | \*svh | struct mipi\_syst\_handle | SyS-T handle |
|  | \*loc | struct mipi\_syst\_msglocation | pointer to instrumentation location or null |
|  | severity | enum mipi\_syst\_severity | severity message severity level (0..7) |
|  | fmt | Const char \* | String to be printed |
|  | status | Uint32\_t | Status of the INFO or ERROR print |
|  |  |  |  |
| Return value: |  | |  |
|  |  | |  |
| Description: | This api will be called from SCI Firmware modules using the wrapper  MIPI\_SYST\_PRINTF | | |
| Preconditions: | The mipi syst macro will check for configuration parameter. If prints are enabled it calls the library and dtf layer. | | |
| Timing Constraints: |  | | |
| Caveats: |  | | |
| Configuration: |  | | |
| Error Handling: |  | | |
| Implementation comments |  | | |
| Design Decisions | Mipi Sys T is a QM Component. It does not have osil and osdl layers | | |
| SAS traceability | Private Function | | |

## 4.5.3 DTF Wrapper API

|  |  |  |  |
| --- | --- | --- | --- |
| Service name: | DTF Printf | | |
| Syntax: | Void dtf\_printf(int log\_level, const char \*fmt, uint32\_t status) | | |
| Sync/Async: | Synchronous | | |
| Reentrant: | Reentrant | | |
| Parameters (in): | log level | void | Log levels INFO WARNING OR ERROR |
|  | fmt | char \* | String to be printed |
|  | status | uint32\_t | Variable to print status of error or info |
| Return value: |  | |  |
| Description: | dtf printf will be mapped to debug macro which will be called from firmware modules. The wrapper API calls MIPI syst init and MIPI printf API. | | |
| Preconditions: |  | | |
| Timing Constraints: |  | | |
| Caveats: |  | | |
| Configuration: |  | | |
| Error Handling: | Errors are printed at dtf driver level if prints are enabled. | | |
| Implementation comments |  | | |
| Design Decisions | Wrapper API is part of sci service layer | | |
| SAS traceability | 849300 | | |

### 4.6 Imported Types

None

1. Data Description

#define DEBUG(log\_level, fmt, status) dtf\_printf(log\_level, fmt, status)

#define SCI\_DTF\_DATA\_WITH\_EOP\_REG (SCI\_DTF\_REGISTERS\_I\_BASE + (0xFFCu))



Doxygen documentation

# 6. Module Documentation

## API Sets

### Modules

* **State and Lifetime handling macros**
* **Raw Data Writing Macros**

### Detailed Description

SyS-T provided Instrumentation API sets

SyS-T provides different API sets. Most sets can be individually enabled or disabled using the PCFG\_ApiSet platform feature defines.

## State and Lifetime handling macros

### Macros

* #define **MIPI\_SYST\_INIT\_STATE**(s, f, p)  **mipi\_syst\_init**((s), (f), (p))
* #define **MIPI\_SYST\_SHUTDOWN\_STATE**(s, f)  **mipi\_syst\_destroy**((s), (f))
* #define **MIPI\_SYST\_INIT**(f, p)  **MIPI\_SYST\_INIT\_STATE**((struct mipi\_syst\_header\*)0, (f), (p))
* #define **MIPI\_SYST\_SHUTDOWN**(f)  **MIPI\_SYST\_SHUTDOWN\_STATE**((struct mipi\_syst\_header\*)0, (f))
* #define **MIPI\_SYST\_INIT\_HANDLE\_STATE**(s, h, p)  **mipi\_syst\_init\_handle**((s), (h), (p), 0)
* #define **MIPI\_SYST\_INIT\_HANDLE**(h, p)  **MIPI\_SYST\_INIT\_HANDLE\_STATE**((struct mipi\_syst\_header\*)0, (h), (p))
* #define **MIPI\_SYST\_ALLOC\_HANDLE**(p)  CFG\_ERROR\_ALLOC\_HANDLE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY
* #define **MIPI\_SYST\_ALLOC\_HANDLE\_STATE**(s, p)  CFG\_ERROR\_SYST\_ALLOC\_HANDLE\_STATE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_LENGTH**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_LENGTH**(h)  (((h) &&(h)->systh\_tag.et\_length) ? 1 : 0)
* #define **MIPI\_SYST\_SET\_HANDLE\_MODULE\_UNIT**(h, module, unit)
* #define **MIPI\_SYST\_SET\_HANDLE\_GUID\_UNIT**(p, g, u)  CFG\_ERROR\_SET\_HANDLE\_GUID\_UNIT\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_ORIGIN\_GUID
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP**(h, v)  CFG\_ERROR\_MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_TIMESTAMP
* #define **MIPI\_SYST\_DELETE\_HANDLE**(h)  **mipi\_syst\_delete\_handle**(h)

### Detailed Description

State and handle lifetime related macros

### Macro Definition Documentation

#### #define MIPI\_SYST\_ALLOC\_HANDLE( p)  CFG\_ERROR\_ALLOC\_HANDLE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY

#### #define MIPI\_SYST\_ALLOC\_HANDLE\_STATE( s, p)  CFG\_ERROR\_SYST\_ALLOC\_HANDLE\_STATE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY

#### #define MIPI\_SYST\_DELETE\_HANDLE( h)  mipi\_syst\_delete\_handle(h)

Delete a SyS-T handle

##### Parameters:

|  |  |
| --- | --- |
| *h* | SyS-T handle from **MIPI\_SYST\_INIT\_HANDLE** or **MIPI\_SYST\_ALLOC\_HANDLE** |

Example:

void foo()

{

struct mipi\_syst\_handle\* svh = MIPI\_SYST\_ALLOC\_HANDLE(NULL);

// ...

MIPI\_SYST\_DELETE\_HANDLE(svh);

}

#### #define MIPI\_SYST\_ENABLE\_HANDLE\_LENGTH( h, v)

#### #define MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP( h, v)  CFG\_ERROR\_MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_TIMESTAMP

#### #define MIPI\_SYST\_GET\_HANDLE\_LENGTH( h)  (((h) &&(h)->systh\_tag.et\_length) ? 1 : 0)

Get length field generation state from given SyS-T handle

##### Parameters:

|  |  |
| --- | --- |
| *h* | SyS-T handle from **MIPI\_SYST\_INIT\_HANDLE** or **MIPI\_SYST\_ALLOC\_HANDLE** |

##### Returns:

0 if disabled, otherwise enabled

Example:

struct mipi\_syst\_handle\* svh = MIPI\_SYST\_ALLOC\_HANDLE(NULL);

if (MIPI\_SYST\_GET\_HANDLE\_LENGTH(svh)) {

// length field enabled ...

}

#### #define MIPI\_SYST\_INIT( f, p)  MIPI\_SYST\_INIT\_STATE((struct mipi\_syst\_header\*)0, (f), (p))

SyS-T global platform initialization

This function must be called during system startup to initialize the SyS-T execution environment.

##### Parameters:

|  |  |
| --- | --- |
| *f* | pointer to platform initialization hook function |
| *p* | pointer value that gets passed to the initialization hook function |

#### #define MIPI\_SYST\_INIT\_HANDLE( h, p)  MIPI\_SYST\_INIT\_HANDLE\_STATE((struct mipi\_syst\_header\*)0, (h), (p))

Initialize non-heap SyS-T handle

This function is used in platforms that don't support heap allocations. The caller has to provide a pointer to a memory location that can hold a mipi\_syst\_handle data structure.

##### Parameters:

|  |  |
| --- | --- |
| *h* | Pointer to handle data structure on the stack or data segment. |
| *p* | Pointer to mipi\_syst\_origin structure with client identifying information, or NULL if not used |

Example

static struct mipi\_syst\_handle systh\_data;

void foo()

{

struct mipi\_syst\_handle\* svh = MIPI\_SYST\_INIT\_HANDLE(&systh\_data, NULL);

...

}

#### #define MIPI\_SYST\_INIT\_HANDLE\_STATE( s, h, p)  mipi\_syst\_init\_handle((s), (h), (p), 0)

Initialize non-heap SyS-T handle with custom global state

This function is used in platforms that don't support heap allocations. The caller has to provide a pointer to a memory location that can hold a mipi\_syst\_handle data structure. This function expect a user provided SyS-T state structure pointer as its first parameter. To create a handle for the shared global state, call **MIPI\_SYST\_INIT\_HANDLE(h,p)** instead.

##### Parameters:

|  |  |
| --- | --- |
| *s* | Pointer to SyS-T state header variable |
| *h* | Pointer to handle data structure on the stack or data segment. |
| *p* | Pointer to data that get passed to the platform handle init hook function. |

Example

extern struct mipi\_syst\_header systh\_header;

static struct mipi\_syst\_handle systh\_data;

void foo()

{

struct mipi\_syst\_handle\* svh;

svh = MIPI\_SYST\_INIT\_HANDLE\_STATE(&systh\_header, &systh\_data, NULL);

...

}

#### #define MIPI\_SYST\_INIT\_STATE( s, f, p)  mipi\_syst\_init((s), (f), (p))

SyS-T platform initialization with user provided state structure

This function must be called during system startup to initialize the SyS-T execution environment. This function expects a user provided SyS-T state structure pointer. This call supports environments with different library states at the same time. To initialize SyS-T for using a global shared state, call **MIPI\_SYST\_INIT(f,p)** instead.

##### Parameters:

|  |  |
| --- | --- |
| *s* | Pointer to SyS-T state header variable |
| *f* | Pointer to platform initialization hook function |
| *p* | Pointer value that gets passed to the initialization hook function |

#### #define MIPI\_SYST\_SET\_HANDLE\_GUID\_UNIT( p, g, u)  CFG\_ERROR\_SET\_HANDLE\_GUID\_UNIT\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_ORIGIN\_GUID

#### #define MIPI\_SYST\_SET\_HANDLE\_MODULE\_UNIT( h, module, unit)

**Value:**do { \

if (h) { \

(h)->systh\_tag.et\_guid = 0; \

(h)->systh\_tag.et\_modunit = \

\_MIPI\_SYST\_MK\_MODUNIT\_ORIGIN((module), (unit)); \

} \

} while (0)

Change module and unit ID of the given SyS-T handle.

##### Parameters:

|  |  |
| --- | --- |
| *h* | SyS-T handle from **MIPI\_SYST\_INIT\_HANDLE** or **MIPI\_SYST\_ALLOC\_HANDLE** |
| *module* | module id (0..0x7F) |
| *unit* | unit id (0x0..0xF) |

##### See also:

**MIPI\_SYST\_SET\_HANDLE\_GUID\_UNIT** mipi\_syst\_msg\_tag

Example:

struct mipi\_syst\_handle\* svh = MIPI\_SYST\_ALLOC\_HANDLE(NULL);

// tag message with 5:1 as module:unit id pair

//

MIPI\_SYST\_SET\_MODULE\_UNIT(svh, 5, 1);

#### #define MIPI\_SYST\_SHUTDOWN( f)  MIPI\_SYST\_SHUTDOWN\_STATE((struct mipi\_syst\_header\*)0, (f))

SyS-T global platform shutdown

##### Parameters:

|  |  |
| --- | --- |
| *f* | pointer to platform resource destruction hook function |

#### #define MIPI\_SYST\_SHUTDOWN\_STATE( s, f)  mipi\_syst\_destroy((s), (f))

SyS-T platform shutdown with user provided state structure

This function expects a user provided SyS-T state structure pointer. This call supports environments with different library states at the same time. To shutdown SyS-T using a global shared state, call **MIPI\_SYST\_SHUTDOWN(f)** instead.

##### Parameters:

|  |  |
| --- | --- |
| *s* | Pointer to SyS-T state header variable |
| *f* | pointer to platform resource destruction hook function |

## Raw Data Writing Macros

### Modules

* **Build Number Message Macros**
* **Printf style catalog Message Macros**

### Macros

* #define **MIPI\_SYST\_SHORT32**(h, v)
* #define **MIPI\_SYST\_SHORT64**(h, v)
* #define **\_MIPI\_SYST\_MK\_PARAM\_LIST**(tag, p)  \_MIPI\_SYST\_CATARG\_##tag, p
* #define **MIPI\_SYST\_PARAM\_INT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(D, (p))
* #define **MIPI\_SYST\_PARAM\_LONG**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LD, (p))
* #define **MIPI\_SYST\_PARAM\_LONGLONG**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LLD, (p))
* #define **MIPI\_SYST\_PARAM\_SIZE\_T**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(ZD, (p))
* #define **MIPI\_SYST\_PARAM\_PTRDIFF\_T**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(TD, (p))
* #define **MIPI\_SYST\_PARAM\_FLOAT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(F, (p))
* #define **MIPI\_SYST\_PARAM\_DOUBLE**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(F, (p))
* #define **MIPI\_SYST\_PARAM\_LONGDOUBLE**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LF, (p))
* #define **MIPI\_SYST\_PARAM\_CHAR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(C, (p))
* #define **MIPI\_SYST\_PARAM\_SHORT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(HHD, (p))
* #define **MIPI\_SYST\_PARAM\_WCHAR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LC, (p))
* #define **MIPI\_SYST\_PARAM\_PTR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(P, (p))
* #define **MIPI\_SYST\_PARAM\_CSTR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(CSTR, (p))
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_CHECKSUM**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_CHECKSUM**(h)  0
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_COUNTER**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_COUNTER**(h)  0
* #define **MIPI\_SYST\_GET\_HANDLE\_TIMESTAMP**(h)  0
* #define **MIPI\_SYST\_WRITE**(h, sev, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOCADDR**(h, sev, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOC16**(h, sev, f, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOC32**(h, sev, f, id, p, len)
* #define **MIPI\_SYST\_DEBUG**(svh, sev, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOCADDR**(svh, sev, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOC16**(svh, sev, file, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOC32**(svh, sev, file, str, len)
* #define **MIPI\_SYST\_FUNC\_ENTER**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOCADDR**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOC16**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOC32**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_EXIT**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOCADDR**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOC16**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOC32**(svh, sev, file)
* #define **MIPI\_SYST\_ASSERT**(svh, sev, cond)
* #define **MIPI\_SYST\_ASSERT\_LOC16**(svh, sev, file, cond)
* #define **MIPI\_SYST\_ASSERT\_LOC32**(svh, sev, file, cond)
* #define **MIPI\_SYST\_ASSERT\_LOCADDR**(svh, sev, cond)
* #define **MIPI\_SYST\_CLOCK\_SYNC**(h, c, f)
* #define **MIPI\_SYST\_PRINTF**(...)
* #define **MIPI\_SYST\_PRINTF\_LOC16**(...)
* #define **MIPI\_SYST\_PRINTF\_LOC32**(...)
* #define **MIPI\_SYST\_PRINTF\_LOCADDR**(...)
* #define **MIPI\_SYST\_CATPRINTF64**(...)
* #define **MIPI\_SYST\_CATPRINTF64\_LOC16**(...)
* #define **MMIPI\_SYST\_CATPRINTF64\_LOC32**(...)
* #define **MIPI\_SYST\_CATPRINTF64\_LOCADDR**(...)
* #define **MIPI\_SYST\_CATPRINTF32**(...)
* #define **MIPI\_SYST\_CATPRINTF32\_LOC16**(...)
* #define **MMIPI\_SYST\_CATPRINTF32\_LOC32**(...)
* #define **MIPI\_SYST\_CATPRINTF32\_LOCADDR**(...)

### Enumerations

* enum **mipi\_syst\_catalog\_parameter\_types** { **\_MIPI\_SYST\_CATARG\_END** = 0, **\_MIPI\_SYST\_CATARG\_D** = 1, **\_MIPI\_SYST\_CATARG\_LD** = 2, **\_MIPI\_SYST\_CATARG\_LLD** = 3, **\_MIPI\_SYST\_CATARG\_ZD** = 4, **\_MIPI\_SYST\_CATARG\_TD** = 5, **\_MIPI\_SYST\_CATARG\_F** = 6, **\_MIPI\_SYST\_CATARG\_LF** = 7, **\_MIPI\_SYST\_CATARG\_C** = 8, **\_MIPI\_SYST\_CATARG\_HHD** = 9, **\_MIPI\_SYST\_CATARG\_LC** = 10, **\_MIPI\_SYST\_CATARG\_P** = 11, **\_MIPI\_SYST\_CATARG\_CSTR** = 12 }

### Detailed Description

Raw data writing macros:

### Macro Definition Documentation

#### #define \_MIPI\_SYST\_MK\_PARAM\_LIST( tag, p)  \_MIPI\_SYST\_CATARG\_##tag, p

#### #define MIPI\_SYST\_ASSERT( svh, sev, cond)

#### #define MIPI\_SYST\_ASSERT\_LOC16( svh, sev, file, cond)

#### #define MIPI\_SYST\_ASSERT\_LOC32( svh, sev, file, cond)

#### #define MIPI\_SYST\_ASSERT\_LOCADDR( svh, sev, cond)

#### #define MIPI\_SYST\_CATPRINTF32( *...*)

#### #define MIPI\_SYST\_CATPRINTF32\_LOC16( *...*)

#### #define MIPI\_SYST\_CATPRINTF32\_LOCADDR( *...*)

#### #define MIPI\_SYST\_CATPRINTF64( *...*)

#### #define MIPI\_SYST\_CATPRINTF64\_LOC16( *...*)

#### #define MIPI\_SYST\_CATPRINTF64\_LOCADDR( *...*)

#### #define MIPI\_SYST\_CLOCK\_SYNC( h, c, f)

#### #define MIPI\_SYST\_DEBUG( svh, sev, str, len)

#### #define MIPI\_SYST\_DEBUG\_LOC16( svh, sev, file, str, len)

#### #define MIPI\_SYST\_DEBUG\_LOC32( svh, sev, file, str, len)

#### #define MIPI\_SYST\_DEBUG\_LOCADDR( svh, sev, str, len)

#### #define MIPI\_SYST\_ENABLE\_HANDLE\_CHECKSUM( h, v)

#### #define MIPI\_SYST\_ENABLE\_HANDLE\_COUNTER( h, v)

#### #define MIPI\_SYST\_FUNC\_ENTER( svh, sev)

#### #define MIPI\_SYST\_FUNC\_ENTER\_LOC16( svh, sev, file)

#### #define MIPI\_SYST\_FUNC\_ENTER\_LOC32( svh, sev, file)

#### #define MIPI\_SYST\_FUNC\_ENTER\_LOCADDR( svh, sev)

#### #define MIPI\_SYST\_FUNC\_EXIT( svh, sev)

#### #define MIPI\_SYST\_FUNC\_EXIT\_LOC16( svh, sev, file)

#### #define MIPI\_SYST\_FUNC\_EXIT\_LOC32( svh, sev, file)

#### #define MIPI\_SYST\_FUNC\_EXIT\_LOCADDR( svh, sev)

#### #define MIPI\_SYST\_GET\_HANDLE\_CHECKSUM( h)  0

#### #define MIPI\_SYST\_GET\_HANDLE\_COUNTER( h)  0

#### #define MIPI\_SYST\_GET\_HANDLE\_TIMESTAMP( h)  0

#### #define MIPI\_SYST\_PARAM\_CHAR( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(C, (p))

#### #define MIPI\_SYST\_PARAM\_CSTR( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(CSTR, (p))

#### #define MIPI\_SYST\_PARAM\_DOUBLE( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(F, (p))

#### #define MIPI\_SYST\_PARAM\_FLOAT( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(F, (p))

#### #define MIPI\_SYST\_PARAM\_INT( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(D, (p))

#### #define MIPI\_SYST\_PARAM\_LONG( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(LD, (p))

#### #define MIPI\_SYST\_PARAM\_LONGDOUBLE( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(LF, (p))

#### #define MIPI\_SYST\_PARAM\_LONGLONG( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(LLD, (p))

#### #define MIPI\_SYST\_PARAM\_PTR( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(P, (p))

#### 39 .#define MIPI\_SYST\_PARAM\_PTRDIFF\_T( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(TD, (p))

#### 40. #define MIPI\_SYST\_PARAM\_SHORT( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(HHD, (p))

#### 41. #define MIPI\_SYST\_PARAM\_SIZE\_T( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(ZD, (p))

#### 42. #define MIPI\_SYST\_PARAM\_WCHAR( p)  \_MIPI\_SYST\_MK\_PARAM\_LIST(LC, (p))

#### 43.#define MIPI\_SYST\_PRINTF( *...*)

#### 44.#define MIPI\_SYST\_PRITF\_LOC16( *...*)

#### 45.#define MIPI\_SYST\_PRINTF\_LOC32( *...*)

#### 46#define MIPI\_SYST\_PRINTF\_LOCADDR( *...*)

#### 47#define MIPI\_SYST\_SHORT32( h, v)

**Value:**MIPI\_SYST\_OUTPUT\_D32MTS(h,\

(((v)<<4) | (mipi\_syst\_u32)MIPI\_SYST\_TYPE\_SHORT32))

Send short data message with 28-bit payload.

This API is indented for space and speed restricted environments that cannot support the more complex message types.

##### Parameters:

|  |  |
| --- | --- |
| *h* | mipi\_syst\_handle\* SyS-T handle |
| *v* | 28-bit output value. The upper 4 bits of the parameter value are discarded |

Example:

#define INIT\_START 1

#define INIT\_DONE 2

foo()

{

MIPI\_SYST\_SHORT32(svh, INIT\_START);

// processing ..

//

MIPI\_SYST\_SHORT32(svh, INIT\_DONE);

// ..

}

#### #define MIPI\_SYST\_SHORT64( h, v)

**Value:**MIPI\_SYST\_OUTPUT\_D64MTS(h,\

(((v)<<4) | (mipi\_syst\_u64)MIPI\_SYST\_TYPE\_SHORT64))

Send short data message with 60-bit payload.

This API is indented for space and speed restricted environments that cannot support the more complex message types.

##### Parameters:

|  |  |
| --- | --- |
| *h* | mipi\_syst\_handle\* SyS-T handle |
| *v* | 60-bit output value. The upper 4 bits of the parameter value are discarded. |

Example:

#define INIT\_START 1

#define INIT\_DONE 2

foo()

{

MIPI\_SYST\_SHORT64(svh, INIT\_START);

// processing ..

//

MIPI\_SYST\_SHORT64(svh, INIT\_DONE);

// ..

}

#### #define MIPI\_SYST\_WRITE( h, sev, id, p, len)

#### #define MIPI\_SYST\_WRITE\_LOC16( h, sev, f, id, p, len)

#### #define MIPI\_SYST\_WRITE\_LOC32( h, sev, f, id, p, len)

#### #define MIPI\_SYST\_WRITE\_LOCADDR( h, sev, id, p, len)

#### #define MMIPI\_SYST\_CATPRINTF32\_LOC32( *...*)

#### #define MMIPI\_SYST\_CATPRINTF64\_LOC32( *...*)

### Enumeration Type Documentation

#### enum mipi\_syst\_catalog\_parameter\_types

Parameter encoding values for vararg style catalog APIs

##### Enumerator:

|  |  |
| --- | --- |
| \_MIPI\_SYST\_CATARG\_END | end of parameter list |
| \_MIPI\_SYST\_CATARG\_D | int like d |
| \_MIPI\_SYST\_CATARG\_LD | long like ld |
| \_MIPI\_SYST\_CATARG\_LLD | long long like lld |
| \_MIPI\_SYST\_CATARG\_ZD | size\_t like z |
| \_MIPI\_SYST\_CATARG\_TD | ptrdiff\_t like t |
| \_MIPI\_SYST\_CATARG\_F | double like f |
| \_MIPI\_SYST\_CATARG\_LF | long double like lf |
| \_MIPI\_SYST\_CATARG\_C | char like c |
| \_MIPI\_SYST\_CATARG\_HHD | short like hhd |
| \_MIPI\_SYST\_CATARG\_LC | long char like lc |
| \_MIPI\_SYST\_CATARG\_P | void \* like p or n |
| \_MIPI\_SYST\_CATARG\_CSTR | char \* like s |

## Build Number Message Macros

### Macros

* #define **MIPI\_SYST\_BUILD\_COMPACT32**(h, n)
* #define **MIPI\_SYST\_BUILD\_COMPACT64**(h, n)

### Functions

* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_init** (struct mipi\_syst\_header \*header, mipi\_syst\_inithook\_t pfinit, const void \*init\_data)
* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_destroy** (struct mipi\_syst\_header \*header, mipi\_syst\_destroyhook\_t pfdestroy)
* **MIPI\_SYST\_EXPORT** struct mipi\_syst\_handle \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_init\_handle** (struct mipi\_syst\_header \*header, struct mipi\_syst\_handle \*svh, const struct mipi\_syst\_origin \*origin, **mipi\_syst\_u32** heap)
* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_delete\_handle** (struct mipi\_syst\_handle \*svh)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_file\_location32** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u16** f, **mipi\_syst\_u16** l)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_file\_location64** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** f, **mipi\_syst\_u32** l)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_address\_location** (struct mipi\_syst\_handle \*h, void \*p)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param0** (struct mipi\_syst\_handle \*h)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param1** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param2** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param3** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param4** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param5** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param6** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5, **mipi\_syst\_u32** p6)

### Detailed Description

Build Number writing macros:

### Macro Definition Documentation

#### #define MIPI\_SYST\_BUILD\_COMPACT32( h, n)

**Value:**if (h) { MIPI\_SYST\_OUTPUT\_D32MTS(h, (\

(((n) & 0x000FFFFF) << 4) | \

((n) & 0xFFF00000) << 10));}

Send compact client build number message with 22-bit payload.

This API is indented for space and speed restricted environments that cannot support the more complex message types.

##### Parameters:

|  |  |
| --- | --- |
| *h* | mipi\_syst\_handle\* SyS-T handle |
| *n* | 22-bit output value. The upper 10 bits of the parameter value are discarded. |

Example:

#define MY\_BUILDID 0x3FFFFF

foo()

{

MIPI\_SYST\_BUILD\_COMPACT32(svh, MY\_BUILDID);

}

#### #define MIPI\_SYST\_BUILD\_COMPACT64( h, n)

**Value:**if (h) { MIPI\_SYST\_OUTPUT\_D64MTS(h, (\

(0x00000000001000000ull | \

((n) & 0x00000000000FFFFFull) << 4) | \

((n) & 0xFFFFFFFFFFF00000ull) << 10));}

Send compact client build number message with 54-bit payload.

This API is indented for space and speed restricted environments that cannot support the more complex message types.

##### Parameters:

|  |  |
| --- | --- |
| *h* | mipi\_syst\_handle\* SyS-T handle |
| *n* | 54-bit output value. The upper 10 bits of the parameter value are discarded. |

Example:

#define MY\_BUILDID 0x3FFFFFFFFFFFFFull

foo()

{

MIPI\_SYST\_BUILD\_COMPACT64(svh, MY\_BUILDID);

}

### Function Documentation

#### MIPI\_SYST\_EXPORT void MIPI\_SYST\_CALLCONV mipi\_syst\_delete\_handle (struct mipi\_syst\_handle \* *svh*)

#### MIPI\_SYST\_EXPORT void MIPI\_SYST\_CALLCONV mipi\_syst\_destroy (struct mipi\_syst\_header \* *header*, mipi\_syst\_destroyhook\_t *pfdestroy*)

#### MIPI\_SYST\_EXPORT void MIPI\_SYST\_CALLCONV mipi\_syst\_init (struct mipi\_syst\_header \* *header*, mipi\_syst\_inithook\_t *pfinit*, const void \* *init\_data*)

#### MIPI\_SYST\_EXPORT struct mipi\_syst\_handle\* MIPI\_SYST\_CALLCONV mipi\_syst\_init\_handle (struct mipi\_syst\_header \* *header*, struct mipi\_syst\_handle \* *svh*, const struct mipi\_syst\_origin \* *origin*, mipi\_syst\_u32 *heap*)

#### MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation\* MIPI\_SYST\_CALLCONV mipi\_syst\_make\_address\_location (struct mipi\_syst\_handle \* *h*, void \* *p*)

#### MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation\* MIPI\_SYST\_CALLCONV mipi\_syst\_make\_file\_location32 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u16 *f*, mipi\_syst\_u16 *l*)

#### MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation\* MIPI\_SYST\_CALLCONV mipi\_syst\_make\_file\_location64 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *f*, mipi\_syst\_u32 *l*)

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param0 (struct mipi\_syst\_handle \* *h*)

Setup handle for 0 parameters passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param1 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*)

Setup handle for 1 parameter passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param2 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*, mipi\_syst\_u32 *p2*)

Setup handle for 2 parameters passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param3 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*, mipi\_syst\_u32 *p2*, mipi\_syst\_u32 *p3*)

Setup handle for 3 parameters passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param4 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*, mipi\_syst\_u32 *p2*, mipi\_syst\_u32 *p3*, mipi\_syst\_u32 *p4*)

Setup handle for 4 parameters passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param5 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*, mipi\_syst\_u32 *p2*, mipi\_syst\_u32 *p3*, mipi\_syst\_u32 *p4*, mipi\_syst\_u32 *p5*)

Setup handle for 5 parameters passed to catid message.

#### MIPI\_SYST\_INLINE void MIPI\_SYST\_CALLCONV mipi\_syst\_make\_param6 (struct mipi\_syst\_handle \* *h*, mipi\_syst\_u32 *p1*, mipi\_syst\_u32 *p2*, mipi\_syst\_u32 *p3*, mipi\_syst\_u32 *p4*, mipi\_syst\_u32 *p5*, mipi\_syst\_u32 *p6*)

Setup handle for 6 parameters passed to catid message.

## Printf style catalog Message Macros

### Macros

* #define **MIPI\_SYST\_CATPRINTF32\_0**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOCADDR**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOCADDR(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOCADDR**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOCADDR(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOCADDR**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOCADDR(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOCADDR(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOC16**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC16(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOC16**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC16(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOC16**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC16(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC16(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOC32**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC32(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOC32**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC32(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOC32**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC32(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC32(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOCADDR**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOCADDR(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOCADDR**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOCADDR(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOCADDR**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOCADDR(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOCADDR(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOC16**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC16(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOC16**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC16(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOC16**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC16(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC16(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOC32**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC32(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOC32**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC32(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOC32**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC32(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC32(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

### Detailed Description

Printf style catalog message instrumentation API wrappers. The MIPI\_SYST\_CATPRINTF{ID-WIDTH}\_{PARAMETER-COUNT} macros call their corresponding catalog API macro by dropping the format parameter. The API is intended to have printf style instrumentation inside the sources, but drop the format string during compilation and replace it with a catalog API call instead. Source scanning tools are used to extract the format strings from sources into catalog dictionary files. The dictionary files can then be used to reconstruct the printf formatting during trace decode time. This saves both space and execution time in the resulting application, and bandwidth over a trace link.

This API set is enabled or disabled by the #MIPI\_SYST\_PCFG\_ENABLE\_CATID64\_API and/or #MIPI\_SYST\_PCFG\_ENABLE\_CATID32\_API platform feature defines.

### Macro Definition Documentation

#### #define MIPI\_SYST\_CATPRINTF32\_0( svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0(svh, sev, id)

#### #define MIPI\_SYST\_CATPRINTF32\_0\_LOC16( svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC16(svh, sev, f, id)

#### #define MIPI\_SYST\_CATPRINTF32\_0\_LOC32( svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC32(svh, sev, f, id)

#### #define MIPI\_SYST\_CATPRINTF32\_0\_LOCADDR( svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOCADDR(svh, sev, id)

#### #define MIPI\_SYST\_CATPRINTF32\_1( svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1(svh, sev, id, p1)

#### #define MIPI\_SYST\_CATPRINTF32\_1\_LOC16( svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC16(svh, sev, f, id, p1)

#### #define MIPI\_SYST\_CATPRINTF32\_1\_LOC32( svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC32(svh, sev, f, id, p1)

#### #define MIPI\_SYST\_CATPRINTF32\_1\_LOCADDR( svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOCADDR(svh, sev, id, p1)

#### #define MIPI\_SYST\_CATPRINTF32\_2( svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2(svh, sev, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF32\_2\_LOC16( svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC16(svh, sev, f, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF32\_2\_LOC32( svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC32(svh, sev, f, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF32\_2\_LOCADDR( svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOCADDR(svh, sev, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF32\_3( svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3(svh, sev, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF32\_3\_LOC16( svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC16(svh, sev, f, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF32\_3\_LOC32( svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC32(svh, sev, f, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF32\_3\_LOCADDR( svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOCADDR(svh, sev, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF32\_4( svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4(svh, sev, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF32\_4\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF32\_4\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF32\_4\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF32\_5( svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5(svh, sev, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF32\_5\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF32\_5\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF32\_5\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF32\_6( svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF32\_6\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF32\_6\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF32\_6\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF64\_0( svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0(svh, sev, id)

#### #define MIPI\_SYST\_CATPRINTF64\_0\_LOC16( svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC16(svh, sev, f, id)

#### #define MIPI\_SYST\_CATPRINTF64\_0\_LOC32( svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC32(svh, sev, f, id)

#### #define MIPI\_SYST\_CATPRINTF64\_0\_LOCADDR( svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOCADDR(svh, sev, id)

#### #define MIPI\_SYST\_CATPRINTF64\_1( svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1(svh, sev, id, p1)

#### #define MIPI\_SYST\_CATPRINTF64\_1\_LOC16( svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC16(svh, sev, f, id, p1)

#### #define MIPI\_SYST\_CATPRINTF64\_1\_LOC32( svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC32(svh, sev, f, id, p1)

#### #define MIPI\_SYST\_CATPRINTF64\_1\_LOCADDR( svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOCADDR(svh, sev, id, p1)

#### #define MIPI\_SYST\_CATPRINTF64\_2( svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2(svh, sev, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF64\_2\_LOC16( svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC16(svh, sev, f, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF64\_2\_LOC32( svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC32(svh, sev, f, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF64\_2\_LOCADDR( svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOCADDR(svh, sev, id, p1, p2)

#### #define MIPI\_SYST\_CATPRINTF64\_3( svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3(svh, sev, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF64\_3\_LOC16( svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC16(svh, sev, f, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF64\_3\_LOC32( svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC32(svh, sev, f, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF64\_3\_LOCADDR( svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOCADDR(svh, sev, id, p1, p2, p3)

#### #define MIPI\_SYST\_CATPRINTF64\_4( svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4(svh, sev, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF64\_4\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF64\_4\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF64\_4\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)

#### #define MIPI\_SYST\_CATPRINTF64\_5( svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5(svh, sev, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF64\_5\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF64\_5\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF64\_5\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)

#### #define MIPI\_SYST\_CATPRINTF64\_6( svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF64\_6\_LOC16( svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF64\_6\_LOC32( svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

#### #define MIPI\_SYST\_CATPRINTF64\_6\_LOCADDR( svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)

# Data Structure Documentation

## mipi\_syst\_platform\_handle Struct Reference

#include <platform.h>

### Data Fields

* **mipi\_syst\_u32** **sph\_io\_count**
* **mipi\_syst\_u32** **sph\_raw\_count**
* **mipi\_syst\_u8** **sph\_raw** [1024]

### Detailed Description

Platform specific SyS-T handle state extension

The contents of this structure can be freely defined to match platform specific data needs. It can later be accessed through the syst\_handles systh\_platform member.

##### See also:

MIPI\_SYST\_PCFG\_ENABLE\_PLATFORM\_HANDLE\_DATA struct mipi\_syst\_handle

### Field Documentation

#### mipi\_syst\_u32 sph\_io\_count

cnt io's, used for pretty printing

#### mipi\_syst\_u8 sph\_raw[1024]

buffer for printing

#### mipi\_syst\_u32 sph\_raw\_count

number of raw bytes in sph\_raw

#### The documentation for this struct was generated from the following file:

* **platform.h**

## mipi\_syst\_platform\_state Struct Reference

#include <platform.h>

### Data Fields

* void(\* **write\_d32ts** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u32** v)
* void(\* **write\_d32mts** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u32** v)
* void(\* **write\_d64mts** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u64** v)
* void(\* **write\_d8** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u8** v)
* void(\* **write\_d16** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u16** v)
* void(\* **write\_d32** )(struct mipi\_syst\_handle \*systh, **mipi\_syst\_u32** v)
* void(\* **write\_flag** )(struct mipi\_syst\_handle \*systh)
* void \* **sph\_init\_data**

### Detailed Description

Platform specific SyS-T global state extension

The contents of this structure can be freely defined to match platform specific data needs. It can later be accessed through the mipi\_syst\_header systh\_platform member.

This platform example puts low-level output function pointers here. Real implementations may have them "inlined" for performance reasons.

### Field Documentation

#### void\* sph\_init\_data

#### void(\* write\_d16) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u16 v)

#### void(\* write\_d32) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u32 v)

#### void(\* write\_d32mts) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u32 v)

#### void(\* write\_d32ts) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u32 v)

#### void(\* write\_d64mts) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u64 v)

#### void(\* write\_d8) (struct mipi\_syst\_handle \*systh, mipi\_syst\_u8 v)

#### void(\* write\_flag) (struct mipi\_syst\_handle \*systh)

#### The documentation for this struct was generated from the following file:

* **platform.h**

# File Documentation

#include "mipi\_syst.h"

### Macros

* #define **MIPI\_SYST\_INIT\_STATE**(s, f, p)  **mipi\_syst\_init**((s), (f), (p))
* #define **MIPI\_SYST\_SHUTDOWN\_STATE**(s, f)  **mipi\_syst\_destroy**((s), (f))
* #define **MIPI\_SYST\_INIT**(f, p)  **MIPI\_SYST\_INIT\_STATE**((struct mipi\_syst\_header\*)0, (f), (p))
* #define **MIPI\_SYST\_SHUTDOWN**(f)  **MIPI\_SYST\_SHUTDOWN\_STATE**((struct mipi\_syst\_header\*)0, (f))
* #define **MIPI\_SYST\_INIT\_HANDLE\_STATE**(s, h, p)  **mipi\_syst\_init\_handle**((s), (h), (p), 0)
* #define **MIPI\_SYST\_INIT\_HANDLE**(h, p)  **MIPI\_SYST\_INIT\_HANDLE\_STATE**((struct mipi\_syst\_header\*)0, (h), (p))
* #define **MIPI\_SYST\_ALLOC\_HANDLE**(p)  CFG\_ERROR\_ALLOC\_HANDLE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY
* #define **MIPI\_SYST\_ALLOC\_HANDLE\_STATE**(s, p)  CFG\_ERROR\_SYST\_ALLOC\_HANDLE\_STATE\_CALLED\_WITHOUT\_PCFG\_ENABLE\_HEAP\_MEMORY
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_LENGTH**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_LENGTH**(h)  (((h) &&(h)->systh\_tag.et\_length) ? 1 : 0)
* #define **MIPI\_SYST\_SET\_HANDLE\_MODULE\_UNIT**(h, module, unit)
* #define **MIPI\_SYST\_SET\_HANDLE\_GUID\_UNIT**(p, g, u)  CFG\_ERROR\_SET\_HANDLE\_GUID\_UNIT\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_ORIGIN\_GUID
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP**(h, v)  CFG\_ERROR\_MIPI\_SYST\_ENABLE\_HANDLE\_TIMESTAMP\_WITHOUT\_MIPI\_SYST\_PCFG\_ENABLE\_TIMESTAMP
* #define **MIPI\_SYST\_DELETE\_HANDLE**(h)  **mipi\_syst\_delete\_handle**(h)
* #define **MIPI\_SYST\_NOLOCATION**  (struct mipi\_syst\_msglocation \*)0
* #define **MIPI\_SYST\_SHORT32**(h, v)
* #define **MIPI\_SYST\_SHORT64**(h, v)
* #define **MIPI\_SYST\_BUILD\_COMPACT32**(h, n)
* #define **MIPI\_SYST\_BUILD\_COMPACT64**(h, n)
* #define **\_MIPI\_SYST\_MK\_PARAM\_LIST**(tag, p)  \_MIPI\_SYST\_CATARG\_##tag, p
* #define **MIPI\_SYST\_PARAM\_INT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(D, (p))
* #define **MIPI\_SYST\_PARAM\_LONG**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LD, (p))
* #define **MIPI\_SYST\_PARAM\_LONGLONG**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LLD, (p))
* #define **MIPI\_SYST\_PARAM\_SIZE\_T**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(ZD, (p))
* #define **MIPI\_SYST\_PARAM\_PTRDIFF\_T**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(TD, (p))
* #define **MIPI\_SYST\_PARAM\_FLOAT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(F, (p))
* #define **MIPI\_SYST\_PARAM\_DOUBLE**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(F, (p))
* #define **MIPI\_SYST\_PARAM\_LONGDOUBLE**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LF, (p))
* #define **MIPI\_SYST\_PARAM\_CHAR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(C, (p))
* #define **MIPI\_SYST\_PARAM\_SHORT**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(HHD, (p))
* #define **MIPI\_SYST\_PARAM\_WCHAR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(LC, (p))
* #define **MIPI\_SYST\_PARAM\_PTR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(P, (p))
* #define **MIPI\_SYST\_PARAM\_CSTR**(p)  **\_MIPI\_SYST\_MK\_PARAM\_LIST**(CSTR, (p))
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_CHECKSUM**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_CHECKSUM**(h)  0
* #define **MIPI\_SYST\_ENABLE\_HANDLE\_COUNTER**(h, v)
* #define **MIPI\_SYST\_GET\_HANDLE\_COUNTER**(h)  0
* #define **MIPI\_SYST\_GET\_HANDLE\_TIMESTAMP**(h)  0
* #define **MIPI\_SYST\_WRITE**(h, sev, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOCADDR**(h, sev, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOC16**(h, sev, f, id, p, len)
* #define **MIPI\_SYST\_WRITE\_LOC32**(h, sev, f, id, p, len)
* #define **MIPI\_SYST\_DEBUG**(svh, sev, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOCADDR**(svh, sev, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOC16**(svh, sev, file, str, len)
* #define **MIPI\_SYST\_DEBUG\_LOC32**(svh, sev, file, str, len)
* #define **MIPI\_SYST\_FUNC\_ENTER**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOCADDR**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOC16**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_ENTER\_LOC32**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_EXIT**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOCADDR**(svh, sev)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOC16**(svh, sev, file)
* #define **MIPI\_SYST\_FUNC\_EXIT\_LOC32**(svh, sev, file)
* #define **MIPI\_SYST\_ASSERT**(svh, sev, cond)
* #define **MIPI\_SYST\_ASSERT\_LOC16**(svh, sev, file, cond)
* #define **MIPI\_SYST\_ASSERT\_LOC32**(svh, sev, file, cond)
* #define **MIPI\_SYST\_ASSERT\_LOCADDR**(svh, sev, cond)
* #define **MIPI\_SYST\_CLOCK\_SYNC**(h, c, f)
* #define **MIPI\_SYST\_PRINTF**(...)
* #define **MIPI\_SYST\_PRINTF\_LOC16**(...)
* #define **MIPI\_SYST\_PRINTF\_LOC32**(...)
* #define **MIPI\_SYST\_PRINTF\_LOCADDR**(...)
* #define **MIPI\_SYST\_CATPRINTF64**(...)
* #define **MIPI\_SYST\_CATPRINTF64\_LOC16**(...)
* #define **MMIPI\_SYST\_CATPRINTF64\_LOC32**(...)
* #define **MIPI\_SYST\_CATPRINTF64\_LOCADDR**(...)
* #define **MIPI\_SYST\_CATPRINTF32**(...)
* #define **MIPI\_SYST\_CATPRINTF32\_LOC16**(...)
* #define **MMIPI\_SYST\_CATPRINTF32\_LOC32**(...)
* #define **MIPI\_SYST\_CATPRINTF32\_LOCADDR**(...)
* #define **MIPI\_SYST\_CATPRINTF32\_0**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOCADDR**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOCADDR(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOCADDR**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOCADDR(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOCADDR**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOCADDR(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOCADDR(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOC16**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC16(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOC16**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC16(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOC16**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC16(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC16(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF32\_0\_LOC32**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG32\_0\_LOC32(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF32\_1\_LOC32**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG32\_1\_LOC32(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF32\_2\_LOC32**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG32\_2\_LOC32(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF32\_3\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG32\_3\_LOC32(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF32\_4\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG32\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF32\_5\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG32\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF32\_6\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG32\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOCADDR**(svh, sev, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOCADDR(svh, sev, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOCADDR**(svh, sev, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOCADDR(svh, sev, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOCADDR**(svh, sev, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOCADDR(svh, sev, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOCADDR(svh, sev, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOCADDR(svh, sev, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOCADDR**(svh, sev, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOCADDR(svh, sev, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOC16**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC16(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOC16**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC16(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOC16**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC16(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC16(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC16(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOC16**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC16(svh, sev, f, id, p1, p2, p3, p4, p5, p6)
* #define **MIPI\_SYST\_CATPRINTF64\_0\_LOC32**(svh, sev, f, id, fmt)  MIPI\_SYST\_CATALOG64\_0\_LOC32(svh, sev, f, id)
* #define **MIPI\_SYST\_CATPRINTF64\_1\_LOC32**(svh, sev, f, id, fmt, p1)  MIPI\_SYST\_CATALOG64\_1\_LOC32(svh, sev, f, id, p1)
* #define **MIPI\_SYST\_CATPRINTF64\_2\_LOC32**(svh, sev, f, id, fmt, p1, p2)  MIPI\_SYST\_CATALOG64\_2\_LOC32(svh, sev, f, id, p1, p2)
* #define **MIPI\_SYST\_CATPRINTF64\_3\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3)  MIPI\_SYST\_CATALOG64\_3\_LOC32(svh, sev, f, id, p1, p2, p3)
* #define **MIPI\_SYST\_CATPRINTF64\_4\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4)  MIPI\_SYST\_CATALOG64\_4\_LOC32(svh, sev, f, id, p1, p2, p3, p4)
* #define **MIPI\_SYST\_CATPRINTF64\_5\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5)  MIPI\_SYST\_CATALOG64\_5\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5)
* #define **MIPI\_SYST\_CATPRINTF64\_6\_LOC32**(svh, sev, f, id, fmt, p1, p2, p3, p4, p5, p6)  MIPI\_SYST\_CATALOG64\_6\_LOC32(svh, sev, f, id, p1, p2, p3, p4, p5, p6)

### Enumerations

* enum **mipi\_syst\_catalog\_parameter\_types** { **\_MIPI\_SYST\_CATARG\_END** = 0, **\_MIPI\_SYST\_CATARG\_D** = 1, **\_MIPI\_SYST\_CATARG\_LD** = 2, **\_MIPI\_SYST\_CATARG\_LLD** = 3, **\_MIPI\_SYST\_CATARG\_ZD** = 4, **\_MIPI\_SYST\_CATARG\_TD** = 5, **\_MIPI\_SYST\_CATARG\_F** = 6, **\_MIPI\_SYST\_CATARG\_LF** = 7, **\_MIPI\_SYST\_CATARG\_C** = 8, **\_MIPI\_SYST\_CATARG\_HHD** = 9, **\_MIPI\_SYST\_CATARG\_LC** = 10, **\_MIPI\_SYST\_CATARG\_P** = 11, **\_MIPI\_SYST\_CATARG\_CSTR** = 12 }

### Functions

* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_init** (struct mipi\_syst\_header \*header, mipi\_syst\_inithook\_t pfinit, const void \*init\_data)
* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_destroy** (struct mipi\_syst\_header \*header, mipi\_syst\_destroyhook\_t pfdestroy)
* **MIPI\_SYST\_EXPORT** struct mipi\_syst\_handle \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_init\_handle** (struct mipi\_syst\_header \*header, struct mipi\_syst\_handle \*svh, const struct mipi\_syst\_origin \*origin, **mipi\_syst\_u32** heap)
* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_delete\_handle** (struct mipi\_syst\_handle \*svh)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_file\_location32** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u16** f, **mipi\_syst\_u16** l)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_file\_location64** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** f, **mipi\_syst\_u32** l)
* MIPI\_SYST\_INLINE struct mipi\_syst\_msglocation \***MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_address\_location** (struct mipi\_syst\_handle \*h, void \*p)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param0** (struct mipi\_syst\_handle \*h)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param1** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param2** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param3** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param4** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param5** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5)
* MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param6** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5, **mipi\_syst\_u32** p6)

### Macro Definition Documentation

#### #define MIPI\_SYST\_NOLOCATION  (struct mipi\_syst\_msglocation \*)0

Pass null to instrumentation API to skip location information generation.

## compiler.h File Reference

#include <stdint.h>

### Macros

* #define **MIPI\_SYST\_EXPORT**
* #define **MIPI\_SYST\_CALLCONV**
* #define **MIPI\_SYST\_FUNCTION\_NAME**  \_\_PRETTY\_FUNCTION\_\_
* #define **MIPI\_SYST\_LINE**  \_\_LINE\_\_
* #define **MIPI\_SYST\_FILE**  \_\_FILE\_\_
* #define **MIPI\_SYST\_CC\_INLINE**  inline
* #define **MIPI\_SYST\_HTOLE16**(v)  (v)
* #define **MIPI\_SYST\_HTOLE32**(v)  (v)
* #define **MIPI\_SYST\_HTOLE64**(v)  (v)

### Typedefs

* typedef char **mipi\_syst\_s8**
* typedef int16\_t **mipi\_syst\_s16**
* typedef int **mipi\_syst\_s32**
* typedef long long **mipi\_syst\_s64**
* typedef uint8\_t **mipi\_syst\_u8**
* typedef uint16\_t **mipi\_syst\_u16**
* typedef uint32\_t **mipi\_syst\_u32**
* typedef uint64\_t **mipi\_syst\_u64**

### Macro Definition Documentation

#### #define MIPI\_SYST\_CALLCONV

#### #define MIPI\_SYST\_CC\_INLINE  inline

#### #define MIPI\_SYST\_EXPORT

#### #define MIPI\_SYST\_FILE  \_\_FILE\_\_

#### #define MIPI\_SYST\_FUNCTION\_NAME  \_\_PRETTY\_FUNCTION\_\_

#### #define MIPI\_SYST\_HTOLE16( v)  (v)

#### #define MIPI\_SYST\_HTOLE32( v)  (v)

#### #define MIPI\_SYST\_HTOLE64( v)  (v)

#### #define MIPI\_SYST\_LINE  \_\_LINE\_\_

### Typedef Documentation

#### typedef int16\_t mipi\_syst\_s16

#### typedef int mipi\_syst\_s32

#### typedef long long mipi\_syst\_s64

#### typedef char mipi\_syst\_s8

#### typedef uint16\_t mipi\_syst\_u16

#### typedef uint32\_t mipi\_syst\_u32

#### typedef uint64\_t mipi\_syst\_u64

#### typedef uint8\_t mipi\_syst\_u8

#### 

## inline.h File Reference

### Functions

1. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param0** (struct mipi\_syst\_handle \*h)
2. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param1** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1)
3. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param2** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2)
4. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param3** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3)
5. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param4** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4)
6. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param5** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5)
7. MIPI\_SYST\_INLINE void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_make\_param6** (struct mipi\_syst\_handle \*h, **mipi\_syst\_u32** p1, **mipi\_syst\_u32** p2, **mipi\_syst\_u32** p3, **mipi\_syst\_u32** p4, **mipi\_syst\_u32** p5, **mipi\_syst\_u32** p6)
8. MIPI\_SYST\_INLINE **mipi\_syst\_u32** **mipi\_syst\_hash\_x65599** (const char \*p, **mipi\_syst\_u32** length)

### Function Documentation

#### MIPI\_SYST\_INLINE mipi\_syst\_u32 mipi\_syst\_hash\_x65599 (const char \* *p*, mipi\_syst\_u32 *length*)

Runtime computation of hash values for catalog message IDs. This function is used in debug builds to avoid a code explosion by the preprocessor method (

##### See also:

**mipi\_syst\_hash\_x65599** ). The preprocessor methods does compute this value at compile time only if const expression optimizations are enabled.

## platform.h File Reference

### Data Structures

* struct **mipi\_syst\_platform\_state**
* struct **mipi\_syst\_platform\_handle**

### Macros

* #define **MIPI\_SYST\_OUTPUT\_D32TS**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_D32MTS**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_D64MTS**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_D8**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_D16**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_D32**(syst\_handle, data)
* #define **MIPI\_SYST\_OUTPUT\_FLAG**(syst\_handle)

### Functions

* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_platform\_init** (struct mipi\_syst\_header \*, const void \*)
* **MIPI\_SYST\_EXPORT** void **MIPI\_SYST\_CALLCONV** **mipi\_syst\_platform\_destroy** (struct mipi\_syst\_header \*systh)

### Macro Definition Documentation

#### #define MIPI\_SYST\_OUTPUT\_D16( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_D32( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_D32MTS( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_D32TS( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_D64MTS( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_D8( syst\_handle, data)

#### #define MIPI\_SYST\_OUTPUT\_FLAG( syst\_handle)

### Function Documentation

#### MIPI\_SYST\_EXPORT void MIPI\_SYST\_CALLCONV mipi\_syst\_platform\_destroy (struct mipi\_syst\_header \* *systh*)

#### MIPI\_SYST\_EXPORT void MIPI\_SYST\_CALLCONV mipi\_syst\_platform\_init (struct mipi\_syst\_header \* , const void \* )