# GRAPHICS DISPLAY CONTROLLER

MB88F332 'INDIGO'

# APIX® AUTOMOTIVE SHELL SW-EMULATION USE CASE

APPLICATION NOTE

# Revision History

| Date | Issue |
|---|---|
| 17/11/08 | Rev. 0.10, V. Thormann<br>First draft |
| 18/11/08 | Rev. 0.20 v. Treuberg, minor typo corrections |
| 01/12/08 | Rev. 0.21 v. Treuberg, modified 8, INAP125T24 Device Configuration, table entry for 03 hex (upstream link data recovery). |
| 27/04/09 | Rev. 0.22 V, Thormann, fixed typo (APIX RX Config_byte 7, bit 4 changed to 1, otherwise APIX-RX is set to test mode) |

This document contains 26 pages.

# Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (eg. software include or header files, application examples, target boards, evaluation boards, engineering samples of IC's etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. **Note, all these products are intended and must only be used in an evaluation laboratory environment**.

1. Fujitsu Microelectronics Europe GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days form the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Microelectronics Europe GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.

2. Should a Product turn out to be defect, Fujitsu Microelectronics Europe GmbH´s entire liability and the customer´s exclusive remedy shall be, at Fujitsu Microelectronics Europe GmbH´s sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Microelectronics Europe GmbH in original packing and without further defects resulting from the customer´s use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.

4. To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH´s and its suppliers´ liability is restricted to intention and gross negligence.

   **NO LIABILITY FOR CONSEQUENTIAL DAMAGES**

   **To the maximum extent permitted by applicable law, in no event shall Fujitsu Microelectronics Europe GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.**

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect

# Contents

# 1 Preface

We thank Inova Semiconductors GmbH for their valuable contribution to the content of this document.

# 2 Introduction

This document describes an example system which is characterized by the following:

- Video downstream transmission via the APIX pixel link
- Use of an INAP125T24 IC APIX transmitter
- An APIX receiver is integrated in the MB88F332 'Indigo' GDC
- The APIX Automotive Shell (referred to as the 'Ashell' in this document) is used for transmission of configuration data, control data (downlink direction) and status data (uplink direction) of Indigo IC
- On the transmitter side, the APIX Automotive Shell is operated in “compatibility mode” to allow the use of the standalone INAP125T24 APIX PHY device.
- On the transmitter side, the APIX Automotive Shell is implemented in software running on a standard MCU using two SPI peripherals
- The APIX Automotive Shell (on both receiver and transmitter side) is additionally operated in a combined “integrity-only” and “payload-only” mode to simplify implementation in software.
- On the receiver side, the APIX Automotive Shell is integrated in the Indigo GDC

The following description refers to the use case outlined above as the “Ashell SW-emulation use case” (please note that only the transmitter Ashell is emulated in SW, the receiver Ashell is implemented in Indigo HW). This document concentrates on how to establish communication via the APIX sideband interface. The transmission of video data is not the primary subject of this document. Please also refer to documents [1] and [2] in section 10 'References'.

# 3 Layered Communication Architecture

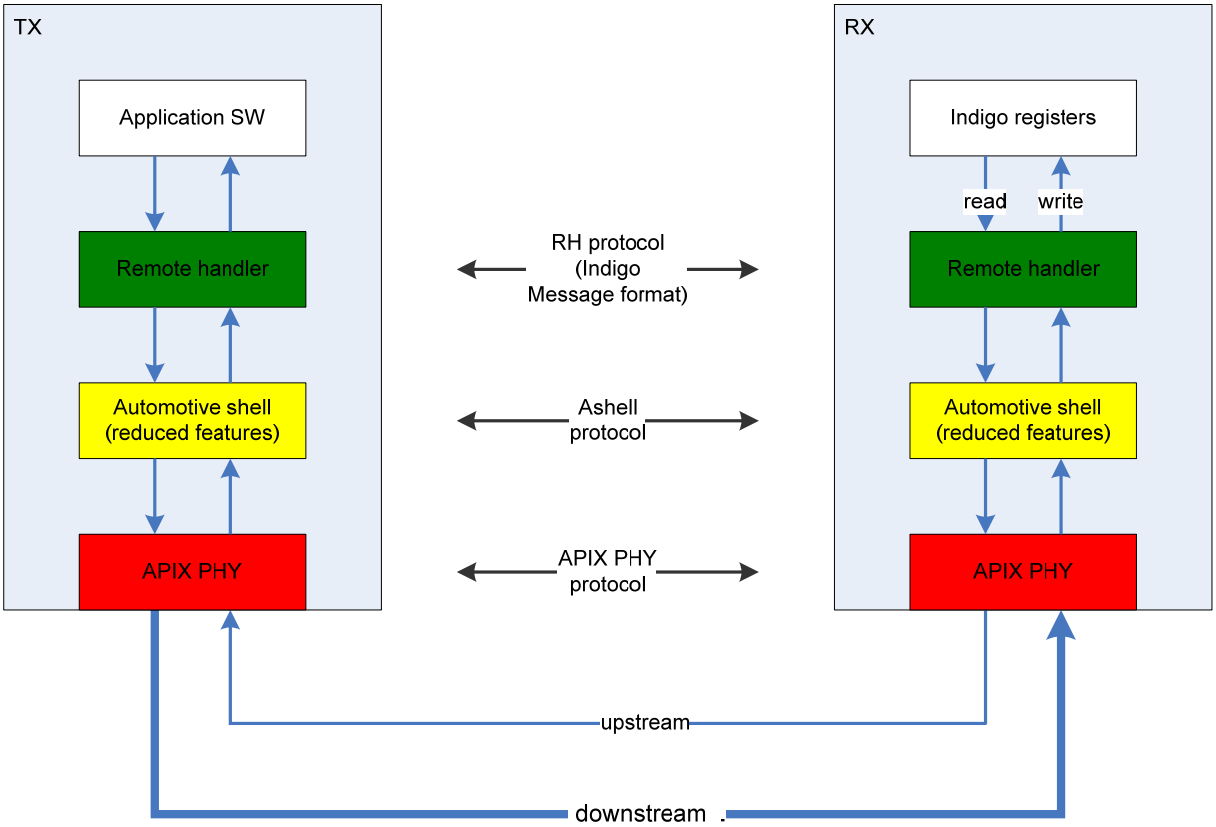


**Figure 3-1, APIX layered communication system**

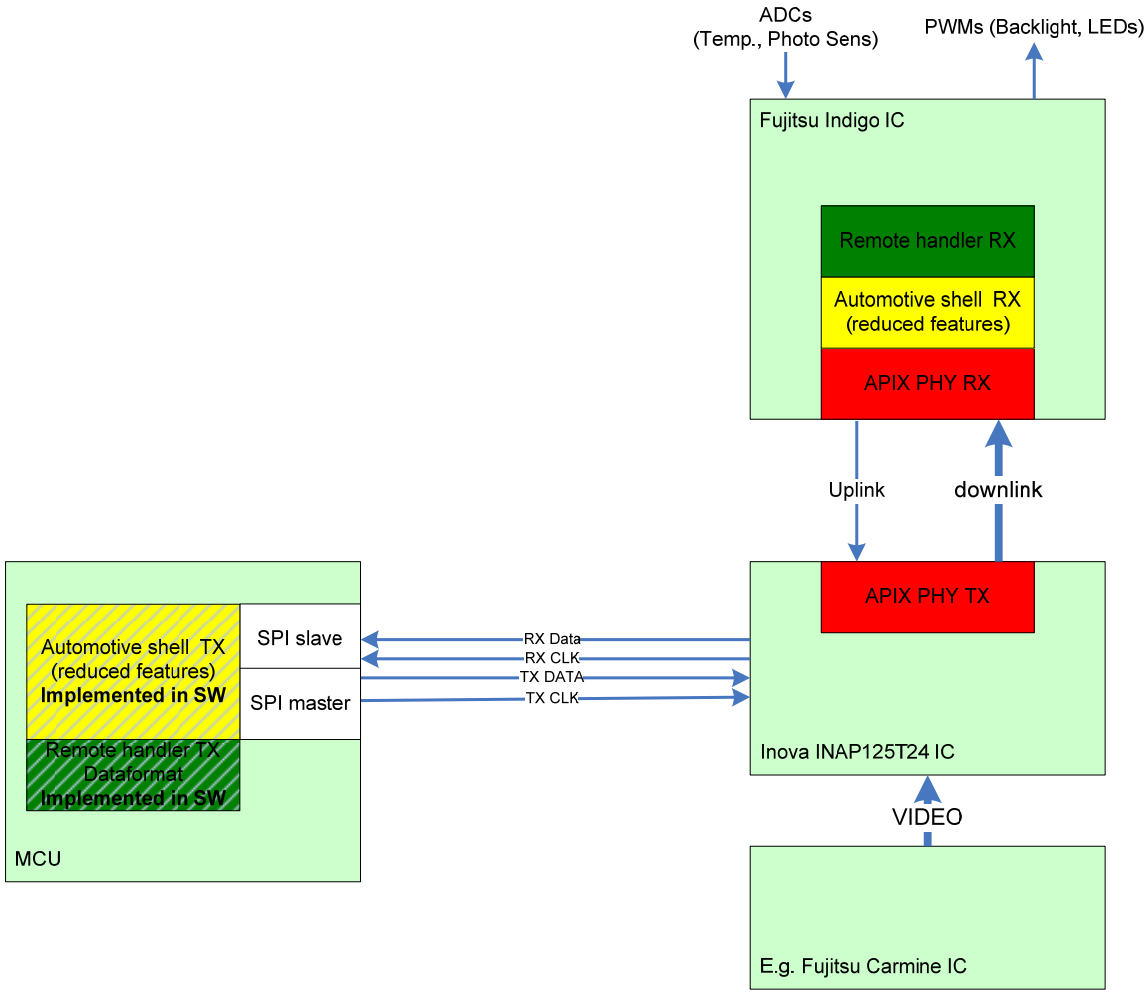


**Figure 3-2, Mapping communication layers to hardware devices and software**

# 4 Wiring

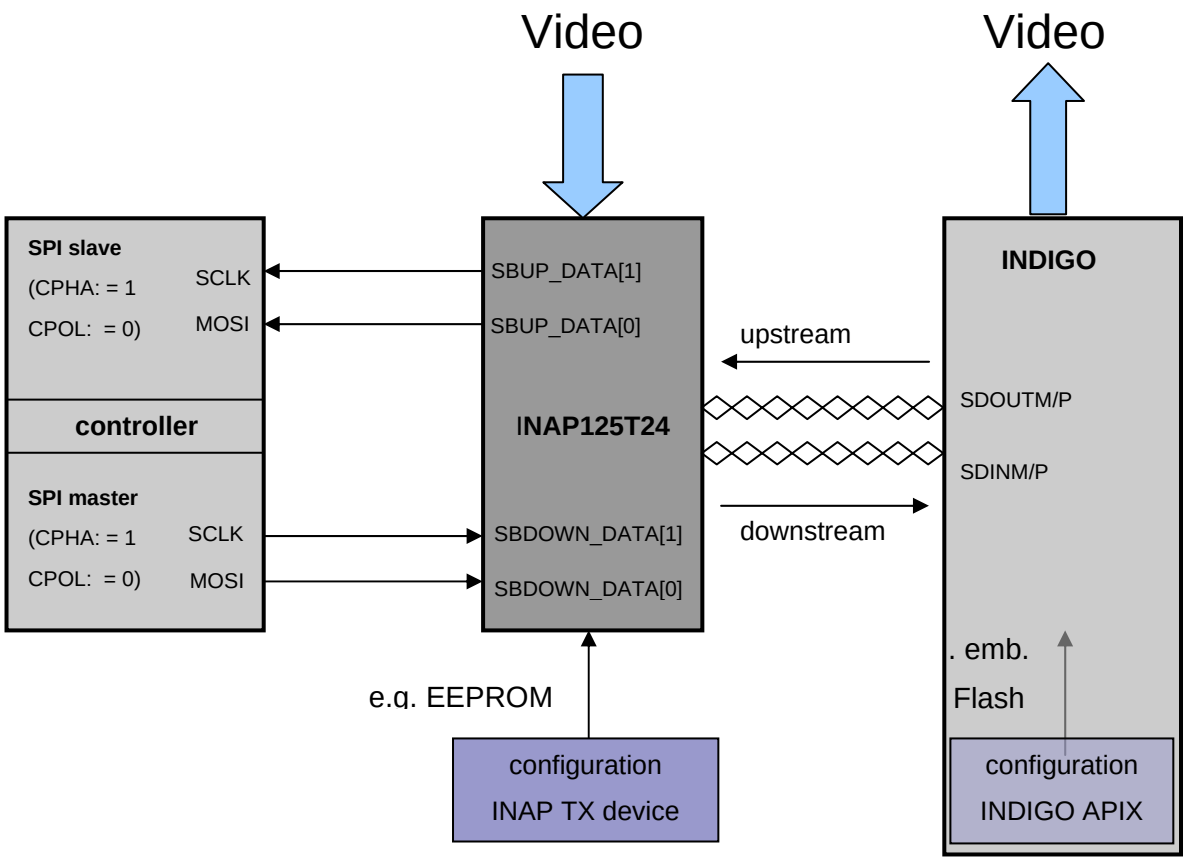


**Figure 4-1, Wiring of for APIX communication relevant pins**

# 5 Automotive Shell Layer

To allow interfacing to the standalone TX PHY device INAP125T, the TX Ashell operates in the so called “compatibility mode”.

To simplify implementation and to adapt to a use case’s needs, the Ashell supports both a so called “integrity only mode” and a “payload only mode”. If both Ashells on transmitter and receiver side are operated in these two modes, a cost effective implementation in software is possible. For the use case described in this document SW implementation is choosen for the transmitter side Ashell.

For details please refer to documents [3] and [4], which can be ordered from Inova Semiconductors GmbH.

The Ashell uses an SPI-like protocol for the sideband interface to allow the use of standard SPI peripherals.

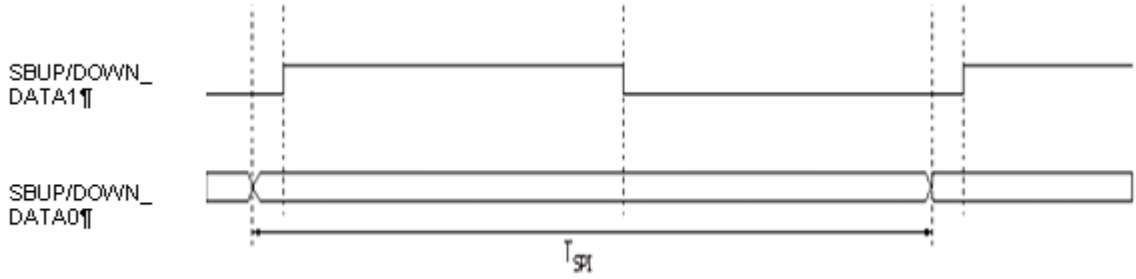


**Figure 5-1 SPI-like protocol timing**

The maximum SPI downlink speed is determined by the INAP125T24 sideband sample rate, see document [5], chapter “Sideband Channel Downstream Interface”.

The uplink SPI speed is configured by the parameter cfg_sbup_daclk_clength.

To establish communication between the Ashell TX and RX, it is important that several Ashell and Apix PHY parameters on both the RX and TX sides match. Notes concerning these parameters can be found in chapters 7 and 8.

# 6 Remote Handler Communication Layer

The Indigo remote handler provides write and read access to Indigo's register space. For this purpose write request, read request and read result messages are exchanged between TX and RX side. The message format which is transmitted as payload by the lower layers (Ashell and PHY) is defined in document [1], remote handler chapter "Data Formats".

Read and write requests can be sent in any order. Read requests are answered by Indigo within an undeterministic time period by a read result message. Therefore not more than 16 read requests should be sent before receiving the associated answers. Please provide a "read index" number to each read request to be able to assign the correct read result to the relevant request.

Additional services such as message-based interrupts can be disabled to avoid the need for decoding such message types and so reducing the processor load. For this purpose, set the remote handler register CommonControl.MSIenable = 0.

# 7 INDIGO configuration

The configuration software interface for the APIX® PHY and APIX® Ashell® is available via the registers of the remote handler configuration interface (base address 0x001_1000). Please refer also to document [1], Remote Handler chapter “Registers”. Registers PHYConfig0/1 and ShellConfig0 contain all the configuration bytes of the APIX PHY and APIX Automotive Shell.

These registers can be programmed after a chip reset by the Indigo Command Sequencer bootup sequence. **Please note that it is not possible to reprogramm these configuration registers during active APIX operation.** The APIX® PHY and APIX® Ashell® must be put into reset state during reprogramming. Use the following flow for this purpose:

set APConfig.rapix_config = 1

set PhyConfig0 = xx

set PhyConfig1 = xx

set ShellConfig0 = xx

set APConfig.rapix_config = 0

This configuration sequence can be stored in Indigo’s embedded flash memory. Indigo’s Command sequencer executes this sequence at startup.

The following tables list recommended values for all parameters. The descriptions are limited to parameters which are relevant for the “Ashell SW-emulation use case”. For a full description of parameters, please refer to [1]. Some application specific parameters are marked with “Appl.” and must be adapted to the requirements of the characteristics of the relevant application.

| config_byte_1 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | Appl | cfg_up_clk_divider[1] | APIX PHY<br>upstream channel bandwidth setting<br><br>bandwidth mode of downstream link<br>1000 Mbit/s 125 MBit/s<br>500 Mbit/s<br><br>00: not applicable 62.50 MBit/s<br>01: 62.50 MBit/s 31.25 MBit/s<br>10: 41.67 MBit/s 20.83 MBit/s<br>11: 31.25 MBit/s not applicable<br><br>**Note**: upstream bandwidth setting has to match related transmitter device configuration |
| 6 | Appl | cfg_up_clk_divider[0] | |
| 5 | 0 | cfg_upDataSwing[0] | APIX PHY<br>upstream serial output current swing<br><br>(binary coded, 1 LSB = 0.65mA)<br>00000: min 0mA<br>00001: 0.6mA<br>00010: 1,3mA<br>...<br>11111: max 20mA<br><br>**Note:** please adapt to used cable length... |
| 4 | 0 | cfg_upDataSwing[1] | |
| 3 | 0 | cfg_upDataSwing[2] | |
| 2 | 0 | cfg_upDataSwing[3] | |
| 1 | 1 | cfg_upDataSwing[4] | |
| 0 | 1 | cfg_sbup_smode | APIX PHY<br>mandatory setting for “Ashell SW-emulation use case” |

**Table 7-1, config_byte_1**

| config_byte_2 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | Appl | cfg_pxdata_width[1] | APIX PHY<br>bit width of pixel data<br><br>00: reserved<br>01: reserved<br>10: 18 bits<br>11: 24 bits<br><br>**Note:** width of pixel data setting has to match related transmitter device configuration |
| 6 | Appl | cfg_pxdata_width[0] | |
| 5 | Appl | cfg_px_out_ctrl_piggyback[1] | APIX PHY<br>transmission of pixel control signals (px_ctrl[2:0])<br><br>00: never<br>01: unused<br>10: with even pixels only<br>11: with every pixel<br><br>**Note:** pixel control signals setting has to match related transmitter device configuration<br><br>**Note:** to achieve maximum pixel link net bandwidth setting “10” is necessary, see document [5] chapter 1.2 |
| 4 | Appl | cfg_px_out_ctrl_piggyback[0] | |
| 3 | 1 | Reserved | do not change |
| 2 | 1 | Reserved | do not change |
| 1 | 0 | Reserved | do not change |
| 0 | 0 | Reserved | do not change |

**Table 7-2, config_byte_2**

| config_byte_3 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 1 | reserved | do not change |
| 6 | 0 | reserved | do not change |
| 5 | 0 | reserved | do not change |
| 4 | 0 | reserved | |
| 3 | 0 | reserved | |
| 2 | 1 | reserved | do not change |
| 1 | 1 | reserved | do not change |
| 0 | 0 | reserved | do not change |

**Table 7-3, config_byte_3**

| config_byte_4 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 0 | reserved | do not change |
| 6 | 0 | reserved | do not change |
| 5 | 0 | reserved | do not change |
| 4 | 0 | reserved | do not change |
| 3 | 0 | reserved | do not change |
| 2 | 0 | reserved | do not change |
| 1 | 0 | reserved | do not change |
| 0 | 0 | reserved | do not change |

**Table 7-4, config_byte_4**

| config_byte_5 | | | |
|---|---|---|---|
| Bit: | Value: | Name: | Description: |
| 7 | 1 | reserved | Do not change |
| 6 | 1 | reserved | |
| 5 | 1 | reserved | |
| 4 | 0 | reserved | |
| 3 | 0 | reserved | do not change |
| 2 | 0 | reserved | do not change |
| 1 | 0 | reserved | do not change |
| 0 | 1 | reserved | do not change |

**Table 7-5, config_byte_5**

| config_byte_6 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | Appl | cfg_downBwMode[1] | APIX PHY<br>selects downstream bandwidth mode<br><br>11: 1000 MBit/s (Full Bandwidth Mode)<br>10: 500 MBit/s (Half Bandwidth Mode)<br>00: 125 MBit/s (Low Bandwidth Mode 1)<br>01: not applicable<br><br>**Note**: downstream bandwidth setting has to match related transmitter device configuration |
| 6 | Appl | cfg_downBwMode[0] | |
| 5 | 0 | cfg_ddown_enable | APIX PHY / Ashell<br><br>mandatory setting for “Ashell SW-emulation use case” |
| 4 | 0 | reserved | do not change |
| 3 | 0 | reserved | do not change |
| 2 | 0 | reserved | do not change |
| 1 | 0 | reserved | do not change |
| 0 | 0 | reserved | do not change |

**Table 7-6, config_byte_6**

| config_byte_7 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 1 | reserved | do not change |
| 6 | 0 | reserved | |
| 5 | 0 | reserved | |
| 4 | 1 | reserved | |
| 3 | 0 | reserved | do not change |
| 2 | 0 | reserved | do not change |
| 1 | 1 | reserved | do not change |
| 0 | 1 | reserved | do not change |

**Table 7-7, config_byte_7**

| config_byte_shell1 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 0 | reserved | Do not change |
| 6 | Appl. | cfg_sbup_daclk_clength[10] | AShell<br>configures data rate of upstream sideband (see tables below)<br><br>**Note:** valid if cfg_sbup_daclk[1:0] = “10” |
| 5 | Appl. | cfg_sbup_daclk_clength[9] | |
| 4 | Appl. | cfg_sbup_daclk_clength[8] | |
| 3 | Appl. | cfg_sbup_daclk_clength[7] | |
| 2 | Appl. | cfg_sbup_daclk_clength[6] | |
| 1 | Appl. | cfg_sbup_daclk_clength[5] | |
| 0 | Appl. | cfg_sbup_daclk_clength[4] | |

**Table 7-8, config_byte_shell1**

| cfg_downBWMode[1:0] | cfg_up_clk_divider[1:0] | cfg_sbup_daclk_clength[10:0]<br>**supported minimum value** |
|---|---|---|
| 11 | 01 | 14 |
| 11 | 10 | 20 |
| 11 | 11 | 26 |
| 10, 00 | 00 | 8 |
| 10, 00 | 01 | 14 |
| 10, 00 | 10 | 20 |

**Table 7-9, Rule for minimum cfg_sbup_daclk_clength paramter**

| cfg_spi_over_sb | cfg_downBWMode | **C** = cfg_sbup_daclk_clength[10:0]<br>**resulting data rate (Mbit/s)** |
|---|---|---|
| 0 | 11 | $125 * 10^6 / C$ |
| 0 | 10, 00 | $62,5 * 10^6 / C$ |
| 1 | 11 | $125 * 10^6 / (2 * C)$ |
| 1 | 10, 00 | $62,5 * 10^6 / (2 * C)$ |

**Table 7-10, Formula for resulting uplink datarate**

**Note: cfg_spi_over_sb = 1 is mandatory for “Ashell SW-emulation use case”**

| config_byte_shell2 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | Appl | cfg_sbup_daclk_clength[3] | AShell<br>configures data rate<br><br>**Note:** valid if cfg_sbup_daclk[1:0] = “10” |
| 6 | Appl | cfg_sbup_daclk_clength[2] | |
| 5 | Appl | cfg_sbup_daclk_clength[1] | |
| 4 | Appl | cfg_sbup_daclk_clength[0] | |
| 3 | 0 | cfg_sbup_dwidth | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 2 | 1 | cfg_sbup_daclk[1] | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 1 | 0 | cfg_sbup_daclk[0] | |
| 0 | 0 | cfg_sbdown_dwidth | AShell<br>mandatory setting for “Ashell SW-emulation use case” |

**Table 7-11, config_byte_shell2**

| config_byte_shell3 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 1 | cfg_sbdown_daclk | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 6 | 0 | cfg_ephy | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 5 | 0 | cfg_eshell | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 4 | 1 | cfg_spi_over_sb | AShell<br>mandatory setting for “Ashell SW-emulation use case” |
| 3 | 1 | cfg_crc_timeout_value [3] | AShell<br>*CRC timeout error* is generated after N consecutively received and corrupted transitions (CRC mismatch) |
| 2 | 0 | cfg_crc_timeout_value [2] | |
| 1 | 0 | cfg_crc_timeout_value [1] | |

| | | | |
|---|---|---|---|
| 0 | 1 | cfg_crc_timeout_value [0] | |

**Table 7-12, config_byte_shell3**

| config_byte_shell4 | | | |
|---|---|---|---|
| Bit | Value | Name | Description |
| 7 | 1 | reserved<br>cfg_window_size[3] | AShell<br><br>do not change |
| 6 | 1 | reserved<br>cfg_window_size[2] | |
| 5 | 0 | reserved<br>cfg_window_size[1] | |
| 4 | 0 | reserved<br>cfg_window_size[0] | |
| 3 | 1 | cfg_arq_off | AShell<br>selects “data integrity only” mode<br>mandatory setting for “Ashell SW-emulation use case” |
| 2 | 1 | cfg_suppress_ita | AShell<br>selects “payload only” mode<br>mandatory setting for “Ashell SW-emulation use case” |
| 1 | 0 | reserved | |
| 0 | 0 | reserved | |

**Table 7-13, config_byte_shell4**

## 7.1 Example for application specific parameters

| Parameter | Value | Description |
|---|---|---|
| cfg_up_clk_divider | 01 | Upstream channel bandwidth 31.25MHz |
| Cfg_pxdata_width | 10 | Bit width of pixeldata 18bits |
| Cfg_px_out_ctrl_piggyback | 10 | Control signals with even pixels only,<br>→ allows higher netto bandwidth for pixel link see table of APIX standard |
| cfg_downBwMode | 11 | Downstream bandwidth mode 1000MBit/s |

| | | |
|---|---|---|
| cfg_sbup_daclk_clength | 128 (dec) | Rate of upstream sideband → 488 KHz |

**Table 7-1, Example for application specific parameters**

Resulting configuration vector based on application specific values of Table 7-1

| | |
|---|---|
| config_byte_1 | C3h |
| config_byte_2 | ACh |
| config_byte_3 | 86h |
| config_byte_4 | 00h |
| config_byte_5 | E1h |
| config_byte_6 | C0h |
| config_byte_7 | 93h |
| config_byte_shell_1 | 08h |
| config_byte_shell_2 | 04h |
| config_byte_shell_3 | 99h |
| config_byte_shell_4 | CCh |

**Table 7-2, Example of Indigo Apix configuration vector**

# 8 INAP125T24 Device Configuration

| Address (hex) | Bit # | Parameter | Configuration Value | Comment |
|---|---|---|---|---|
| 00 | 7:0 | PROM_start | 1011_1101 | PROM valid byte 0 |
| 01 | 2:0 | reserved | 000 | do not change |
| | 3 | dedicated upstream | 0 | mandatory setting for “Ashell SW-emulation use case”<br>(dedicated upstream link enabled) |
| | 4 | embedded upstream | 1 | mandatory setting for “Ashell SW-emulation use case”<br>(embedded upstream link disabled) |
| | 5 | reserved | 1 | do not change |
| | 6 | bandwidth mode | Appl. | downstream bandwidth<br>0: 500 MBit/s<br>1: 1000 MBit/s<br><br>**Note**: downstream bandwidth setting has to match related receiver device configuration |
| | 7 | reserved | 1 | do not change |
| 02 | 1:0 | pixel data width | Appl. | selects the width of pixel data to be transmitted<br><br>00: reserved 01: reserved<br>10: 18 bit 11: 24 bit<br><br>**Note**: width of pixel data setting has to match related receiver device configuration |
| | 3:2 | control signal transmit mode | Appl. | transmission of pixel control signals<br>00: never<br>01: unused<br>10: on every second (even) pixels<br>11: on each pixel<br><br>**Note**: pixel control signals setting has to match related receiver device configuration |
| | 4 | reserved | 1 | do not change |
| | 5 | pixel clock active edge | Appl. | 0: falling edge<br>1: rising edge |
| | 7:6 | upstream link serial clock | Appl. | upstream bandwidth<br><br>downstream bandwidth<br>1000 Mbit/s 500 MBit/s<br><br>00: 62.50 Mbit/s 62.50 MBit/s<br>01: 41.67 Mbit/s not applicable<br>10: 31.25 Mbit/s 31.25 MBit/s<br>11: not applicable 20.83 MBit/s<br><br>**Note**: upstream bandwidth setting has to match related receiver device configuration |
| 03 | 3:0 | Upstream link data recovery | Appl. | 0011 Upstream link serial clock 62.5/20.83 MHz<br>0001 Upstream link serial clock 41.67/31.25 MHz |
| | 7:4 | reserved | 0000 | do not change |
| 04 | 0 | reserved | 0 | do not change |
| | 4:1 | reserved | 1000 | do not change |

| Address (hex) | Bit # | Parameter | Configuration Value | Comment |
|---|---|---|---|---|
| | 7:5 | reserved | 100 | do not change |
| 05 | 7:0 | PROM_end | 1001_1001 | PROM valid byte 1 |

# 9 Order of Synchronization Tasks

In a layered communication system, several levels of synchronization must be established.

For an APIX communication system they are listed in the following table:

| Nr | Layer Module | Synchronization at Level | Purpose |
|---|---|---|---|
| (3) | Remote Handler | (Activate Access) | Control of write access (unlock sequence) |
| (2) | AShell | Transaction | Alignment to transaction boundary |
| (1) | Apix PHY | Serial Frame | Alignment to frame boundary |
| (0) | | Serial bit | Data/bit recovery |

**Table 9-1, Synchronization Tasks**

Synchronization (0) and (1) are established by hardware - for details please refer to [2].

As the Ashell TX is implemented in software, synchronization (2) must also be done in software. A simplified algorithm can be applied for the “software emulation use case”. It is described in document [4], chapter “Simplified Alignment Algorithm”.

Remotehandler register write access to Indigo is locked after startup. Therefore an unlock pattern must be sent. Please refer to document [1], chapter “Locked AHB write master”.

If any lower layer is in an unsynchronized state, the higher layers can not communicate.

The synchronization status of each layer can be checked by the following status registers or signals:

| Nr | Layer Module | RX-side | TX-side |
|---|---|---|---|
| (3) | Remote Handler | Indigo Remote handler<br>AHBMlock.lock | Not applicable |
| (2) | AShell | Indigo Remote handler<br>ASStatus.operational | TX AShell<br>The "operational" signal is software implementation specific |
| (1) | Apix PHY | Indigo Remote handler<br>ASStatus.rx_down_ready | INAP125T24<br>(Signal TX_ERROR = low) |
| (0) | | | |

**Table 9-2, Synchronization Status information**

After reaching a synchronized state at all levels, register write and read requests can be sent.

# 10 References

[1] “MB88F332, LSI Product Specification”, FUJITSU Microelectronics Europe GmbH
[2] “APIX Industrial Standard”, Inova Semiconductors GmbH
[3] “APIX Automotive Shell Technical Documentation”, Inova Semiconductors GmbH
[4] “Appendix B to [3]”, Inova Semiconductors GmbH
[5] “INAP125T24, Digital Automotive Pixel Link, Datasheet“, Inova Semiconductors GmbH