How to set up predefined message answer using STUSB4531?

Introduction

The STUSB4531 can respond in an autorun mode to a set of messages, based on the configuration of the NVM or I²C register.

This application note provides a list of messages that can be answered automatically and details on how to configure or customize the device in line with application capabilities.

1 Message list

In autorun mode, the STUSB4531 can reply to the following messages without software intervention.

Messages that shall always be answered by sinks:

- Source_Capabilities
- Get_Sink_Cap
- Get_Sink_Cap_Extended
- Get_Revision

Messages that shall be answered based on product declaration:

- Get_Battery_Cap
- Get_Battery_Status
- DR_Swap
- VCONN_Swap
- Discover_Identity

2 PD2 vs PD3

Unless otherwise specified in this document, when an autonomously prepared message is NOT_SUPPORTED, and a contract has been established in PD2, the ‘REJECT’ message is sent instead (see [6]).

3 Message answer content to

3.1 Source_Capabilities

The **REQUEST** message is calculated using a proprietary algorithm (see [1]) based on:

- source_capabilities received
- sink PDOs list, calculated with
  - NUM_PDO.NUM_SNK_PDO
  - SNK_PDO_PARAMS
  - SNK_PDO_CAPABILITIES
  - SNK_APDO_FILL_x
- DEVICE_PDP register
- ALGO register settings
- calculated source PDP

Power negotiation is handled autonomously by the STUSB4531 when it receives the source_capabilities message.

Note: STUSB4531-GUI (see [5]) helps visualize sink PDOs exposed in the sink capability message

3.2 Get_Sink_Cap

Sink_Capabilities message lists Sink power data objects (PDOs) based on:

- NUM_PDO.NUM_SNK_PDO
- SNK_PDO_PARAMS
- SNK_PDO_CAPABILITIES
- SNK_APDO_FILL_x
- DEVICE_PDP register
- ALGO register settings

Note: STUSB4531-GUI (see [5]) helps visualize sink PDOs exposed in the sink capability message.

3.3 Get_sink_cap_extended

Sink_capabilities_extended message is filled by using the following registers:

- VID
- PID
- XID
- FW_REVISION
- HW_REVISION
- DPM_CTRL
- SNK_PDO_CAPABILITIES
- PDP
- REQUEST_SRC_PDP

Based on [6] message content is:

Table 1. Sink_Cap_Extended message

Byte(s)	Field name	STUSB4531 field	Hardcoded value
1..0	VID	VID_HIGH VID_LOW	
3..2	PID	PID_HIGH PID_LOW	
7..4	XID	XID_3 XID_2 XID_1 XID_0	
8	FW Version	FW_REVISION	
9	HW Version	HW_REVISION	
10	SKEDB	-	01h
11	Load Step	-	00h
13..12	Sink Load Characteristics	-	0000h
14	Compliance	-	00h
15	Touch Temp	-	00h
16	Battery Info	Bits 3:0: DPM_CTRL.BAT_PRES & not(DPM_CTRL.BAT_SWP) Bits 7:4: DPM_CTRL.BAT_PRES & DPM_CTRL.BAT_SWP	
17	Sink Modes	Bit0: ALGO.SINK_FEATURE[7] Bit1: not(SNK_PDO_CAPABILITIES.UNCONSTRAINED_POWER) Bit2: SNK_PDO_CAPABILITIES.UNCONSTRAINED_POWER Bit3: DPM_CTRL. BAT_PRES	Bit4: 0 Bit5: 1 Bit 6..7: 00
18	SPR Sink Minimum PDP	REQUEST_SRC_PDP.SINK_MIN	
19	SPR Sink Operational PDP	REQUEST_SRC_PDP.SINK_OP	
20	SPR Sink Maximum PDP	REQUEST_SRC_PDP.SINK_MAX	
21	EPR Sink Minimum PDP	-	00h
22	EPR Sink Operational PDP	-	00h
23	EPR Sink Maximum PDP	-	00h

3.4 Get_Revision

The Revision message indicates the USB Power Delivery specification revision supported by the device. The message is based on the PD_REVISION and PD_VERSION registers.

3.5 Get_Battery_Cap

If the DPM_CTRL.BAT_PRES bit is set, the Battery_Capability message is sent.

If the bit is not set, the NOT_SUPPORTED message is sent.

The Get_Battery_Cap command requests information about a battery reference in the BATTERY_CAP_REF[7:0] field.

The BAT_SWAP bit is part of the DPM_CTRL register.

Table 2. Battery_capability message

Byte(s)	Field name	STUSB4531 field	Hardcoded value
1..0	VID	VID_HIGH VID_LOW	
3..2	PID	PID_HIGH PID_LOW	
5..4	Battery Design Capacity	-	If Battery Type FFFFh Else 0h
7..6	Battery Last Full Charge Capacity	-	If battery type FFFFh Else 0h
8	Battery Type	Not (BATTERY_CAP_REF [0] & not(BAT_SWAP) or NOT(BATTERY_CAP_REF [4] & BAT_SWAP)	

3.6 Get_Battery_Status

The Get_Battery_Status command requests information about a battery slot in the BATTERY_STATUS_REF[7:0] field.

If the DPM_CTRL.BAT_PRES bit is set, the Battery_Status message is sent when required. Otherwise, the NOT_SUPPORTED message is sent.

The BAT_SWAP bit belongs to the DPM_CTRL register.

Table 3. Battery_Status message

Byte(s)	Field name	STUSB4531 field	Hardcoded value
31..16	Battery Present Capacity	-	FFFFh
15..12	Reserved	-	0h
11..10	Battery Charging Status	-	2h
9	Battery Present	(BATTERY_STATUS_REF [0] & not(BAT_SWAP)) or (BATTERY_STATUS_REF [4] & BAT_SWAP)	
8	Invalid Battery Reference	Not (Battery Present)	
7..0	Reserved	-	0h

3.7 DR_Swap

The DR_SWAP_2_UFP and DR_SWAP_2_DFP bits are part of the DPM_CTRL register.
The DATA_ROLE bit is part of the PD_STATUS register.
The DRS_AUTO_ACCEPT bit is part of the APPLI_CTRL register.

Table 4. DR_SWAP management

DR_SWAP_2_UFP	DR_SWAP_2_DFP	DATA_ROLE	DRS_AUTO_ACCEPT	Message sent
0	0	x	x	NOT_SUPPORTED
0	1	UFP	0	WAIT
0	1	UFP	1	ACCEPT
0	1	DFP	x	REJECT
1	0	UFP	x	REJECT
1	0	DFP	0	WAIT
1	0	DFP	1	ACCEPT
1	1	x	0	WAIT
1	1	x	1	ACCEPT

3.8 VCONN_Swap

In the following table, the VCONN_SRC and VCONN_ON bits are in the PD_STATUS register.
The VCONN_SWAP_TO_ON, VCONN_SWAP_TO_OFF and VCONN_EN bits are in the DPM_CTRL register.
The STUSB4531 is VCONN capable only if GPIO_SETTING.GPIO_CONF[2:0] = b01 and VCONN_EN bit is set.

Table 5. VCONN_Swap management

VCONN capable	VCONN_SWAP_TO_OFF	VCONN_SWAP_TO_ON	VCONN_SRC	Message sent
1	0	0	x	REJECT
1	x	1	0	ACCEPT
1	x	1	1	REJECT
1	1	x	0	REJECT
1	1	x	1	ACCEPT
0	x	x	x	NOT_SUPPORTED

3.9 Vendor Defined Messages (VDM)

VDM can be either structured or unstructured.
Unstructured VDM are always ignored in PD2 and not supported in PD3.
The VDM_SUPPORT bit is part of the DPM_CTRL register.
Managing VDM sequences uses the HYBRID mode. The methodology is explained in the dedicated application note [9] for VDM messaging.

3.9.1 Discover_Identity

If the VDM_SUPPORT bit is set, the BUSY VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.2 Discover SVIDs

If the VDM_SUPPORT bit is set, the BUSY VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.3 Discover Modes

If the VDM_SUPPORT bit is set, the BUSY VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.4 Enter Mode

If the VDM_SUPPORT bit is set, the NAK VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.5 Exit Mode

If the VDM_SUPPORT bit is set, the NAK VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.6 Attention

If the VDM_SUPPORT bit is set, the NAK VDM message is answered. Otherwise, the ‘NOT_SUPPORTED’ message is sent in PD3 and the message is ignored in PD2.

3.9.7 Others

Other VDM messages are ignored in PD2 and the ‘NOT_SUPPORTED’ message is sent in PD3.

3.9.8 Busy message structure

Table 6. BUSY VDM message

Bit(s)	Field	Value
B31..16	Standard or Vendor ID	16’hFF00
B15	VDM Type	b1 (structured)
B14..13	Structured VDM Version	b01 if PD#3 else b00
B12..11	Structured VDM Version (minor) in PD#3 (01) And Reserved in PD2 (00)	b01 if PD#3 else b00
B10..8	Object Position	Send back the same received info (Object Position) if Command={enter mode, exit mode, attention} Else b000
B7..6	Command Type	**b11 (BUSY)**
B5	Reserved	b0
B4..0	Command	Send back the same received info (Command)

3.9.9 Nak message structure

Table 7. **NAK VDM message**

Bit(s)	Field	Value
B31..16	Standard or Vendor ID	16'hFF00
B15	VDM type	b1 (Structured)
B14..13	Structured VDM Version	b01 if PD#3 else b00
B12..11	Structured VDM Version (minor) in PD#3 (01) And Reserved in PD2 (00)	b01 if PD#3 else b00
B10..8	Object Position	Send back the same received info (Object Position) if Command={enter mode, exit mode, attention} Else b000
B7..6	Command Type	**b10 (NAK)**
B5	Reserved	b0
B4..0	Command	Send back the same received info (command)

4 References

[1] DS15023: STUSB4531 datasheet
[2] RM0562: STUSB4531 register map
[3] AN6406: How to handle STUSB4531 interrupts
[4] AN6466: How to identify the current explicit contract using STUSB4531?
[5] STSW-STUSB020: graphical user interface for STUSB4531
[6] USB power delivery specification: https://www.usb.org/document-library/usb-power-delivery
[7] USB type-C cable and connector specification: https://www.usb.org/document-library/usb-type-cr-cable-and-connector-specification-release-25
[8] AN6469: How to change voltage/current by posting a new request message using hybrid
[9] AN6501: How to post VDMs using STUSB4531

Revision history

Table 8. **Document revision history**

Date	Revision	Changes
09-Jun-2026	1	Initial release.

Contents

List of tables