



## How to change voltage/current by posting a new request message using hybrid

### Introduction

#### Hybrid mode

The STUSB4531 includes a patented hybrid mode that allows the system to send or to modify a request message by using I<sup>2</sup>C read and write operations. The STUSB4531 flags the system by using its ALERT interrupt pin.

## 1 I<sup>2</sup>C commands

---

Two generic functions are defined in this document to indicate I<sup>2</sup>C read or write actions.

For both functions, the device address is omitted to avoid overloading the document. The device address can be 0x28 or 0x29 as specified in the datasheet.

I2C\_Read(register\_address, number\_of\_byte):

- Register\_address: indicates the address starting point to be read
- Number\_of\_byte: indicates if a single or multiple consecutive reads will be performed

I2C\_Write(register\_address, data, number\_of\_byte)

- Register\_address: indicates the address starting point to be written
- Data: is the content to be written
- Number\_of\_byte: indicates if a single or multiple consecutive writes will be performed

## 2 STUSB4531 prerequisite

As per USBPD specification, besides 'request message', no other message can be sent prior to being in explicit contract, i.e. the policy engine being in the ready state.

### 2.1 Interrupt used for sending a request message

Some PD interrupts need to be unmasked in order to indicate that:

- a message can be posted in TxBuffer: TX\_BUFFER\_READY
- a request message has been accepted by the source: TRANSITION\_WINDOW
- the source has sent PSRdy, signaling the end of the power transition: TRANSITION\_END

As mentioned in [3], PD\_STATUS\_AL needs to also be unmasked.

### 2.2 TX registers for sending message

To send any message, the USBPD controller needs to be aware of:

- the number of bytes to be sent: TX\_BYTE\_CNT
- the message header content: TX\_HEADER
- the message data object content, if any: TX\_DATA\_OBJ\_224BITS (this register needs to be accessed in multi read/write only)

The following section will describe how to fill those registers.

## 3 USB PD message construction for request message

### 3.1 Tx byte count

Tx byte count is defined by the number of bytes, including the header, to be transmitted.

The request message has single a 32-bit data object (4 bytes).

Tx byte count is then 2 bytes for the header and 4 bytes for the data objects:

@TX\_BYTE\_CNT = 0x06

### 3.2 Request header

As per [5], the message header construction for a request message is:

Table 1. Header

Bit(s)	Field name	Value	Comment
15	Extended	0	Control or data message
14...12	Number of data objects	1	Single 32-bit data object
11...9	MessageID	Handled by hardware	
8	Port power role	0	Sink
7..6	Specification revision	<a href="#">Section 3.2.1: Specification revision</a>	
5	Port data role	<a href="#">Section 3.2.2: Port data role</a>	
4..0	Message type	00010	Request message code

The TX\_HEADER register needs to be filled with the above request header.

#### 3.2.1 Specification revision

Specification revision is established during the first explicit contract and is static until disconnection occurs.

- Rx\_buffer[15:0] = I2C\_Read(RX\_HEADER,2)
- Spec\_rev= Rx\_buffer[4:6]

#### 3.2.2 Port data role

As per [5], sink is UFP by default. However, the data role might have changed before the MCU is up and running, depending on STUSB4531 NVM settings.

The port data role can be ascertained by checking @PD\_STATUS.DATA\_ROLE.

### 3.3 RDO

The RDO content needs to be written to the @TX\_DATA\_OBJ\_224BITS register.

The RDO is a 32-bit object.

The voltage and current bit fields, located in bits [19..0], depend on the requested source PDO: see [4].

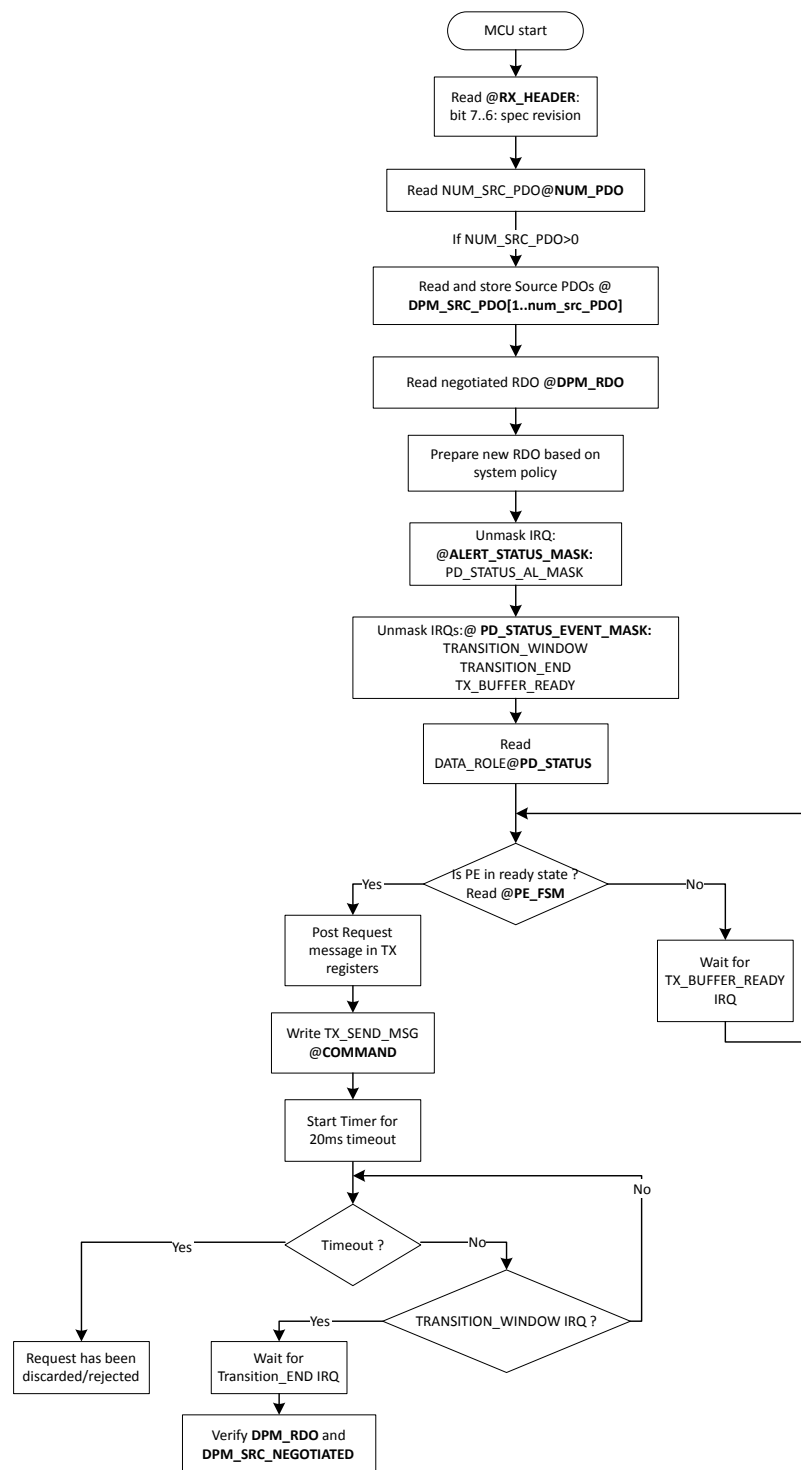
**Table 2. RDO data object**

Bit(s)	Field name	Value
31..28	Object position	Position extracted from Source_capability list
27	Giveback	0b: depreciated field
26	Capability mismatch	Set only to indicate that Sink would prefer other voltage/current offer
25	USB comm capable	0b/1b: in line with USB data support
24	No USB suspend	1b: Recommended
23	Unchunked extended message supported	0b: not supported
22	EPR capable	0b: no EPR stack available
21..20	Reserved	00b
19..0	Voltage and current bit fields	See [4]

## 4 Sending a new request flow chart

Figure 1 describes the sequence to post a new request to the source, and to confirm that the posted request has been taken into account.

Figure 1. Sending a new request flow chart



---

## 5 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] USB Power Delivery specification: <https://www.usb.org/document-library/usb-power-delivery>
- [6] USB Type-C Cable and Connector Specification: <https://www.usb.org/document-library/usb-type-cr-cable-and-connector-specification-release-24>

## Revision history

**Table 3. Document revision history**

Date	Version	Changes
25-Mar-2026	1	Initial release.

## Contents

<b>1</b>	<b>I<sup>2</sup>C commands</b> .....	<b>2</b>
<b>2</b>	<b>STUSB4531 prerequisite</b> .....	<b>3</b>
<b>2.1</b>	Interrupt used for sending a request message .....	3
<b>2.2</b>	TX registers for sending message .....	3
<b>3</b>	<b>USB PD message construction for request message</b> .....	<b>4</b>
<b>3.1</b>	Tx byte count .....	4
<b>3.2</b>	Request header .....	4
<b>3.2.1</b>	Specification revision .....	4
<b>3.2.2</b>	Port data role .....	4
<b>3.3</b>	RDO .....	5
<b>4</b>	<b>Sending a new request flow chart</b> .....	<b>6</b>
<b>5</b>	<b>References</b> .....	<b>7</b>
	<b>Revision history</b> .....	<b>8</b>

## List of tables

<b>Table 1.</b>	Header . . . . .	4
<b>Table 2.</b>	RDO data object . . . . .	5
<b>Table 3.</b>	Document revision history . . . . .	8

## List of figures

Figure 1. Sending a new request flow chart ..... 6

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers’ market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved