How to handle STUSB4531 interrupts

Introduction

Interrupts in STUSB4531

The STUSB4531 includes an interrupt management system to notify the application about various types of events.

The interrupt pin, which is used to flag an event to the application of, is an open-drain I/O named ALERT.

Event information that has triggered the alert event is available in the I²C registers.

I²C commands

Two generic functions are defined in this document to indicate I²C 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

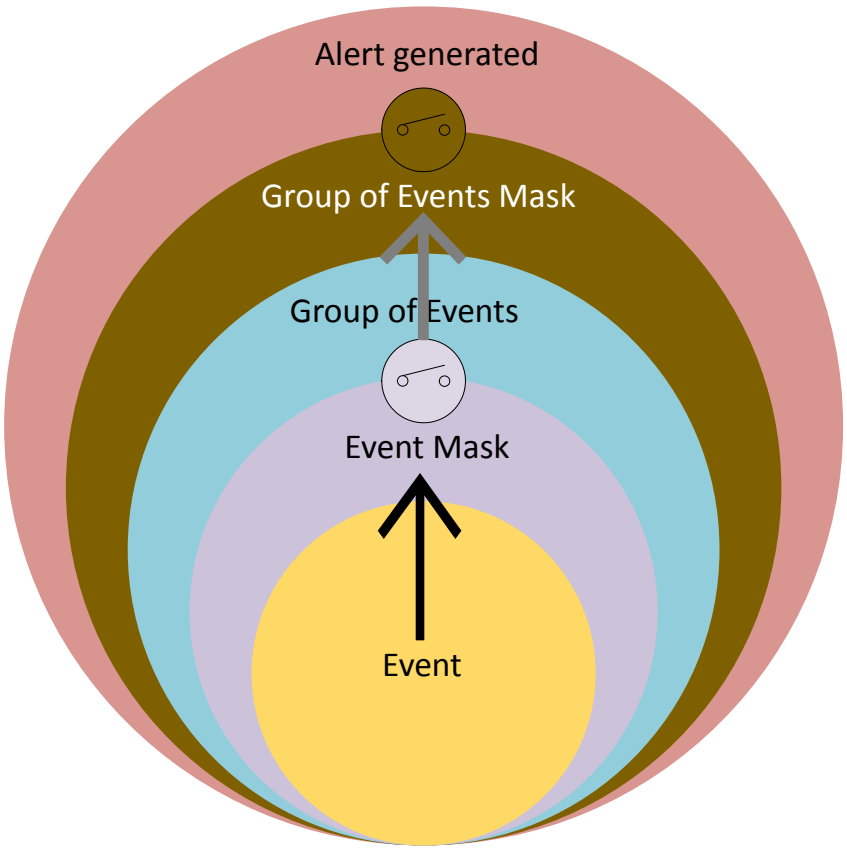
1 STUSB4531 prerequisite

1.1 Interrupt default setup

By default, all functional interrupts are masked to the application. Only hardware fault interrupts are non-maskable and could lead to an alert event.

1.2 Interrupt structure

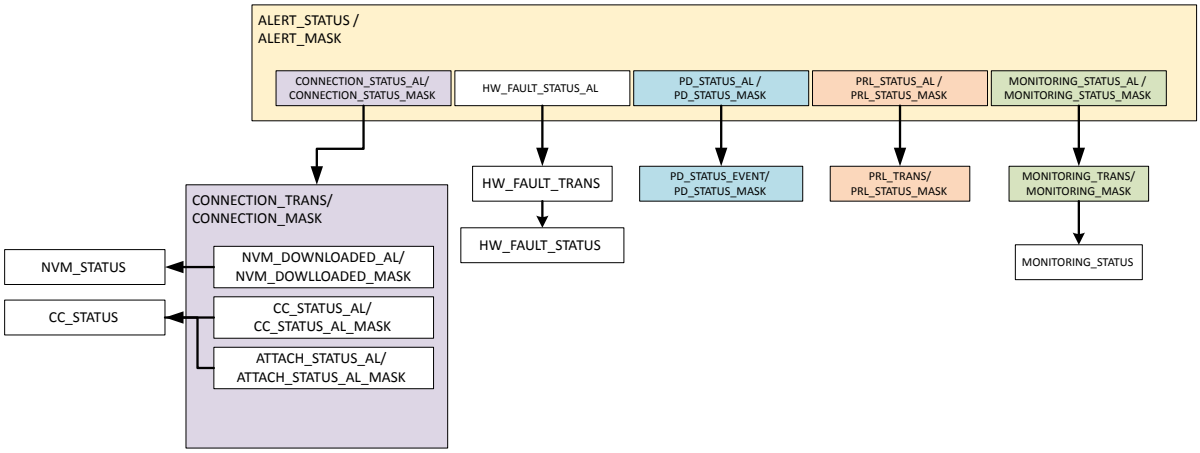
Figure 1. Interrupt generation structure



To generate an Alert, the Event and Group it belongs to need to be unmasked.

Only hardware faults have a direct path from the Event to the Alert generation.

Figure 2. Interrupt register interlinking



1.3 Event groups

STUSB4531 interrupts are structured into five groups:

- Configuration and connection: CONNECTION_STATUS_AL
- Communication Protocol Layer: PRL_STATUS_AL
- Power Delivery: PD_STATUS_AL
- Vbus Monitoring: MONITORING_STATUS_AL
- Hardware faults: HW_FAULT_STATUS_AL

When an unmasked event occurs in a group, the associated bit in the ALERT_STATUS register (address 0x0010) will flag an Alert, assuming the associated group is unmasked.

Figure 3. ALERT_STATUS and ALERT_STATUS_MASK registers address 0x0010 and 0x0011

0x0010	*ALERT_STATUS*	[4] CONNECTION_STATUS_AL [3] HW_FAULT_STATUS_AL [2] PD_STATUS_AL [1] PRL_STATUS_AL [0] MONITORING_STATUS_AL
0x0011	*ALERT_STATUS_MASK*	[4] CONNECTION_STATUS_AL_MASK [2] PD_STATUS_AL_MASK [1] PRL_STATUS_AL_MASK [0] MONITORING_STATUS_AL_MASK

1.4 Reading events

The following registers are cleared upon an I²C read action:

- CONNECTION_TRANS: address 0x0027
- PRL_TRANS: address 0x0017
- PD_STATUS_EVENT: address 0x0012
- MONITORING_TRANS: address 0x0014
- HW_FAULT_TRANS: address 0x001C

Consequently, the content of these registers must be stored after being read to take the required action.

2 Event signification

2.1 Configuration and connection - CONNECTION_STATUS_AL:

Figure 4. Event registers linked to CONNECTION_STATUS_AL

0x0027	*CONNECTION_TRANS*	[2] NVM_DOWNLOADED_AL [1] CC_STATUS_AL [0] ATTACH_STATUS_AL
0x0028	*CONNECTION_MASK*	[2] NVM_DOWNLOADED_AL_MASK [1] CC_STATUS_AL_MASK [0] ATTACH_STATUS_AL_MASK

This register mainly provides information on STUSB4531 startup.
Each event is linked to a dedicated status register, as illustrated in Figure 2. Interrupt register interlinking.

2.1.1 NVM_DOWNLOADED_AL

When set, the NVM_STATUS register should be read.

Figure 5. NVM_STATUS register

0x001F	*NVM_STATUS*	[7] NVM_LOADED [6] NVM_END_OP [5:0] NVM_WRITE_STATUS

During VDD power-up operation, NVM_LOADED indicates that the STUSB4531 is ready to operate. All I²C registers can then be accessed.

NVM_END_OP and NVM_WRITE_STATUS are used only during NVM read and write operations.

2.1.2 CC_STATUS_AL

When set, it indicates that an event occurred on the CC pin. A source Rp value change will generate a CC_STATUS_AL event, as CCx_state will change.
The real-time CC_STATUS register can then be read.

Figure 6. CC_STATUS register

0x001A	*CC_STATUS*	[7] PLUG_ORIENTATION [6] ATTACH_STATUS [5] LOOKING_4_CONNECTION [4] CONNECT_RESULT [3:2] CC2_STATE [1:0] CC1_STATE

2.1.3 ATTACH_STATUS_AL

When set, it indicates that the STUSB4531 is in the Type-C ATTACH state.
By reading the CC_STATUS register, the exact status of the connection can be determined.

2.2 Communication - Protocol Layer: PRL_STATUS_AL

Unless specified, events from this register can be available both in autorun and hybrid mode.

Figure 7. Event registers linked to PRL_STATUS_AL

0x0017	*PRL_TRANS*	[7] PRL_TX_ERR [6] PRL_TX_DISCARD [3] PRL_MSG_SENT [2] PRL_MSG_RECEIVED [1] PRL_HR_DONE [0] PRL_HR_RECEIVED
0x0018	*PRL_STATUS_MASK*	[7] PRL_TX_ERR_MASK [6] PRL_TX_DISCARD_MASK [3] PRL_MSG_SENT_MASK [2] PRL_MSG_RECEIVED_MASK [1] PRL_HR_DONE_MASK [0] PRL_HR_RECEIVED_MASK

- PRL_TX_ERR: after retrying nRetry times to send a message to the port partner without acknowledgement, the STUSB4531 notifies that message has not been sent, and the protocol error will be handled through soft reset or hard reset.
- PRL_TX_DISCARD: occurs in PD3 when an AMS [(1)] sequence is initiated by software in hybrid mode. It indicates that the message cannot be sent due to collision avoidance. Event valid only in hybrid mode.
- PRL_MSG_SENT: successful message sending and acknowledge notification. Event is available in hybrid or autorun operations.
- PRL_MSG_RECEIVED: indicates the reception of a new message. Message is available in Rx registers.
- PRL_HR_DONE: Hard reset has been sent by STUSB4531.
- PRL_HR_RECEIVED: Hard reset message received from the port partner.

1. *AMS: Atomic Message Sequence. See "USB Power delivery specification" for more details.*

2.3 Power Delivery: PD_STATUS_AL

Unless specified, events from this register can be available both in autorun and hybrid mode.

Figure 8. Event registers linked to PD_STATUS_AL

0x0012	*PD_STATUS_EVENT*	[7] TX_BUFFER_READY [2] AMS_STOPPED [1] TRANSITION_END [0] TRANSITION_WINDOW
0x0013	*PD_STATUS_EVENT_MASK*	[7] TX_BUFFER_READY_MASK [2] AMS_STOPPED_MASK [1] TRANSITION_END_MASK [0] TRANSITION_WINDOW_MASK

- TX_BUFFER_READY: This event indicates when the Tx buffer is ready to be written or updated, to post a new message or to update a predefined hardware answer.
- AMS_STOPPED: This event indicates the end of an AMS sequence initiated by the counterpart, or an AMS sequence initiated by STUSB4531 and handled by its policy engine hardware.
- TRANSITION_END: This event indicates that the power transition due to a new explicit contract is complete.
- TRANSITION_WINDOW: This event indicates that a PD request has been accepted, and the power transition phase has started.
- MONITORING_TRANS register is cleared on read register.
- MONITORING_MASK register is used to mask the event, in order not to generate an unwanted alert on monitoring events.
- MONITORING_STATUS register is providing real-time status information.

2.4 VBUS Monitoring: MONITORING_STATUS_AL

This group of events provides various information on VBUS monitoring related to different thresholds reached.

- MONITORING_TRANS register is cleared on read register.
- MONITORING_MASK register is used to mask events, in order not to generate an unwanted alert on monitoring events.
- MONITORING_STATUS register is providing real-time status information.

Figure 9. Event and status registers linked to MONITORING_STATUS_AL

0x0014	*MONITORING_TRANS*	[4] VBUS_HIGH_TRANS [3] VBUS_LOW_TRANS [1] VBUS_VALID_TRANS [0] VBUS_VSAFE0V_TRANS
0x0015	*MONITORING_MASK*	[4] VBUS_HIGH_MASK [3] VBUS_LOW_MASK [1] VBUS_VALID_AL_MASK [0] VBUS_VSAFE0V_MASK
0x0016	*MONITORING_STATUS*	[4] VBUS_HIGH_STATUS [3] VBUS_LOW_STATUS [1] VBUS_VALID [0] VBUS_VSAFE0V

- VBUS_HIGH_TRANS: This event indicates that VBUS crosses the VBUS overvoltage threshold (see V_{TH_HIGH} definition in product datasheet). VBUS_HIGH_STATUS reflects comparator output.
- VBUS_LOW_TRANS: This event indicates that VBUS crosses the VBUS undervoltage threshold (see V_{TH_LOW} definition in product datasheet). VBUS_LOW_STATUS reflects comparator output.
- VBUS_VALID_TRANS: This event indicates that VBUS crosses the VBUS_VALID threshold (see VBUS_VALID definition in product datasheet). VBUS_VALID_STATUS reflects comparator output.
- VBUS_VSAFE0V_TRANS: This event indicates that VBUS crosses the VBUS_VSAFE0V threshold (see VBUS_VSAFE0V definition in product datasheet). VBUS_VSAFE0V_STATUS reflects comparator output.

2.5 Hardware faults: HW_FAULT_STATUS_AL

Hardware faults event cannot be masked.

Figure 10. Event and status registers linked to HW_FAULT_STATUS_AL

0x001C	*HW_FAULT_TRANS*	[7] I2C_WATCHDOG [6] CS_OVP [1] TH_WARN_TRANS [0] TH_SHUTDOWN_TRANS
0x001D	*HW_FAULT_STATUS*	[1] TH_WARN [0] TH_SHUTDOWN

- I2C_WATCHDOG: This event indicates that an I^2C action has not been observed for approximately 2 seconds after ALERT has been asserted. I2C_WATCHDOG functionality is not enabled by default.
- CS_OVP: This event indicates that at least one CC pin voltage has reached V_{THOVP_CC}.
- TH_WARN_TRANS: This event indicates that the internal temperature has crossed the warning temperature threshold (see TH_{WARN} definition in product datasheet). TH_WARN indicates that the current internal temperature is greater than TH_{WARN}.
- TH_SHUTDOWN_TRANS: This event indicates that the internal temperature has crossed the shutdown temperature threshold (see $TH_{SHUTDOWN}$ definition in product datasheet). TH_SHUTDOWN indicates that current internal temperature is greater than $TH_{SHUTDOWN}$.

3 References

- STUSB4531 datasheet
- RM0562: STUSB4531 register map
- USB Power Delivery specification: https://www.usb.org/document-library/usb-power-delivery
- 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 1. **Document revision history**

Date	Version	Changes
16-Dec-2025	1	Initial release.

Contents

List of tables

List of figures