USB Behind the Scenes

Section 3 - Packets & Transaction Types

UNIVERSAL SERIAL BUS

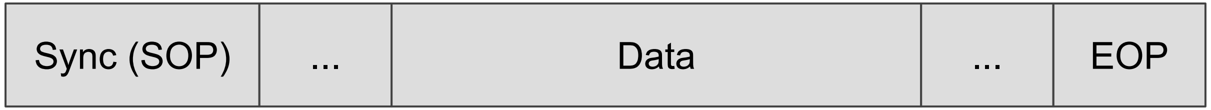
ARM Cortex-M4

STM32

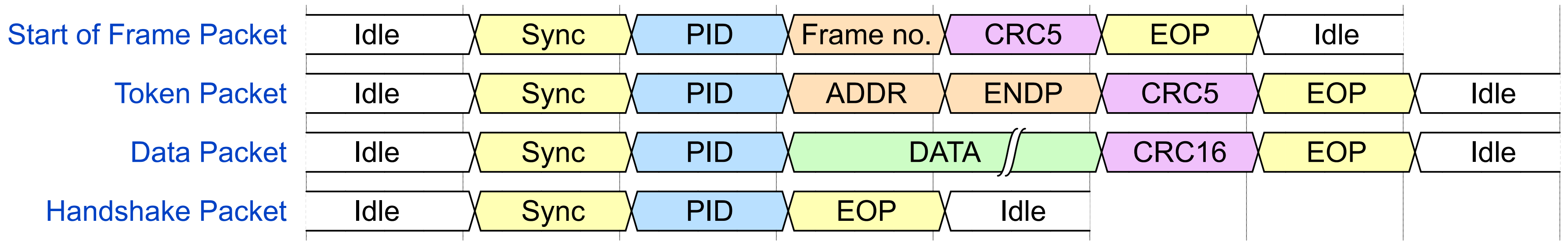
Mohammed Noureldin, Ph.D.

Packets

In USB, data is transmitted as packets.



Packet Types and Fields



Transaction

Every USB transaction consists of a sequence of three (or two) packets:

1. Token packet.
2. Data packet.
3. Handshake packet (Isochronous transfer does not use handshake packet).

Packet Identifiers (PIDs)

PID Name	PID Value
IN	0b1001
OUT	0b0001
SETUP	0b1101
SOF	0b0101

Token

PID Name	PID Value
DATA0	0b0011
DATA1	0b1011
DATA2	0b0111
MDATA	0b1111

Data

PID Name	PID Value
ACK	0b0010
NAK	0b1010
STALL	0b1110
NYET	0b0110

Handshake

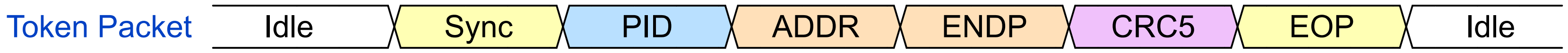
PID Name	PID Value
PRE	0b1100
ERR	0b1100
SPLIT	0b1000
PING	0b0100

Special

Token

Token packets **indicate the start of a transaction** of one of the following types:

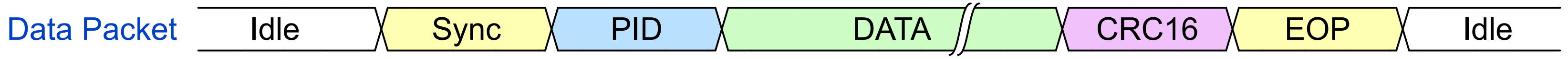
- IN: indicates that the host will read data from the device.
- OUT: indicates that the host will send data to the device.
- SETUP: indicates that the host will send SETUP data to the device



Data

Data0 and Data1 packets do the exact same task, transmitting data.

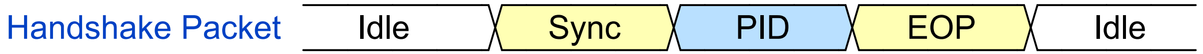
Data0 and Data1 are used alternatively just as a method of error checking. But their job is only to transfer data.



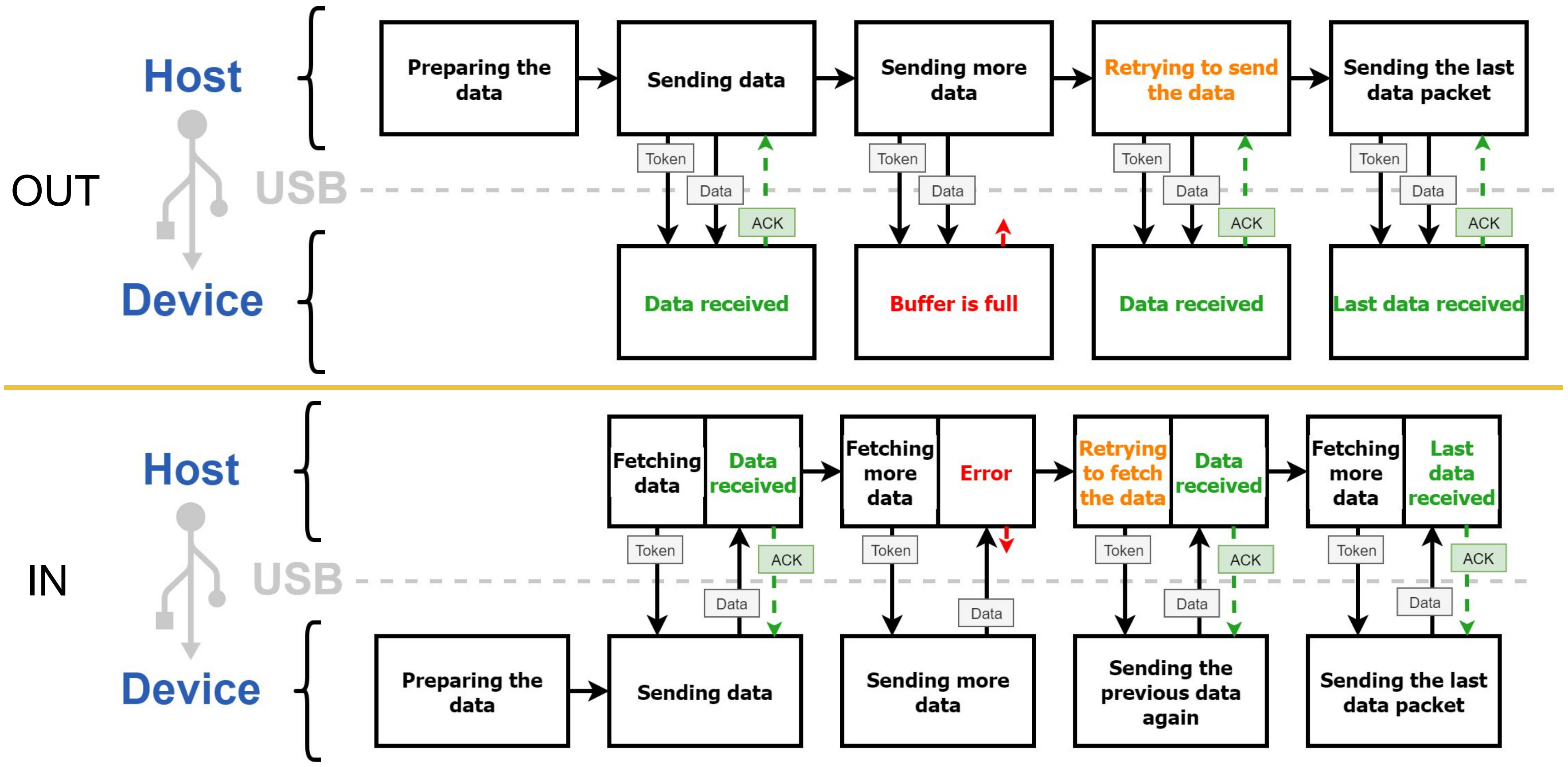
Handshake

Handshake packets **indicates the end of a transaction** with one of the following states:

- ACK:
  - The receiver acknowledges receiving the the packet without errors.
- NAK:
  - The receiver cannot receive the data.
  - The sender cannot send the data now.
- STALL:
  - An error happened, the device puts the corresponding endpoint on hold.
  - The received SETUP request is not supported.



USB is Host-Driven



Transfer Types (Endpoint Types)

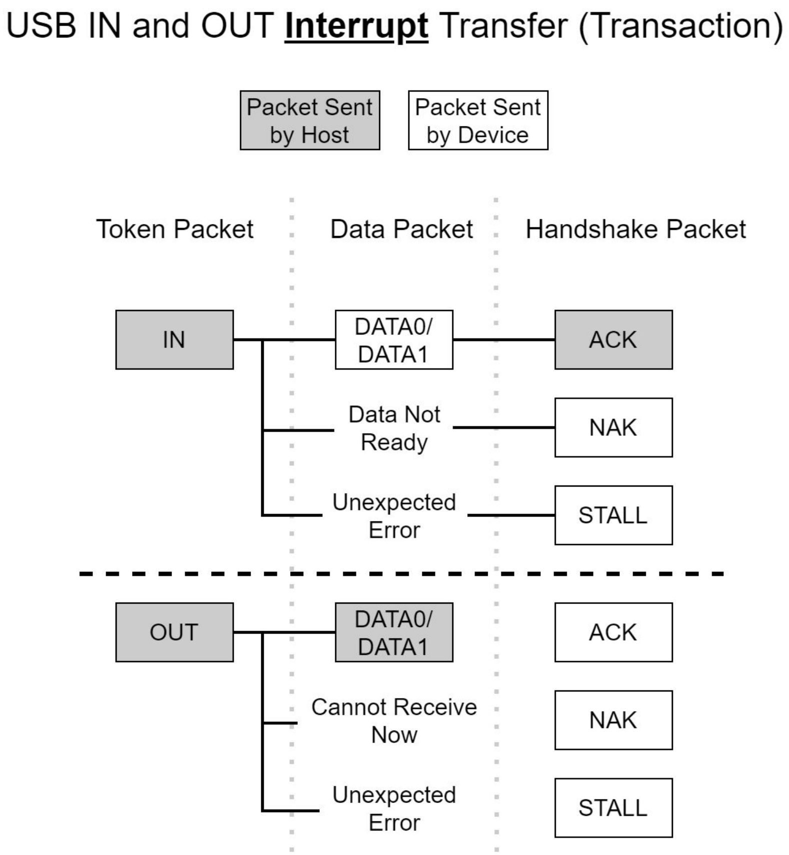
- Interrupt transfer
- Bulk transfer
- Isochronous transfer
- Control transfer

USB device tells the host about the configuration of its endpoints in the early stage of device enumeration.

Interrupt Transfer

- Periodic transfer.
- Guarantees bus bandwidth (limited latency).
- Guarantees error-free transfer (has error detection).
- Used to get updates from a device regularly.
- Maximum data payload: up to 64 bytes for FS, and up to 1024 bytes for HS.
- Example: mouse, keyboard, and joystick.

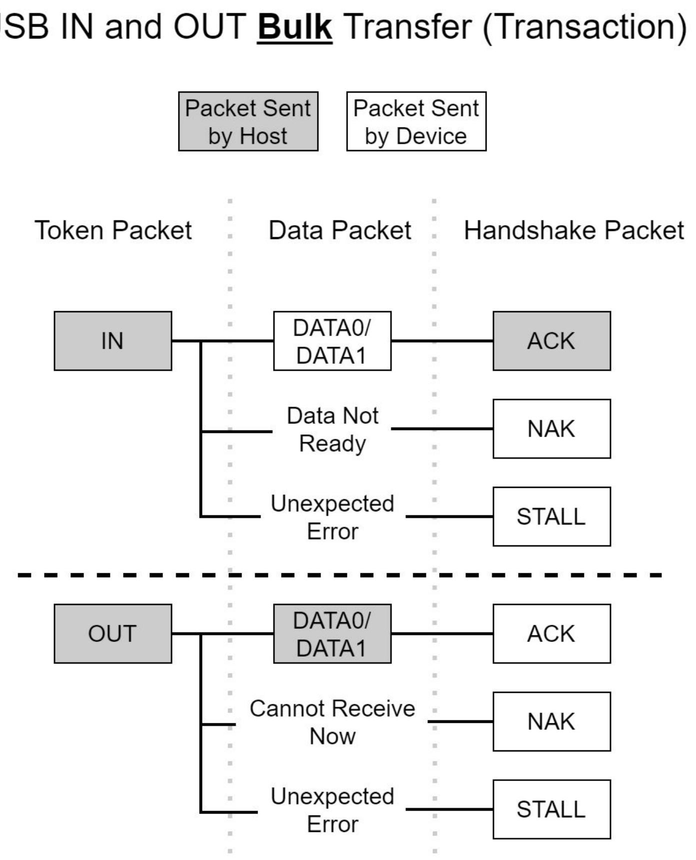
Interrupt Transfer



Bulk Transfer

- Non-periodic transfer.
- No guarantee of bus bandwidth (no specific latency guaranteed).
- Guarantees error-free transfer (has error detection).
- Used to transfer large amount of data.
- Maximum packet size: 8, 16, 32, or 64 bytes for full speed, and 512 bytes for high speed.
- Low speed does not support bulk transfer.
- Example: writing data to an external USB storage.

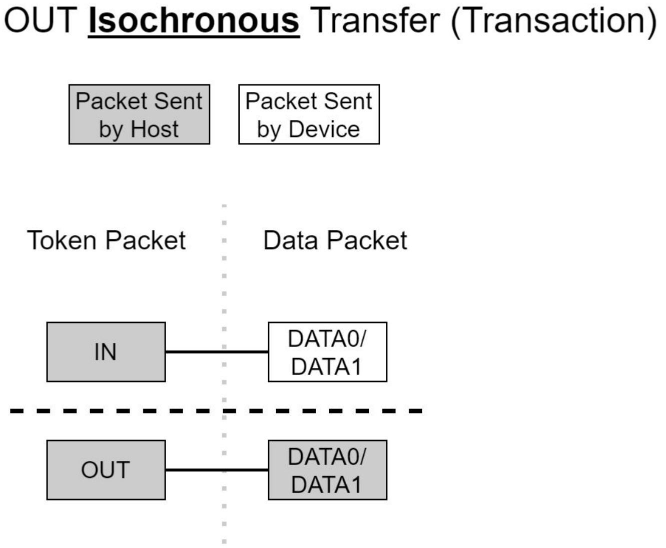
Bulk Transfer



Isochronous Transfer

- Periodic transfer.
- Guarantees bus bandwidth (limited latency).
- Does not have error detection.
- Used to transfer large amount of data without caring if some data gets missed or corrupted.
- Maximum data payload: up to 1023 bytes for FS, and up to 1024 bytes for HS.
- Example: typically use with cam stream and microphone.

Isochronous Transfer



Control Transfer

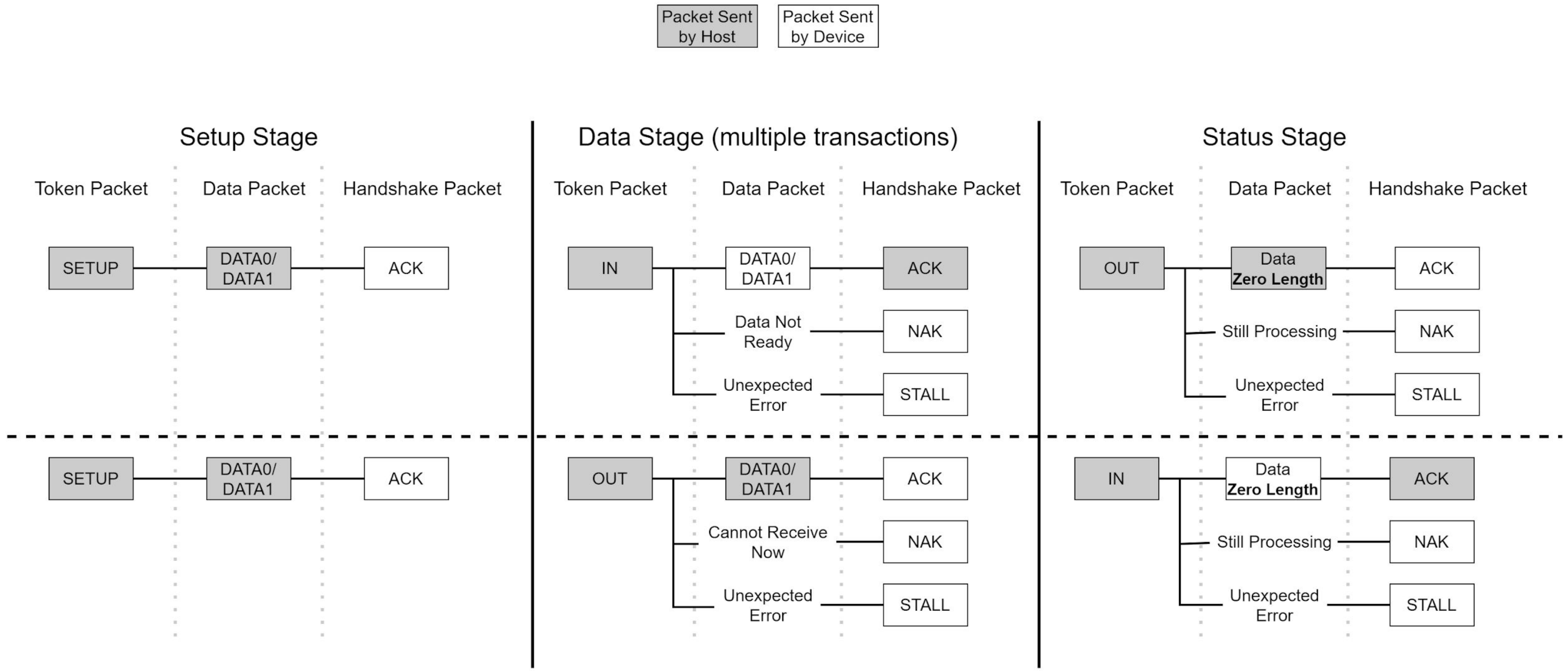
- Non-periodic transfer.
- Guarantees error-free transfer (has error detection).
- Used to transfer device enumeration and configuration packets.
- **Endpoint 0** (IN and OUT) must be configured to operate as “Control Transfer”.
- Maximum data payload: 8, 16, 32, or 64 for full speed, and 64 for high speed.

Control transfer consists of **three stages**:

1. The Setup Stage (one transaction).
2. The Data Stage (optional; zero to multiple transactions).
3. The Status Stage (one transaction).

Control Transfer

USB IN and OUT **Control** Transfer (3 Stages of Transactions)



USB Request (SETUP Transaction Data) Structure

Field	Size (bytes)	Value Description
bmRequestType	1	Data direction (to/from host) Request Type (standard, class, or vendor) Request Recipient (device, interface, endpoint, other)
bRequest	1	Request identity
wValue	2	Request specific value.
wIndex	2	Request specific value.
wLength	2	Number of bytes to transfer (if there is a data stage)

USB Standard Device Requests

bmRequestType	bRequest	wValue	wIndex	wLength	Data
1000 0000b	GET_DESCRIPTOR (0x06)	Descriptor type (H) and index (L)	Zero or Language ID	Descriptor length	Descriptor
0000 0000b	SET_ADDRESS (0x05)	Device address	0	0	-
0000 0000b	SET_CONFIGURATION (0x09)	Config value	0	0	-
1000 0000b	GET_CONFIGURATION (0x08)	0	0	1	Config value
0000 0000b	SET_FEATURE (0x03)	Feature selector	0	0	-
0000 0000b	CLEAR_FEATURE (0x01)	Feature selector	0	0	-
1000 0000b	GET_STATUS (0x00)	0	0	2	Status

⋮