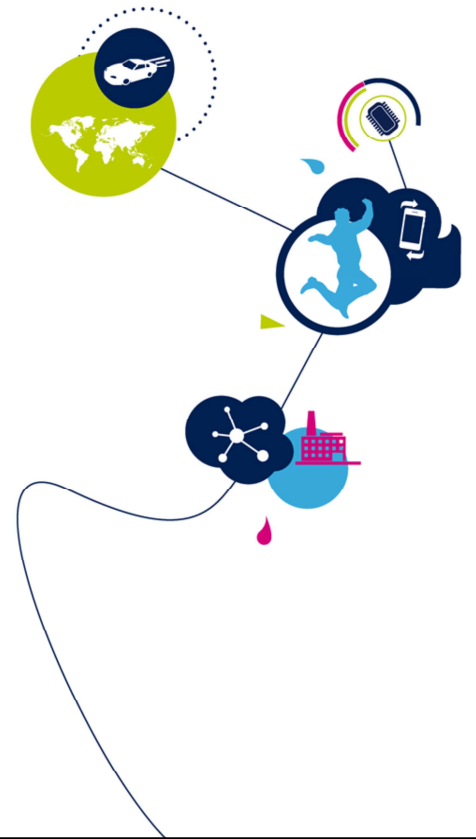


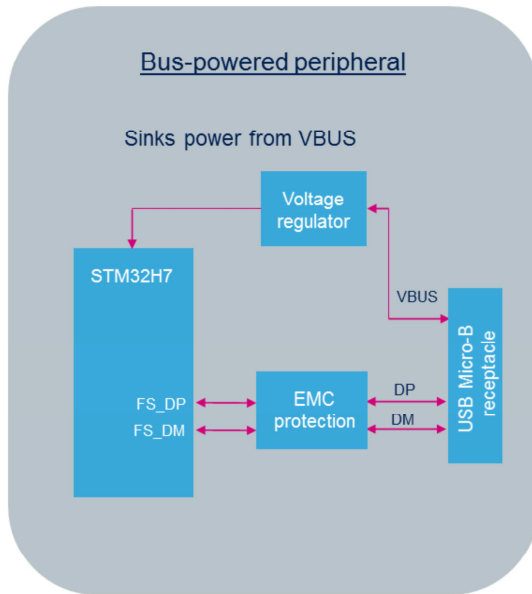
STM32H7 – USB OTG HS

USB On-The-Go High-Speed and Full-Speed interfaces

Revision 1.0



Hello, and welcome to this presentation of the STM32H7 USB High- and Full-Speed interfaces. It covers all the features of these interfaces, which are widely used to connect either a PC or a USB device to the microcontroller.



- Provides USB 2.0 Full-Speed interface, and also High-Speed ULPI (UTMI + Low Pin Interface) interface
- Device only, Host only and OTG functionality

Application benefits

- Low-power modes support
- Two independent instances, each with dedicated DMA
- Battery Charger Detection

This figure shows the connections between an STM32H7 microcontroller and a USB connector. The STM32H7 features a USB Full Speed communication interface, allowing the microcontroller to communicate typically with a PC or a USB storage device. The simplest implementation is a USB peripheral device but the STM32H7 also supports the USB “On-the-go” functionality.

- USB 2.0 Full-Speed (12 Mbit/s)
 - Control (EPO), bulk, interrupt and isochronous transfers support
 - On-the-fly endpoints configuration
 - Supports Link Power Management (LPM)
 - OTG support
 - Includes Battery Charger Detection hardware
 - Latest specification BC 1.2 supported (implementation requires software)
 - By using this specification assisted by hardware, more current (up to 1.5 A) can be safely drawn from BC 1.2 compliant chargers hence decreasing battery recharge time
- USB 2.0 High-Speed (480 Mbit/s) via ULPI
 - Same modes as above, when coupled with an external ULPI transceiver



Let's look at some of the key features of this USB Full-Speed (FS) interface, which is a USB specification 2.0 compliant interface that operates at a rate of 12 Mbit/s. In its simplest form, a USB FS device can be implemented.

Built-in support for Link Power Management adds enhanced power modes on top of the USB 2.0 specification.

In addition, the On-The-Go or "OTG" functionality enables implementation of an OTG product or an embedded host, both of which have the capacity to behave as a targeted host.

The battery charger detection function allows for increased current to be drawn from BC1.2-compliant chargers up to 1.5 A.

USB 2.0 High Speed is also available via the ULPI interface. The same modes of operation are possible when coupling with an external ULPI transceiver.

- USB On-the-Go (OTG)

- Selects the default role (device / host) based on the connected cable
 - Detected through the ID pin signal
 - Allows compatibility with device-only and host-only devices

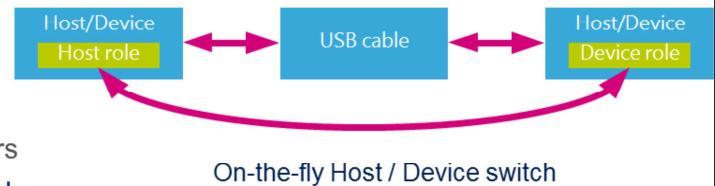
- Can switch roles at runtime through negotiation process
 - Host negotiation protocol (HNP)
 - Only point-to-point connection

- Requires micro-AB receptacle
 - It accepts both micro-A and micro-B connectors

- Not supported by ST library, but HW is capable

- Examples:

- Smart phones, tablets, cameras (file storage)
 - Device when connected to PC
 - Host when USB disk or HID (Human Interface Device) class device (keyboard, mouse) is connected



Let's explain a bit deeper the USB On-the-Go (OTG) functionality.

The default role is determined by the connection of the cable (additional ID pin inside of the connector). If the USB peripheral has only one role, the ID pin is ignored. The role of the peripheral can be switched on the fly while the two OTG devices are connected directly using a point-to-point connection. This functionality is only possible on device with micro-A or micro-B connectors. For example, some smartphones have this capability.

The peripheral is fully capable to work as OTG, but currently there is no firmware support in STM32Cube HAL libraries.

OTG_HS peripheral

5

- Two identical peripherals – OTG_HS1 and OTG_HS2
- One exception – OTG_HS2 has no ULPI interface
 - Not possible to connect HS PHY
- Both OTG_HS1 and OTG_HS2 peripherals are able to work in FS mode
- Only OTG_HS1 with external HS PHY can work in HS mode
- 4-Kbyte RAM for FIFO dedicated to each OTG_HS peripheral
- SOF signal output, dedicated DMA for each OTG_HS



The STM32H7 microcontroller embeds two instances of a USB HS peripheral (OTG_HS1 and OTG_HS2). Both support Full-Speed communication and OTG mode but only the OTG_HS1 peripheral supports High-Speed communication through an external HS PHY thanks to its ULPI interface.

OTG_HS peripheral

6

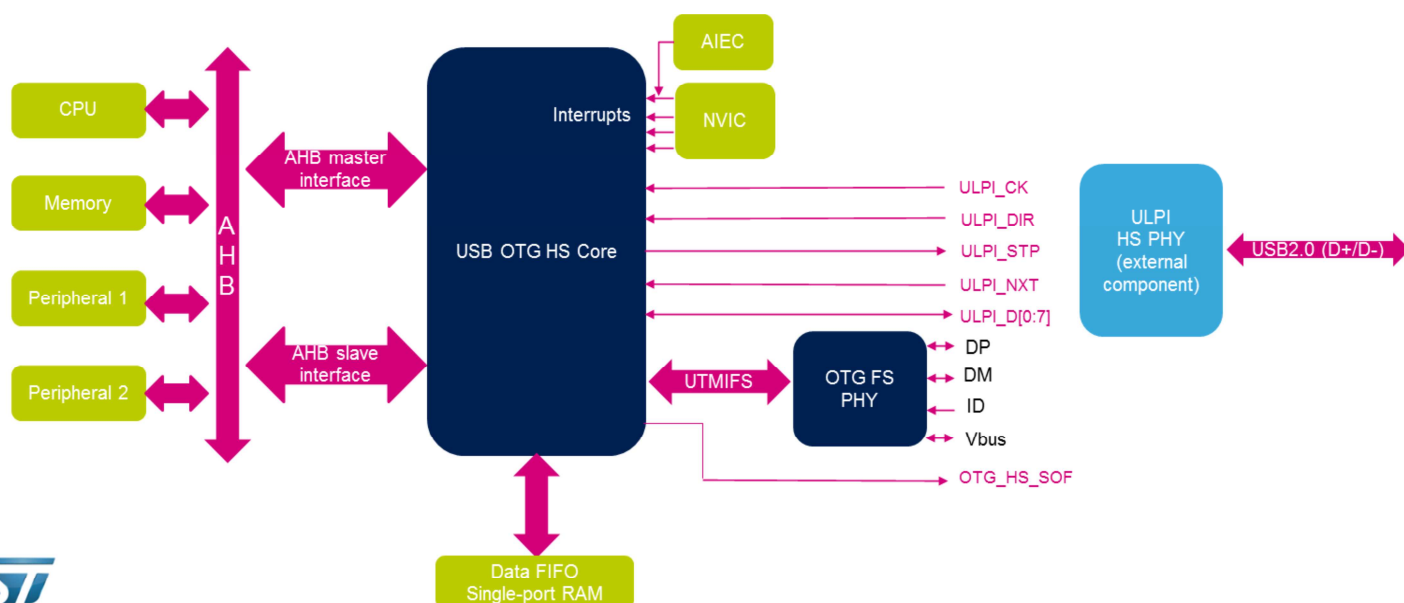
USB features	OTG_HS1	OTG_HS2
Device bidirectional endpoints (including EP0)	9	
Host mode channels	16	
Size of dedicated SRAM	4 Kbytes	
USB 2.0 Link Power Management (LPM) support	Available	
OTG revision supported	2.0	
Attach Detection Protocol (ADP) support	Not supported	
Battery Charging Detection (BCD) support	Available	
ULPI available to primary IOs via multiplexing	Available	Not supported



This is an overview of the peripheral's characteristics. Up to 9 channels (including endpoint 0) can be used for device implementation, which can be useful for creating more complex composite devices. On the host side, up to 16 channels can be used in parallel. As already highlighted, both instances have a dedicated 4-Kbyte RAM for FIFO, support Link Power Management, OTG mode and Battery Charging management. But only OTG_HS1 can connect an external USB PHY using the ULPI hardware interface to work in High Speed mode.

Block diagram (OTG_HS1)

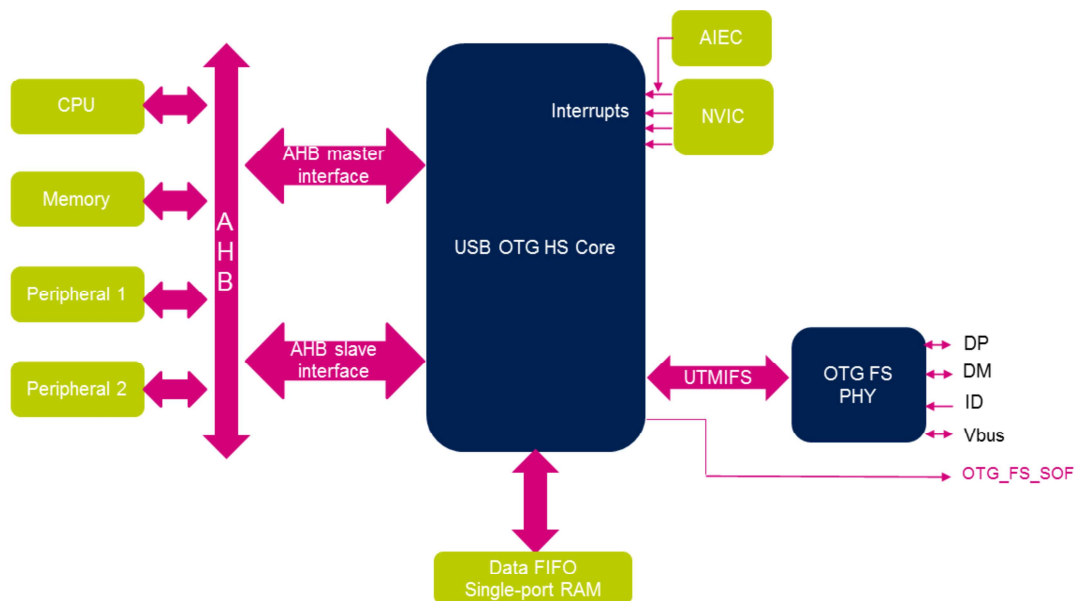
7



In this block diagram, the USB OTG High Speed controller core (“HS1” instance) is shown in the center with its data FIFOs below. The FS Physical Layer, or PHY, on its right side handles the analog signal levels including many specific level detections relating to On-The-Go and Battery Charger Detection functions. For High-Speed operation, an external ULPI PHY transceiver can be connected to the HS2 core. The USB interrupt goes to the Cortex processor to signal various USB events. The AHB slave interface enables read/write access of the controller registers and the Power& Clock control block. Transfers to and from memory are handled by a DMA engine inside the controller via the AHB master interface.

Block diagram (OTG_HS2, FS only)

8

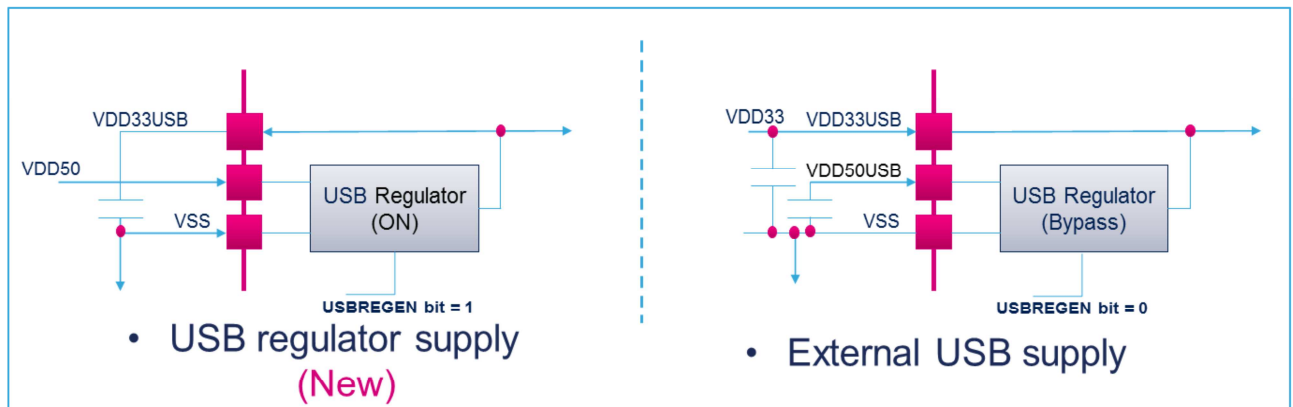


In this block diagram, the USB OTG High Speed controller core (“HS2” instance) is shown in the center with its data FIFOs below. The FS PHY, on its right side handles the analog signal levels including many specific level detections relating to On-The-Go and Battery Charger detection functions. Note that the connectivity of the HS2 core only allows Full Speed operation. The USB interrupt goes to the Cortex processor to signal various USB events. The AHB peripheral bus enables read/write access of the controller registers and the Power& Clock control block. Depending on the use case (i.e. either device only or OTG device), a low- or high-speed crystal oscillator is necessary to provide an accurate timing reference for the USB block.

USB supply configurations

9

- USB regulator: New feature added in STM32H7x3
 - Possibility to source OTG_HS peripheral from 5V DC source (VBUS) while application works with different power source (and possible different voltage)
 - No external part can be sourced from USB regulator



The voltage regulator dedicated for OTG peripherals is a new feature of the STM32H7 family. Using this regulator, only an OTG peripheral can be sourced; other internal or external functions cannot be connected to pin VDD33USB. This allows to power the peripheral directly from VBUS without any additional components and use different voltage level for MCU VDD. Set bit USBREGEN to 0 when a USB peripheral needs to be supplied with an external 3V3 voltage source, as on older devices.

- For USB application, the internal clock source is not precise enough
- USB host and HS device application requires HSE usage
- For USB FS, the device can use CRS (clock recovery system) or HSE
 - Using CRS allows a crystal-less USB device design
- CRS enables internal clock source synchronization from:
 - LSE oscillator
 - USBx SOF packet reception (any OTG peripheral acting as device can be used)



To achieve a High-Speed communication, a high-frequency clock is required. So the USB clock is sourced from the HSE clock.

For Full-Speed communication, a crystal-less design can be implemented based on the clock recovery system (CRS).

Note that this clock recover system is only relevant for the case of a Full-Speed device.

- Peripheral (device) mode function:
 - Device only or OTG device when operating in Peripheral mode
 - Internal settable 1k5 ohms resistor on D+ line
 - Soft disconnect feature support
 - 1 bidirectional control endpoint0
 - 8 IN and 8 OUT endpoints configurable to support Bulk, Interrupt or Isochronous transfers
 - Management of a shared Rx FIFO and up to 9 dedicated Tx-OUT FIFOs



At any given time, one of the two operating modes (host or device modes) is functional.

Peripheral mode is used for a regular device or an OTG device when operating in Device mode.

1.5 Kohm pull resistor on the D+ line can be used to show the presence of the device on the bus. One bidirectional control endpoint 0 is available for application implementation. Another 8 IN and 8 OUT endpoints can then be set on the fly. A dedicated 4-Kbyte RAM can be divided into 1 shared RxFIFO and up to 9 TxFIFOs, 1 one for each out endpoint.

- Host mode function:
 - Host only or OTG device when operating in Host mode
 - External charge pump for VBUS voltage generation
 - Up to 16 host channels (pipes): each channel is dynamically reconfigurable to allocate any type of USB transfer.
 - Built-in hardware scheduler holding:
 - Up to 16 interrupt plus isochronous transfer requests in the periodic hardware queue
 - Up to 16 control plus bulk transfer requests in the non-periodic hardware queue
 - Management of a shared Rx FIFO, a periodic Tx FIFO and a nonperiodic Tx FIFO for efficient usage of the USB data RAM.
 - External charge pump required



Targeted Host mode is used for an embedded host or an OTG device when operating in Host mode.

In Host mode, it is necessary to use an external charge pump to drive the VBUS voltage. An application can use up to 16 host channels (pipes), which can change transfer type on the fly. An embedded hardware scheduler can manage up to 16 periodic and 16 non-periodic requests. Periodic requests are request from interrupts and isochronous channels; non-periodic request are requests from bulk and control channels. FIFO RAM is divided into three shared parts: RxFIFO, TxFIFO for periodic transfers and TxFIFO for non-periodic transfers.

USB class support in STM32 library

13

USB Class	Device support	Host support	Windows driver support
MSC	Yes	Yes	Yes
MTP	Yes	Yes	Yes
CDC	Yes	Yes	From Windows 10 / with ST drivers
HID	Yes	Yes	Yes
AUDIO	Yes	Yes	Yes
DFU	Yes	No	With ST drivers



This slide shows the support of the standard USB classes by the STM32 library.

Low-power modes (OTG_FS)

14

Mode	Description
PHY: power down	Switches on/off the full-speed transceiver module of the PHY.
PHY: VBUS detection	Switches on/off the VBUS sensing comparators associated with OTG operations.
Suspend: Stop PHY clock	Most of the 48 MHz clock domain internal to the OTG full-speed core is switched off by clock gating.
Suspend: Gate HCLK	Most of the system clock domain internal to the OTG full-speed core is switched off by clock gating.
Suspend: USB system stop	Application may decide to drastically reduce the overall power consumption by a complete shutdown of all the clock sources in the system.



Now let's take a brief look at the various low-power modes of the USB Physical Layer (PHY) and the controller.

For the PHY, Power-down mode can be used for example when there is no VBUS present and session is identified to be not OTG ; it is also possible to disable the VBUS sensing related to OTG (A- and B- sessions) if the OTG function is not used.

During Suspend mode, there is no dynamic signaling occurring over the USB interface, so three different controls are offered to lower the power consumption as desired by the application.

Low-power modes (OTG_HS)

15

Mode	Description
Suspend: Gate HCLK	Most of the system clock domain internal to the OTG high-speed core is switched off by clock gating.
Suspend: USB system stop	Application may decide to drastically reduce the overall power consumption by a complete shutdown of all the clock sources in the system.

Low-power modes for the High Speed core are similar to the Full-Speed one, but the modes concerning the PHY are not listed as in this case the PHY (or transceiver) is an external component.

MCU Low-power modes

16

MCU mode	Description	USB availability
Run	MCU fully active.	Required until USB enters Suspend mode.
Sleep	Peripheral interrupts cause the device to exit Sleep mode. Peripheral registers' content is kept.	Available while USB is in Suspend mode.
Stop	Peripheral interrupts cause the device to exit Stop mode. Peripheral registers' content is kept.	Available while USB is in Suspend mode, offers optimal power reduction.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.	Not compatible with USB applications.



The USB peripheral is active in Run and Sleep modes. In Stop mode, the USB is not available but the contents of its registers are kept. In Standby mode, the USB peripheral is powered-down and must be reinitialized when returning to a higher power state.

- A few dedicated bits are implemented to assist debugging of:
 - USB receive data FIFO status/contents (both host and device modes)
 - Host mode: periodic queue scheduling

Debug bit	Description
OTG_FS(/HS) Receive status debug read / OTG status read and pop registers	Special read FIFO access for debug.
Top of the periodic transmit request queue	Indicates the entry in the periodic Tx request queue that is currently being processed by the MAC.



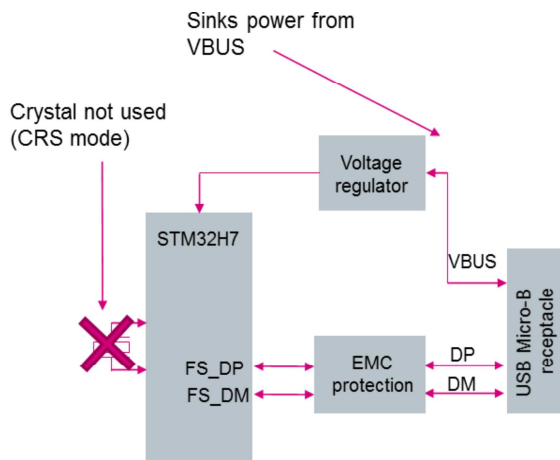
Within the USB module, certain dedicated bits are implemented to allow debug functionality in a USB application.

They relate to FIFO status and contents and the scheduling of periodic queues in Host mode.

Additional details of these debug bits are listed in this table.

Application: low-power device

18



- The schematic shows an example of a low-power design for a “device only” application
- In this example, the device is bus-powered, drawing power only from USB VBUS
- This crystal-less design uses the Clock Recovery System (CRS) to source a precise USB clock

Here is an application example of a low-power device. Power is drawn directly from the USB VBUS signal. To get a precise-enough clock signal for high-speed communication, the USB clock is sourced from the clock recovery system without the need for an external resonator component enabling a crystal-less design for such applications.

- For more details, please refer to the following source pages:
 - www.usb.org:usb20_docs (hosts a ZIP file containing):
 - USB2.0 specification
 - On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification (USB2), latest version 1.1a
 - USB 2.0 ECN: Link Power Management Addendum
 - www.usb.org:devclass_docs
 - Battery Charger v1.2 specification



For complete USB specification documents, please refer to USB.org.

USB2.0 document home page has a ZIP file containing the USB2.0 and OTG2.0 specifications and an ECN for LPM

The USB device class documents page has the battery charger specification.