Application note

STM32MP151/153/157 MPU lines and STPMIC1B integration on a battery powered application

Introduction

This application note applies to the STM32MP151/153/157 MPU line devices and are now referred to as STM32MP15x.

This application note provides a sample hardware reference design based on STM32MP15x and STPMIC1B power management IC, powered from a single cell Li-Ion / Li-Po battery.

This document is intended for product architects and designers who require information on hardware integration and settings, focusing on:

- Reference design block diagram
- Power distribution
- Start up, shutdown and low-power management
- Battery management (USB charging and monitoring overview)
- USB high speed port management.

1 General information

This document applies to the STM32MP15x Lines dual-core Arm®-based Series microprocesor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Overview

This application note covers the STM32MP15x together with the STPMIC1B with DDR and Flash memory and together with the following peripherals:

- DC input power source from 1-Cell Li-Ion / Li-Po: 3.6 V nominal (2.8 V to 4.3 V).
- Peripheral I/O interface voltage at 1.8 V supplied by the STPMIC1B.
- lpDDR2 or lpDDR3 with 32-bit wide bus.
- eMMC Flash memory powered from a 2.9 V power source and 1.8 V I/O interface.
- USB high-speed port supporting power sink mode (for battery charging capability) and source mode (to supply a USB device).
- Battery charging and battery monitoring is introduced for illustration.

Out of scope this application note:

- DDR3 and DDR3L: these memory devices are not appropriate for use with battery powered products due to their high power consumption.
- Peripheral interface with an I/O voltage of 3.3 V: the assumption is that the 1-cell Li-Ion / Li-PO battery has still enough energy when battery voltage drops below 3.3 V. If an I/O peripheral such as the STPMIC1A is used, it no longer operates under these circumstances.

In this document, MPU terminology refers to the STM32MP15x and the PMIC terminology is referring to STPMIC1B device.

2.1 Reference documents

Table 1. Reference documents

Document number	Title
STMicroelectronics documents [1]	
[1]	*Getting started with STM32MP1 Series hardware development* (AN5031)
[2]	*Highly integrated power management IC for micro processor units* (DS12792)
[3]	*STM32MP1 Series using low-power modes* (AN5109)
[4]	*STM32MP151/153/157 Lines and STPMIC1A hardware and software integration* (AN5089)
[5]	*STM32MP157 advanced Arm®-based 32-bit MPUs* (RM0436)
[6]	*Arm® dual Cortex®-A7 650 MHz + Cortex®-M4 MPU, 3D GPU, TFT/DSI, 37 comm. interfaces, 29 timers, adv. analog, crypto* (DS12505)
[7]	*Standalone USB Type-C™ controller with high voltage protections* (DS11503)
USB specification	
[8]	*USB Type-C cable and connector specification* release 1.4 or later available from USB implementation forum web site

1. *Refer to www.st.com*

3 Glossary

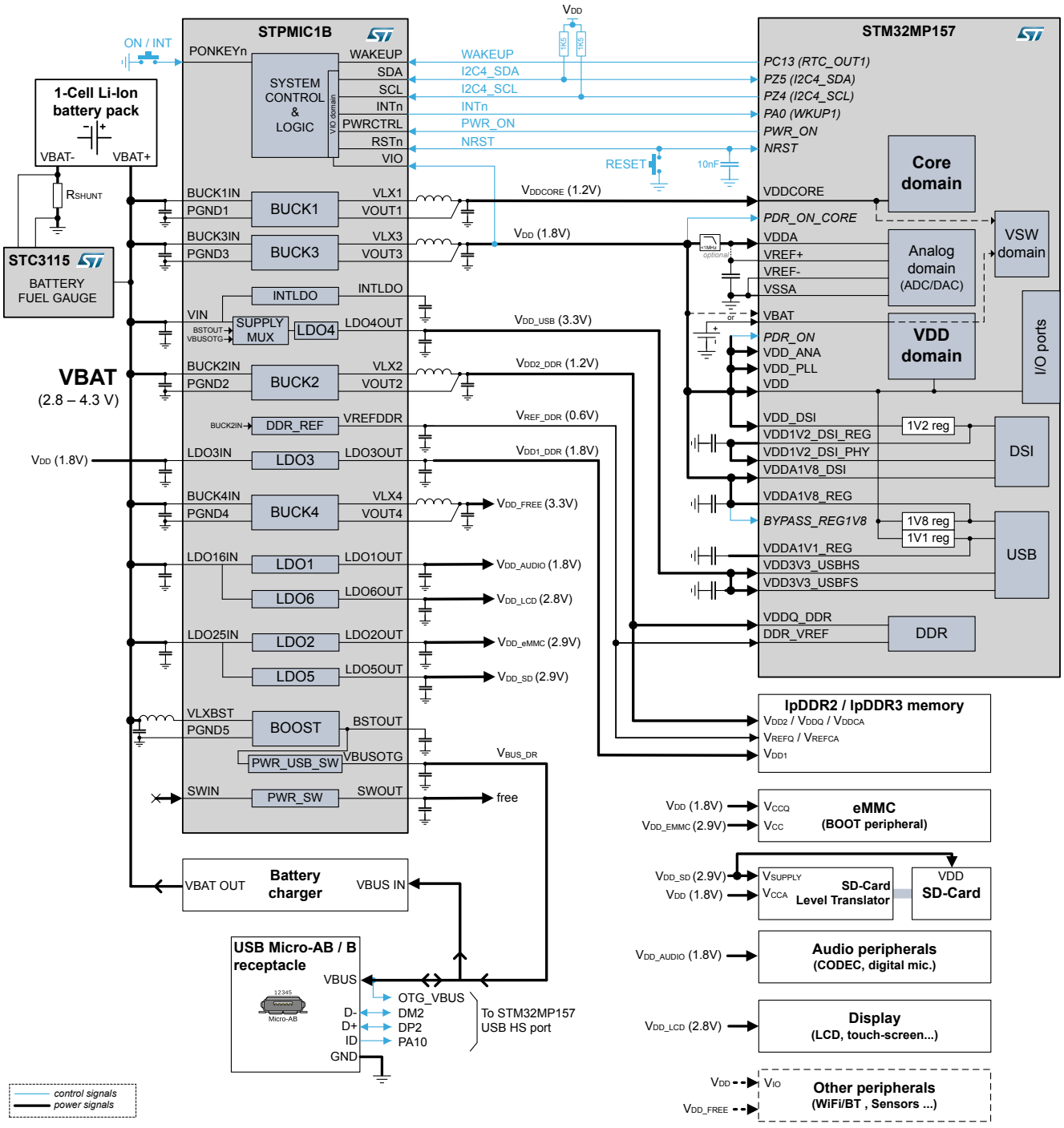
Table 2. **Glossary**

Term	Definition
DR	Dual-role. In this document, a peripheral supporting USB host mode or USB device mode.
FSBL	First stage boot loader
HSI	High speed internal oscillator
IC	Integrated circuit
LDO	Low drop out linear regulator
MPU	Microprocessor unit and refers to the STM32MP15x in this document
PMIC	Power management integrated circuit and stands for the STPMIC1B in this document
RTC	Real-time clock
SMPS	Switching mode power supply
SW	Software
TAMP	Tamper detection circuit

4 Battery application reference design

This reference design targets an application powered from a rechargeable battery with low power lpDDR2 or lpDDR3, an eMMC, an SD-card, a USB HS dual role data port (also known as OTG). The USB HS dual role data port is also used as a power source to charge the battery. A battery fuel gauge monitors the battery energy level. Other peripherals like audio, display, Wi-Fi® / Bluetooth® sensors are added to illustrate the application. The main peripheral interfaces work with an IO voltage of 1.8 V. The overall system is illustrated in Figure 1.

Figure 1. STM32MP15x and STPMIC1B with lpDDR2/3, eMMC, SD-card and USB DR



Note: The following are not shown in the diagram:

- *STM32MP15x decoupling scheme (see [1] related chapter),*
- *STPMIC1B discrete components value (see [2]),*
- *Additional protection, such as ESD, EMI filtering, overvoltage.*

4.1 Battery as main power source

This application is powered by a single cell rechargeable 3.6 V nominal voltage Li-Ion or Li-PO battery pack (VBAT) with a range of 2.8 V to 4.3 V. The battery capacity of the reference design in Figure 1 depends on the application use cases and the peripheral power consumptions. For instance, assuming a 500 mAh – 1000 mAh battery pack is used.

Note: In this document, the STM32MP15x VBAT pin (used for RTC/TAMP supply) must not be confused with the battery voltage domain VBAT that supply the application.

The battery pack is used as main power source (VBAT) to supply the STPMIC1B. The STPMIC1B has all the required power converters to supply the complete application.

The battery voltage (VBAT) is monitored by two STPMIC1B thresholds:

- VINOK_rise : To prevent the STPMIC1B from starting when the battery voltage (VBAT) is too low.

Note: VINOK_rise: STPMIC1B threshold = 3.3V nominal.

- VINOK_fall : To force the STPMIC1B to power down to prevent battery fully discharging (dead battery).

Note: VINOK_fall: STPMIC1B threshold = 2.8V nominal.

Note: VINOK_rise and VINOK_fall thresholds can be adjusted by re-programming the STPMIC1B NVM via I^2C.

The STC3115 fuel gauge monitors the battery voltage and acts as current sensor to accurately estimate the remaining battery charge level. Additionally, the STC3115 also monitors the battery temperature if the battery pack is located close to the STC3115.

The battery charge process is managed by a discrete battery charger IC. Charge energy is provided by the USB connector (Type Micro-AB or Micro-B). The charger IC converts the USB voltage (VBUS) to battery voltage (VBAT).

4.2 Power distribution

The STPMIC1B integrates the regulators required to supply all the STM32MP1 Series devices in addition to a set of regulators to supply the application peripherals.

4.2.1 VDD power domain (1.8 V)

VDD is the reference design's main IO voltage domain used by the STM32MP15x, STPMIC1B and peripherals.

It is powered from the STPMIC1B BUCK 3 step-down SMPS which has high efficiency and low quiescent current across all load conditions.

The VDD voltage domain is the first voltage available during power up sequence (STPMIC1B Rank 1) and the last disabled during power down sequence.

VDD is enabled in Run, LP-stop, LPLV-stop and Standby modes. VDD is disabled in Off mode.

STPMIC1B has dedicated NVM settings to set BUCK 3 (IOs) at 1.8 V during the STPMIC1B power up to fit with battery reference design in Figure 1. See [2] for details.

STM32MP15x VDD power supply:

Connect VDD_PLL, VDD_ANA, VDD_DSI and VDDA1V8_DSI to VDD in addition to supplying the STM32MP1 Series microprocessor VDD IO voltage domain.

ADC/DAC analogue voltage (VDDA) and related analogue reference voltage (VREF+) may be powered from VDD depending on expected ADC performances. If high ADC performances are expected or to ensure the DAC functions correctly, power VDDA (and VREF+) from a low noise power source with a voltage greater than 1.8 V. In that case, a STPMIC1B LDO may be dedicated to supply the STM32MP15x VDDA voltage domain.

4.2.2 VDDCORE power domain (1.2 V)

VDDCORE is the main STM32MP15x digital power domain.

It is powered from the STPMIC1B BUCK1 step-down SMPS. This voltage domain is the next power domain to be available during power up sequence (STPMIC1B Rank 2) and the penultimate to be disabled during power down sequence.

VDDCORE is enabled in Run, LP-Stop and LPLV-Stop modes. VDDCORE is reduced in LPLV-Stop to save power. VDDCORE is disabled in Standby and in Off mode.

STPMIC1B has NVM settings to set BUCK1 (VDDCORE) to 1.2 V when STPMIC1B is in power up.

4.2.3 VDD_USB power domain (3.3 V)

VDD_USB is dedicated to supplying power to the STM32MP15x USB PHYs (VDD3V3_USBHS and VDD3V3_USBFS).

It is powered from the STPMIC1B LDO4 linear regulator, which has been specifically designed for this feature.

LDO4 has a 3 power input source multiplexor which selects automatically the highest input voltage. It is designed specially to keep the STM32MP15x USB PHY working in following uses cases:

- USB embedded host with low battery voltage (VBAT <= 3.3 V): A USB peripheral (eg: mass storage device) is connected to the application and is powered by VBUS_DR (5.2 V) from STPMIC1B boost converter via PWR_USB_SW (see Figure 1). In this case, LDO4 cannot generate VDD_USB at 3.3 V from battery as VBAT is below 3.3 V. The LDO4 input source multiplexor automatically sets BSTOUT (BOOST converter providing a 5.2 V voltage) as LDO4 input source to generate VDD_USB at 3.3 V.

Note: Ignoring LDO4 dropout voltage for illustration.

- USB peripheral (device) with low battery voltage (VBAT <= 3.3 V): A USB host such as a standard downstream port from a PC is connected to the application and it provides VBUS (5 V) to VBUS_DR (see Figure 1). The LDO4 cannot generate VDD_USB at 3.3 V from battery as VBAT is below 3.3 V. The LDO4 input source multiplexor automatically sets VBUSOTG (from VBUS_DR) as LDO4 input source (5 V) to generate VDD_USB at 3.3 V.

Note: STPMIC1B's LDO4 mux input sources: VBAT (via VIN pin) / BOOST converter (via BSTOUT pin) / VBUS_DR (via VBUSOTG pin).

This voltage domain is the last domain available during power up sequence (STPMIC1B Rank 3) and the first disabled during power down sequence (except regulators enabled by software that are disabled before LDO4 in Rank 0. See [2] for details).

VDD_USB is enabled in Run, LP-Stop and LPLV-Stop modes if a USB peripheral is connected to the application. It can be disabled if no USB peripheral is connected. VDD_USB is disabled in Standby and in Off mode.

STPMIC1B has NVM settings to set LDO4 (VDD_USB) to 3.3 V when STPMIC1B powers up. LDO4 (VDD_USB) is needed at power up to supply the USB PHY for the USB flashing use case (STM32MP15x peripheral boot from ROM).

To save power at power up, re-programme the STPMIC1B NVM to automatically enable LDO4 only when VBUS is present on VBUS_DR at power-up.

4.2.4 VDD1_DDR (1.8 V), VDD2_DDR (1.2 V), VREF_DDR (0.6 V) power domains

- VDD1_DDR is dedicated for lpDDR2 or lpDDR3 core power supply 1 (VDD1) at 1.8 V.
- VDD2_DDR is dedicated for lpDDR2 or lpDDR3 core power supply 2 (VDD2 / VDDQ / VDDCA) and for STM32MP15x DDR IOs supply (VDDQ_DDR) at 1.2 V.
- VREF_DDR is dedicated for lpDDR2 or lpDDR3 reference voltage (VREFQ / VREFCA) and for STM32MP15x DDR reference voltage (DDR_VREF) at VDD2_DDR / 2 (0.6 V).
- VDD1_DDR (1.8 V) is powered from the STPMIC1B LDO3. To optimize power efficiency, LDO3 is powered from VDD (LDO3IN = VDD = 1.8 V). LDO3 has been designed to support Bypass mode (like a power switch) with low RDSon to fit lpDDR2 / lpDDR3 VDD1 voltage tolerance .

Note: Taking into account both BUCK3 (VDD) voltage tolerance and LDO3's RDSon tolerance to fit lpDDR's VDD1.

- VDD2_DDR (1.2 V) is powered from the STPMIC1B BUCK2 step down SMPS with high efficiency and low quiescent current across all load conditions. BUCK2 is powered from battery voltage (VBAT).
- VREF_DDR (0.6 V) is powered from the STPMIC1B's REFDDR sink/source LDO. When enabled REFDDR output voltage is equal to VDD2_DDR / 2 (BUCK2 output voltage / 2).

The STPMIC1B does not start VDD1_DDR, VDD2_DDR and VREF_DDR at power up. They must be powered up and powered down by STM32MP15x software, respectively at STM32MP15x boot up and shutdown.

Software lpDDR2 / lpDDR3 power up sequence:

After power up, when the STM32MP15x is booting up, its software must enable VDD1_DDR, VDD2_DDR and VREF_DDR voltage domains in the following order to comply with lpDDR2 / lpDDR3 power sequence (see JESD209-2B chapter 3.4 or latest JEDEC revision):

1. Enable LDO3 in Bypass mode: VDD1_DDR rises to 1.8 V (= VDD voltage) in ~100 µs.
2. After 100 µs, enable REFDDR LDO: VREF_DDR voltage is kept at 0 V as BUCK2 is disabled.
3. Set BUCK2 to 1.2 V and enable BUCK2.

Note: Step 1 to step 3 must be performed in less than 20 ms to comply lpDDR2 / lpDDR3 JEDEC constraints.

Software lpDDR2 / lpDDR3 power down sequence:

Before the STM32MP15x software goes in Off mode or in Standby mode (assuming the software policy is to turn the DDR off in Standby mode), the STM32MP15x software must manage the lpDDR2 / lpDDR3 power down in the following order:

1. Software receives an event to go into Off or Standby mode.
2. Assert lpDDR CKE low.
3. Disable BUCK2:
   a. VDD2_DDR voltage falls in less than 1.5 ms (thanks to the BUCK2 Slow Pull-Down discharge resistor).
   b. VREF_DDR voltage follows VDD2_DDR / 2 (REFDDR is push-pull LDO).
4. Wait 1.5 ms, disable REFDDR (VREF_DDR is already 0 V as voltage falls in previous step).
5. Disable LDO3: VDD1_DDR falls in less than 3 ms (thanks to LDO3's pull-down discharge resistor).

Note: Step 3 to step 5 must be performed in less than 20 ms to comply lpDDR2 / lpDDR3 JEDEC constraints.

lpDDR2 / lpDDR3 uncontrolled power-off sequence:

Uncontrolled power off sequence occurs typically on battery voltage removal or when a reset occurs. In that case, STPMIC1B manages power off sequence:

1. Battery is removed: VBAT drop.
2. VBAT crosses the VINOK_fall : STPMIC1B turn off condition.
3. STPMIC1B starts a power off sequence:
   a. STPMIC1B assert reset (NRST) → STM32MP15x asserts DDR_CKE pin low and DDR_CLKP / DDR_CLKN clock signals are stopped .

Note: As soon as STPMIC1B asserts the NRST, the reset signal is propagated in the STM32MP15x: DDR_CKE is set LOW, DDR_CKP and DDR_CKN are stopped: lpDDR IC power consumption drops immediately.

   b. STPMIC1B disables Rank0 regulators: BUCK2, LDO3 and REFDDR:
      i. Pull-down discharge resistors are set on BUCK2, LDO3, REFDDR.
      ii. VDD1_DDR, VDD2_DDR, VREF_DDR voltage drops. VDD2_DDR drops faster because a pull-down discharge resistor is stronger than VDD1_DDR and VREF_DDR ones.

Note: BUCK2, LDO3 and REFDDR are disabled by STPMIC1B immediately after reset occurs (Rank0) as those regulators are enabled last (by software) after power up. See [2] for details.

4. STPMIC1B disables Rank3, then Rank2 and finally Rank1 regulators (see [2] for details)
5. STPMIC1B is off (or no supply).

See Section 5.2.3 for details and limitations.

4.2.5 VDD_eMMC power domain (2.9 V)

VDD_eMMC is dedicated to power supply eMMC flash memory core domain (VCC). The eMMC Flash memory device has also to be powered from VDD voltage to supply its IOs power domain (VCCQ) .

Note: Pull-up resistors on VDD are required on CLK, CMD and D0 to avoid extra eMMC power consumption on VDDQ when VDD_eMMC is OFF.

VDD_eMMC is powered from the STPMIC1B LDO2 linear regulator. This voltage domain is the second domain to be enabled during the power up sequence (STPMIC1B Rank2) and it is the penultimate domain to be disabled during the power down sequence.

VDD_eMMC is enabled in Run, LP-Stop and LPLV-Stop modes if STM32MP15x software requires a R/W access. Software must be disabled VDD_eMMC when no R/W access is expected. VDD_eMMC is disabled in STANDBY and in Off mode.

STPMIC1B has NVM settings to set LDO2 (VDD_eMMC) to 2.9 V when the STPMIC1B is powered up. LDO2 (VDD_eMMC) is needed at power up to supply eMMC Flash device allowing the STM32MP15x to access this memory from the ROM to boot up.

VDD_eMMC can be either switched on or off at runtime by the application software.

If the eMMC device is the boot Flash peripheral, the application software must program the STPMIC1B in order to power off the eMMC in Standby mode (PWR_ON signal low) and power on the eMMC in Run mode (PWR_ON signal high) before the application goes into Standby mode. In such case, when the application recovers from Standby mode to Run mode, the eMMC is powered up and ready to be accessed by the STM32MP15x bootROM (peripheral boot).

4.2.6 VDD_SD power domain (2.9 V)

VDD_SD is dedicated to powering an SD-card device and an SD-card level shifter (VSUPPLY). The SD-card level shifter device is also to be powered from VDD voltage to supply its IOs power domain (VCCA).

It is powered from the STPMIC1B LDO5 linear regulator.

This voltage domain is the second domain to be enabled during the power up sequence (STPMIC1B Rank2) and it is the penultimate to be disabled during power down sequence.

VDD_SD is enabled in Run, LP-stop and LPLV-stop modes if the STM32MP15x software requires a R/W access. Software must disable VDD_SD at any time if no R/W access expected. VDD_SD is disabled in Standby and in Off modes.

STPMIC1B has NVM settings to set LDO5 (VDD_SD) to 2.9 V when the STPMIC1B powers up. LDO5 (VDD_SD) is needed at power up to supply an SD-card and its level shifter IC allowing the STM32MP15x to access this memory from the ROM to boot up.

VDD_SD can be either switched on or off at runtime by the application software.

If the SD-card device is the boot flash peripheral, the application software must program the STPMIC1B in order to power off the SD-card in standby mode (PWR_ON signal low) and power on the SD-card in Run mode (PWR_ON signal high) before the application goes into standby mode. In such case, when the application recovers from standby mode to run mode, the SD-card is powered and ready to be accessed by the STM32MP15x bootROM (peripheral boot).

4.2.7 VBUS_DR power domain (5 V)

VBUS_DR is a dedicated power domain for the USB High Speed interface. VBUS_DR is connected to the VBUS pin of the USB receptacle and has dual direction current:

- Application in USB device mode: a USB host peripheral is connected to the application and provides VBUS_DR to the application (see Figure 2).
- Application in USB host mode: a USB device peripheral is connected to the application and powered from VBUS_DR by the application (see Figure 3).

Figure 2. **VBUS_DR power path in USB device mode**

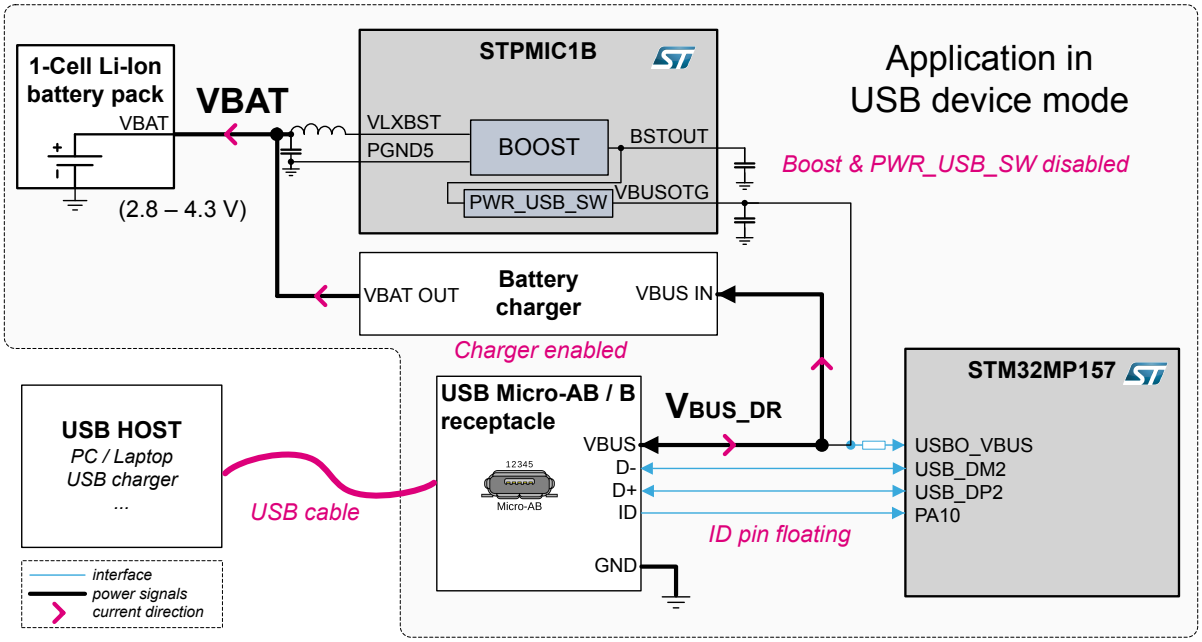
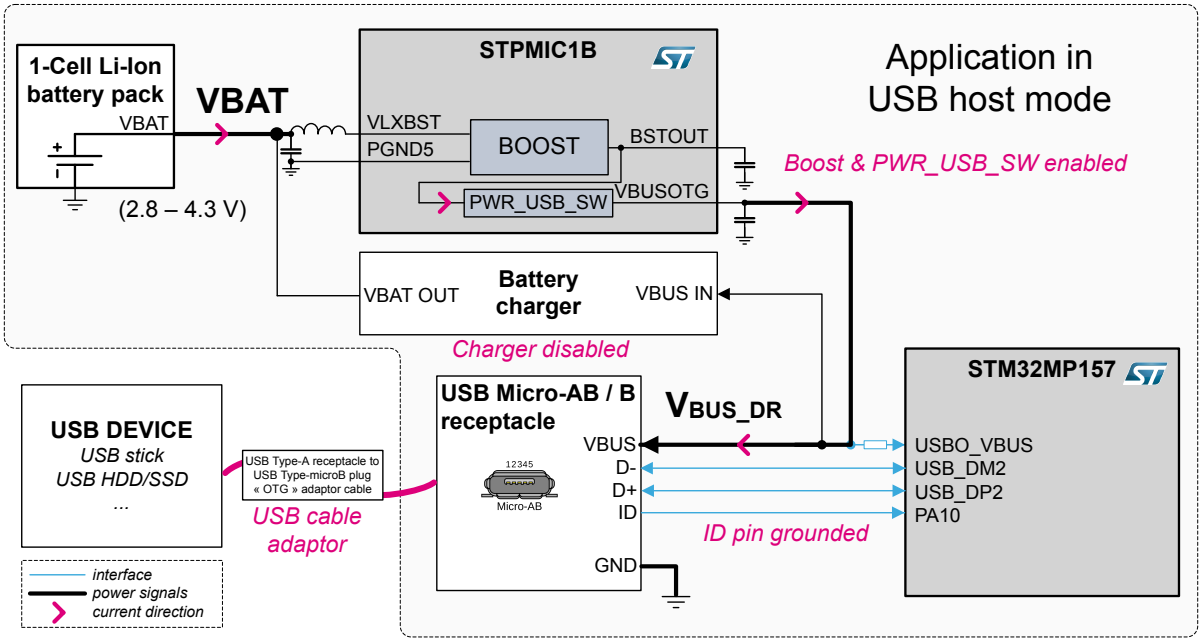


Figure 3. **VBUS_DRD power path in USB host mode**



4.2.8 VDD_AUDIO (1.8 V), VDD_LCD (2.8 V) power domains

VDD_AUDIO and VDD_LCD are provided to illustrate the reference design in Figure 1.They are respectively powered from LDO1 and LDO6 linear regulators.

VDD_AUDIO and VDD_LCD are not enabled by the STPMIC1B at power up. They are enabled and set at the right voltage after power up by software when the related peripheral requires them.

VDD_AUDIO and VDD_LCD may be enabled in Run, LP-stop and LPLV-stop modes when the STM32MP15x software needs to perform an access to the related peripherals. Software can disable VDD_AUDIO and VDD_LCD at any time if no peripheral access is expected. VDD_AUDIO and VDD_LCD are disabled in Standby and in Off modes.

4.3 Control signals and interface between STM32MP157 and STPMIC1B

This sections outlines the way the STM32MP157 microprocessor communicates with the STPMIC1B device. There are several interface choices which can be used depending on the application requirements. Each interface is described in this first part.

I²C interface:

The STPMIC1B can be controlled by the STM32MP15x via the I²C interface to:

- Enable or disable a regulator
- Set a regulator voltage and mode (low power or high power)
- Set low power management (PWRCTRL behavior)
- Set the interrupt controller or read interrupt status
- Set the protection (watchdog, overcurrent, under-voltage) or read protection status.
- Reprogram the NVM to change the startup behavior.

Note: The STPMIC1B has special default NVM settings that allows it to boot an STM32MP15x application with 1.8V IOs from the USB interface (for flashing or loading then executing software) from SD-card or from Flash memory (such as an eMMC). Once the STM32MP15x is able to execute software, it is also able to reprogram the STPMIC1B NVM on the fly to fine tune the application.

ON / INT push button:

The user “ON / INT” push button is connected to the STPMIC1B PONKEYn pin (active low). This button allows:

- To power up the STPMIC1B.
- To send an interrupt to the STM32MP15x on a button press event or a button release event when the application is operating.
- To force a power off of the STPMIC1B with a long press (16s by default).

NRST signal:

The NRST is a bidirectional active low signal for the STM32MP15x and the STPMIC1B. The STM32MP15x NRST pin and STPMIC1B RSTn pin are of digital input / open drain output topology:

- When STPMIC1B asserts RSTn (such as during the power up or the power down sequence), it drives the NRST signal low: the STM32MP15x is forced into a reset state until STPMIC1B releases the NRST.
- When the STM32MP15x asserts an NRST signal (such as an STM32MP15x watchdog reset) or a user preses on the “RESET” button, STPMIC1B immediately asserts the RSTn pin and performs a non-interruptible power-cycle: the STPMIC1B performs a power down sequence followed by a power up sequence and finally releases the RSTn .

Note: At the end of power-cycle sequence, STPMIC1B waits for the NRST signal to go high before rearming the reset to avoid infinite reset loop.

INTn signal:

The INTn is a STPMIC1B output active low interrupt line connected to the STM32MP15x PA0 input pin. PA0 has both interrupt and wakeup capability:

- To manage interrupt from the STPMIC1B when the STM32MP15x is in either Run or Stop mode.
- To wake-up the STM32MP15x when it is in Standby mode.

PWR_ON signal:

The PWR_ON signal is driven by the STM32MP15x PWR_ON pin to control STPMIC1B PWRCTRL pin.

This allows the STM32MP15x to switch the STPMIC1B power strategy very quickly to one of the following application power modes:

- From Run mode to LPLV_Stop mode and back
- From Run mode to STANDBY mode and back.

(See [3] for details about low power mode management and PWR_ON pin setting when using STPMIC1B)

After a power-up or a reset, the STPMIC1B PWRCTRL pin is disabled. Before going in low power mode, the STM32MP15x sets the STPMIC1B via I²C to program the expected power behavior according to the PWR_ON signal state.

WAKEUP signal (optional):

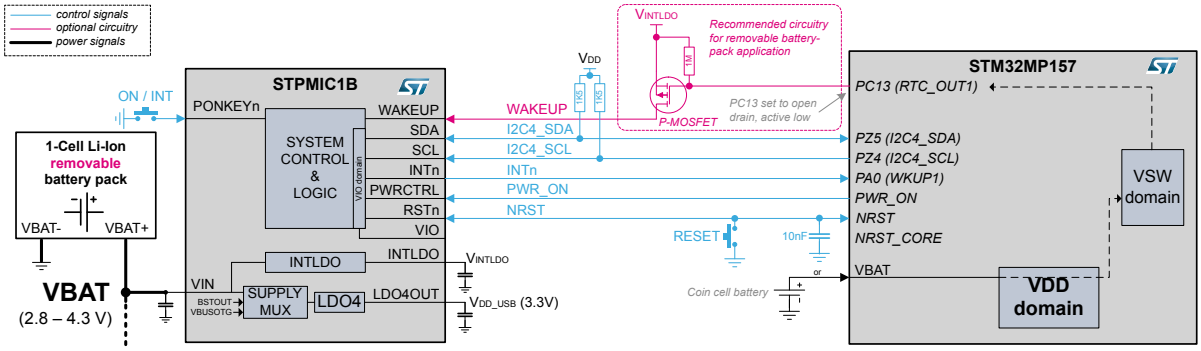
The WAKEUP signal is driven by the STM32MP15x PC13 (RTC_OUT) pin to control the STPMIC1B WAKEUP pin. It allows the STM32MP15x to power up the STPMIC1B; typically, when real time clock timer elapses.

This feature is available if a coin cell battery is connected to the STM32MP15x VBAT pin.

4.3.1 WAKEUP pin management with removable battery pack application.

If a coin cell battery is connected to the STM32MP1 Series microprocessor VBAT pin with a product powered by a removable battery pack, the circuit diagram described in Figure 4 must be implemented.

Figure 4. WAKEUP pin control circuitry for removable battery pack application



The circuit in Figure 4 isolates the STM32MP1 Series microprocessor PC13 IO voltage from the STPMIC1B WAKEUP pin. This prevents current leakage from the STM32MP1 Series microprocessor PC13 IO:

- When PC13 is set to STPMIC1B VIN via STPMIC1B's WAKEUP pin.
- When battery pack is removed.
- When VBAT voltage lower than PC13 voltage.

When using this circuit implementation (see Figure 4), the STM32MP1 Series microprocessor PC13 must be set as open-drain active low IO. The P-MOSFET must be selected to have a minimum Vgs threshold under 1.8 V (below VINTLDO). The P-MOSFET is able to sustain a drain current (ID) greater than 40 µA to drive the WAKEUP pin.

Note: STPMIC1B WAKEUP pin has internal pull-down resistor (45KΩ < Rpd < 80 KΩ) internally connected to GND.

5 Power management

The following power modes are reviewed:

- Operating modes
- Application Power-up and Power-down modes
- Low power management mode
- User reset and crash recovery management
- Software management examples

5.1 Operating modes

The application can switch to different operating modes depending of the system activity. The operating modes are managed by the STM32MP1 Series microprocessor and they control the power management and the clock distribution (see details in [3]).

Table 3 summarizes the application level operating modes. The STPMIC1B power modes depend on the application Operating mode.

Table 3. Application operating modes

Operating mode	STPMIC1B Power mode	VBAT(1)	PWR_ON	Description	Notes
Run	Power-on-main	> VINOK_fall	1	VDD power on VDDCORE power on, system clock on lpDDR active / auto refresh Peripherals power on / off	(2)
Stop	Power-on-main	> VINOK_fall	1	VDD power on VDDCORE power on, system clock off lpDDR self refresh Peripherals power on / off	(2)
LPLV-stop	Power-on-alternate	> VINOK_fall	0	VDD power on VDDCORE power on at lower voltage, system clock off lpDDR self refresh Peripherals power on / off	(3)
Standby	Power-on-alternate	> VINOK_fall	0	VDD power on VDDCORE power off, system clock off lpDDR self refresh / off Peripherals power off	(3)
Power-off	Off	> VIN_POR_fall	-	All power off	-
	No_supply	< VIN_POR_fall	-	All power off	-
Coin-cell-VBAT	No_supply	< VIN_POR_fall	-	All power off except the STM32MP1 Series microprocessor VSW	(4)

1. *STPMIC1B hardware thresholds. See [2] for details.*
2. *The difference between Run and Stop modes is only based on the STM32MP1 Series microprocessor clock management. For power management, there is no difference between Run and Stop mode.*
3. *There is no difference on the PWR_ON control pin when entering LPLV_Stop mode or Standby mode from Run mode (PWR_ON signal goes from high to low in both cases). But before entering LPLV_Stop mode or Standby mode, the STM32MP1 Series microprocessor programs the STPMIC1B via I²C interface to set the regulators accordingly.*
4. *To retain the content of the STM32MP1 Series microprocessor VSW domain (RTC, backup registers, backup RAM and retention RAM) when VDD is turned off, the STM32MP1 Series microprocessor VBAT pin can be connected to an optional coin cell battery.*

5.1.1 Application turn-on / turn-off conditions

When the application is in Power-off mode, a turn-on condition is required to Power-up the STPMIC1B into Run mode. Similarly , if the application needs to go into Power-off mode, a turn-off condition is required to Power-down the STPMIC1B.

Note: Power-up: STPMIC1B transitional power up state where the regulators start sequentially in a predefined order (rank) and voltage and ends by releasing the NRST signal. After this state, the STPMIC1B goes into Power-ON state and remains there, the application can now be run. This state is reached from Off mode with a Turn-ON condition or from NO_SUPPLY with VBAT voltage rising higher than VINOK_rise (AUTO turn-ON)

Power-down: STPMIC1B transitional power down state where the NRST is asserted leading to the regulators stopping sequentially in the reverse order of Power-up sequence. After this state, the STPMIC1B is in the OFF state and remains as such until a Turn-ON condition occurs. This state is reach from Power-ON state with a Turn-OFF condition.

The STPMIC1B autonomously manages the Power-up and the Power-down sequence when respectively a turn-on or a turn-off condition occurs (see [2] for details).

Turn-on conditions:

The STPMIC1B automatically powers up when the battery voltage (VBAT) rises above VINOK_rise (an AUTO turn-on feature enabled by default in STPMIC1B NVM). If the STPMIC1B is in the off state (and VBAT > VINOK_rise), it can be powered up by one of three external triggers:

- "ON / INT" user button press: PONKEYn pin voltage falling edge.
- USB host or USB charger cable insertion: VBUSOTG pin voltage rising edge.
- The STM32MP15x wakeup event occurs (for example RTC or Tamper wake-up via STM32MP15x PC13 pin): WAKEUP pin voltage rising edge.

Note: The STM32MP157 wakeup feature is available if a coin cell battery is connected to STM32MP15x VBAT pin.

Turn-off conditions:

A turn-off condition leads the STPMIC1B to power down and go into the off state. In the off state, all regulators are turned off. If the STPMIC1B is in the ON state, it can be powered down by one of six conditions:

- Software switch-off: I2C command sent by the STM32MP15x to the STPMIC1B.
- "ON / INT" user button long press: when the reset button is pressed for 16 s, the STPMIC1B is turned off (the delay is programmable).
- Thermal shutdown: if overheating, STPMIC1B shuts down and it restarts when the temperature returns to a correct level.
- Over-current protection: if enabled by software, a over-current on a regulator leads to the STPMIC1B shut down.
- Watchdog: if enabled by software, when the countdown timer reaches 0, the STPMIC1B goes to the off state.
- VINOK_fall: if VBAT goes below VINOK_fall threshold, the STPMIC1B goes to the off state.

Note: An application can set the STPMIC1B "restart request" feature to automatically restart the application after a turn-off condition (see [2] for details).

5.1.2 STPMIC1B restart_request and mask_reset options

Before a turn-off condition occurs, the STM32MP15x software can program the STPMIC1B to restart instead of turning it off by setting the restart_request feature in the STPMIC1B. This setting must be done before initiating the turn-off condition; such as after an application power-up.

For example, the software can completely reboot the application by setting the restart request bit in the STPMIC1B (RREQ_EN = 1) then to program a software switch off (SWOFF = 1). The STPMIC1B performs a power cycle sequence, a power down sequence (disabling all regulators) followed by a power up sequence (restarting regulators then releasing NRST signal).

If the application needs one or several STPMIC1B regulators to be kept enabled during a power cycle, the STM32MP15x software can program the STPMIC1B mask_reset option by setting the STPMIC1B BUCKS_MRST_CR register to target the buck converter and LDOS_MRST_CR register to target LDOs (see [2] for details on the STPMIC1B mask_reset option). This setting must be done before a power cycle, such as after the application power-up.

This is typically the case for the BUCK3 powering the STM32MP15x VDD power domains. The power cycle on VDD must be masked (BUCKS_MRST_CR[2] = 1) to prevent losing:

- The STM32MP15x backup RAM
- Retention RAM
- The backup register content.

If the BUCKS_MRST_CR[2] is not set, this information is lost when a power cycle is triggered by an NRST from the STM32MP15x (see Section 5.4) or by a turn-off condition with the restart_request bit enabled.

5.2 Application power-up / power-down sequence

The power-up sequence is the transition managed by the STPMIC1B between Power-off and Run operating modes and similarly for the power-down sequence. The application power-up and power-down sequence is shown in Figure 5 based on the reference design in Figure 1.

5.2.1 Power-up by battery insertion

When the application has no power. A battery is connected and the application starts automatically when VBAT rises (the STPMIC1B has the auto-turn ON enabled by default in its NVM). When the STPMIC1B is powered-up, the application boots (including the lpDDR initialization) and finally the system reaches Run mode. When a turn-off condition occurs, the STPMIC1B powers-down and goes into the Off mode: the application goes into Power-off mode. The whole process is detailed below and illustrated in Figure 5:

1. Application has no power or the STM32MP15x is in coin-cell-VBAT mode (powered from coin cell battery to supply the STM32MP15x VSW).
2. A well charged battery pack (VBAT > VINOK_rise) is connected to the application. VBAT voltage rises.
3. Once VBAT supply is above VIN_POR_rise (STPMIC1B VIN_POR_rise threshold is initially set to 2.3 V):
   a. The STPMIC1B initializes and pre-loads its NVM contents.
   b. The STPMIC1B asserts the NRST.
4. VBAT supply rises above VINOK_rise, the STPMIC1B checks the Turn-ON condition (auto turn-ON is enabled in the STPMIC1B NVM). The STPMIC1B starts a power up sequence as a valid turn-on condition is detected.
5. The STPMIC1B follows the power-up sequence:
   a. Rank1: BUCK3 (VDD) is enabled at 1.8 V and waits for 3 ms.
   b. Rank2: BUCK1 (VDDCORE) is enabled at 1.2 V. LDO2 (VDD_eMMC) and LDO5 (VDD_SD) are enabled at 2.9 V and waits for 3 ms. The STM32MP15x now performs an internal initialization and releases its reset, the STM32MP15x remains in reset as the STPMIC1B is still asserting its NRST signal.
   c. Rank3: LDO4 (VDD_USB) is enabled at 3.3 V (hard setting). After 3 ms, the STPMIC1B releases the NRST signal.
6. As the NRST signal rises, both the STM32MP15x and the STPMIC1B release their respective reset pins:
   a. The STM32MP15x EADLY delay timer (10 ms) starts.

Note: The EADLY timer prevents the Boot ROM from performing any access to the boot peripheral before it is ready. Typically waiting for a stable voltage on the Flash memory (eMMC or SD-card) to ensure the boot software is reliably read by the Boot ROM. Default delay period after reset is 10 ms. (see [5] for details).

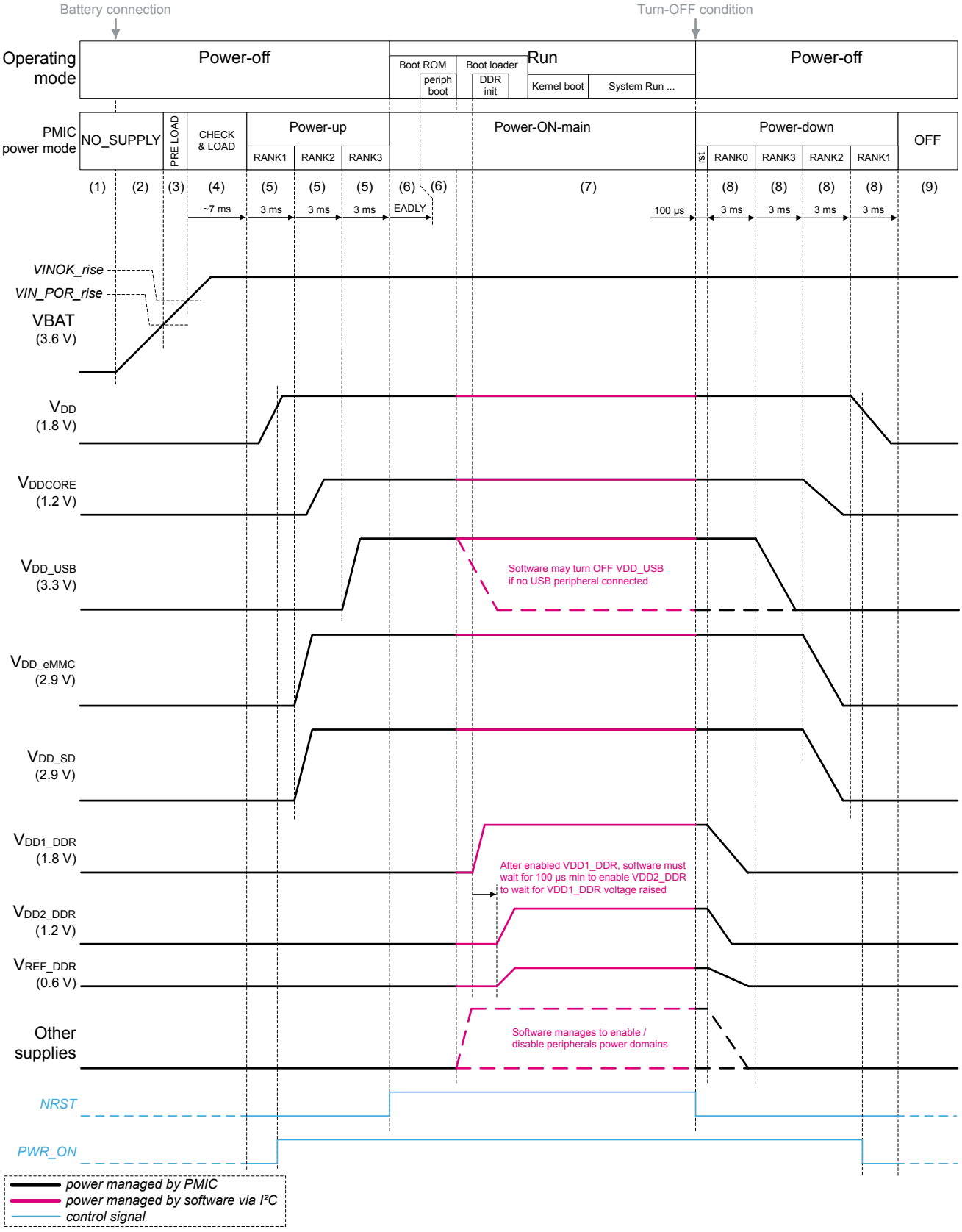
   b. When the EADLY elapses, the boot ROM starts accessing external peripherals (either eMMC or SD-card depending of the STM32MP15x boot pin settings) to load and execute boot loader software.
   c. The Boot loader can control any STPMIC1B regulator (such as initialize an LCD and plot a splash screen).
7. The Boot loader initializes the DDR then loads and executes the Kernel:
   a. LDO3 (VDD1_DDR) is enabled in bypass mode. The software waits at least for a further 100 µs.
   b. DDR_VREF (VREF_DDR) is enabled.
   c. BUCK2 (VDD2_DDR) is enabled at 1.2 V. The software waits for at least 1.4 ms for BUCK2 ready.
   d. The software initializes the STM32MP15x DDR controller and lpDDR device.
   e. the Boot loader loads and executes the kernel. The kernel initializes.
   f. System is now running.

8. A turn-off condition occurs. The STPMIC1B performs a power-down sequence:
   a. The STPMIC1B asserts the NRST (STM32MP15x reset) and waits for 100 µs.
   b. Rank0: the STPMIC1B disables all regulators it has not enabled at power-up (LDO3, BUCK2, DDR_REF, LDO1, LDO6, BOOST, PWR_USB_SW, PWR_SW) and waits for 3 ms.

Note: As soon as a STPMIC1B regulator is disabled, a pull-down resistor is enabled on its output to discharge the decoupling capacitor voltage. The LDO and BUCK regulator output voltages are discharged in 3 ms and 1.5 ms respectively (see [2] for details).

   c. Rank3: LDO4 (VDD_USB) is disabled and waits for 3 ms.
   d. Rank2: BUCK1 (VDDCORE), LDO2 (VDD_eMMC) and LDO5 (VDD_SD) are disabled and wait for 3 ms.
   e. Rank1: BUCK3 (VDD) is disabled waits for 3 ms.
9. The STPMIC1B is now in Off mode: the application is in Power-off.

Figure 5. Power-up / power-down sequence



5.2.2 Power-up from the STPMIC1B Off mode

The application in Figure 5 is powered up from NO_SUPPLY state, where the battery insertion is the turn-on condition (auto turn-on is enabled in the STPMIC1B NVM).

A power-up from the STPMIC1B Off mode follows a similar sequence as in Figure 5 from Off mode. The difference is when a turn-on condition occurs, the sequence starts from step (4) " CHECK & LOAD" instead of waiting for VBAT to rise.

The differences are detailed below:

- The STPMIC1B is initially powered from a VBAT voltage higher than VINOK_rise allowing the STPMIC1B to power-up (instead of a battery insertion which triggers VBAT rise in Figure 5).
- Steps (1), (2), (3) of Figure 5 are replaced by a single one; merging "NO-SUPPLY" and "PRE-LOAD" of the STPMIC1B Power modes to "OFF".
- The "Battery connection" event is replaced by the "Power-ON condition" event and is placed between "OFF" and "CHECK & LOAD" of the STPMIC1B state.

5.2.3 Power-down by battery removal

The application in Figure 5 is powered off by a turn-off condition with VBAT maintaining a valid voltage.

If the application is powered off by a battery removal, the turn-off condition is VBAT dropping below VINOK_fall. Once VBAT supply is below VINOK_fall, the STPMIC1B asserts an NRST for 100µs then powers-down as shown in step (8) onwards in Figure 5.

Limitation: When the battery is removed, VBAT voltage drops very quickly to VINOK_fall value, in less than a few milliseconds (depending on system activity), only then does the power-down sequence start. As soon as the STPMIC1B asserts an NRST, system activity is immediately stopped and power consumption drops, slowing VBAT drop. Nevertheless, VBAT may drop below VIN_POR_fall threshold before the power-down sequence ends. In this case, the STPMIC1B regulators pull-down discharge resistors are no longer controlled by the STPMIC1B. A bulk decoupling capacitor (a few hundred µF) may be inserted on VBAT path to limit VBAT dropping speed.

This sequence is important mainly to manage the "lpDDR2 / lpDDR3 uncontrolled power-off sequence" described in Section 4.2.4 .

5.3 Low power mode management

The STM32MP15x supports several operating modes to reduce power consumption (see Section 5.1). This section describes the LPLV-stop and Standby low-power modes (see [3] for details).

Note: Stop mode concerns the STM32MP15x internal clock management without external power management. So Stop mode is not described in this section.

Low-power modes are managed by the STM32MP15x. The STM32MP15x PWR_ON output pin is connected to the STPMIC1B PWRCTRL input pin. The STPMIC1B states can then be switched: Power-ON-main to Power-ON-alternate and vice versa.

After power-up, the STPMIC1B goes into Power-ON-main until the STM32MP15x tells the STPMIC1B to activate PWRCTRL pin in active low by setting PWRCTRL_POL = 0 and PWRCTRL = 1 in STPMIC1B MAIN_CR register.

Table 3 summarizes the STPMIC1B states according to PWR_ON signal with related operating modes.

When the STPMIC1B goes from Power-ON-main to Power-ON-alternate, it internally switches from the MAIN control registers (xxxx_MAIN_CR) content to the ALTERNATE control registers (xxxx_ALT_CR) content and vice versa (see [2] for details).

Before entering in Low-power mode, the STM32MP15x must set the STPMIC1B ALTERNATE control registers in line with the expected STPMIC1B regulator settings for Low power mode behavior. If needed, the STM32MP15x must set the STPMIC1B MAIN control registers to guarantee that the application leaves the Low power mode.

5.3.1 LPLV-Stop mode

The application LPLV-Stop mode sequence is shown in Figure 6 based on to the implementation shown in Figure 1.

1. The application is powered up and works in Run operating mode; the STPMIC1B is in Power-ON-main state.
2. When the LPLV_Stop operating mode is requested, the software prepares LPLV_ process:
   a. the STM32MP15x settings such as: stoping some clocks, setting DDR to Self-Refresh, setting PWRLP_TEMPO.
   b. STPMIC1B settings:
      - BUCK1 (VDDCORE): 1.2 V HP in main mode, 0.9 V LP in alternate mode
      - BUCK2 (VDD2_DDR), BUCK3 (VDD): ON HP in main mode and ON LP in alternate mode.
3. The STM32MP15x sets the LPDS and LVDS bits of the PWR_CR1 register to wait entering LPLV-: PWR_ON signal is de-asserted when the STM32MP15x enters LPLV-Stop. The STPMIC1B goes in Power-ON-alternate state:
   – Buck1 voltage decreases to 0.9 V
   – All buck regulators go into Low-power mode.
4. On a wakeup event, the STM32MP15x leaves LPLV- mode and asserts a PWR_ON signal:
   a. The STM32MP15x tSEL_VDDCORETEMPO is timed out to wait for VDDCORE to reach the Run mode operating supply voltage level.

Note: tSEL_VDDCORETEMPO = 380 µs typ (see [5] and [6]).

   b. The STPMIC1B goes in Power-ON-main state:
      - BUCK1 voltage rises from 0.9 V to 1.2 V (in 130 µs max)

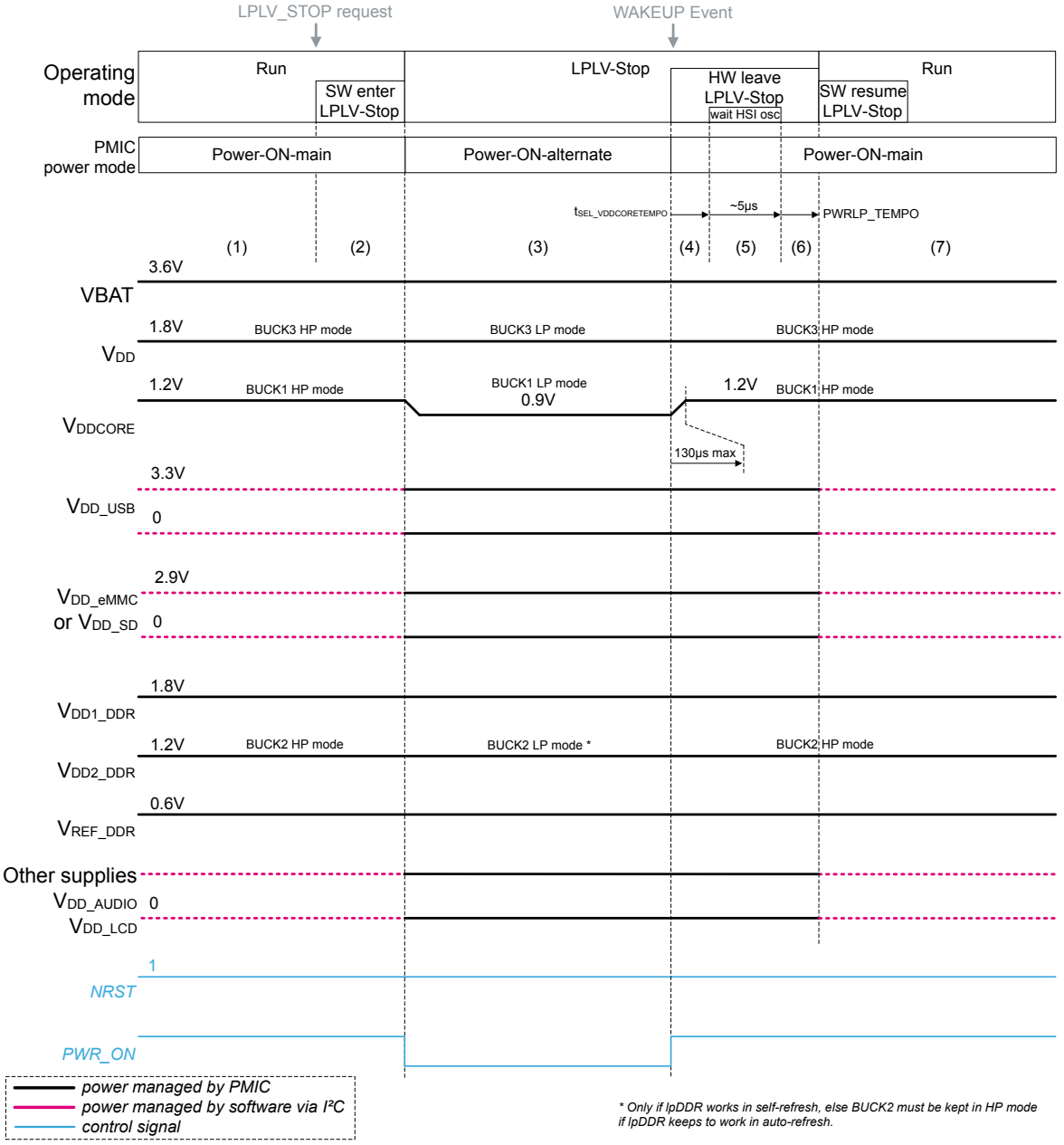
Note: STPMIC1B Buck1 has 2.3 mV/µs min slew rate.

      - In parallel, all buck converters go from Low power to High power mode (100 µs).
5. Once the tSEL_VDDCORETEMPO is elapsed, a clock restore process is performed in the STM32MP15x.
6. Once the STM32MP15x HSI clock oscillator is stable (~5 µs), the PWRLP_TEMPO timer is timed out to waiting for peripheral to be stable.

Note: PWRLP_TEMPO is an STM32MP15x dedicated timer designed to wait for the regulator recovery when the application goes from LPLV- mode to Run mode. When using the STPMIC1, PWRLP_TEMPO delay can be set to 0 or bypassed as tSEL_VDDCORETEMPO duration is higher than the STPMIC1B total recovery time. Nevertheless, if an application is set in LP_ (instead of LPLV_), the STM32MP15x does not time out the tSEL_VDDCORETEMPO delay. So, PWRLP_TEMPO must be set to 100µs to let the STPMIC1B regulators recover from LP to HP mode.

7. When PWRLP_TEMPO elapses, the application goes into Run mode. The software resume LPLV- mode (restores clocks, resumes lpDDR from self-refresh, …).

Figure 6. LPLV-Stop mode sequence



5.3.2 Standby mode

The application Standby mode sequence is shown in Figure 6 according to the implementation shown in Figure 1.

In this application, the eMMC flash memory voltage (VDD_eMMC) must be present when leaving Standby mode allowing the STM32MP15x to read the boot software (FSBL). In Standby mode, lpDDR memory works in self-refresh.

1. The application is powered up and is operating in Run mode; the STPMIC1B is in Power-ON-main state.
2. When Standby mode is requested, the software prepares to enter standby by changing:
   a. The STM32MP15x settings such as:
      - Stopping certain clocks
      - Sets DDR in Self-Refresh
      - Sets POPL and EADLY

Note: The STM32MP15x POPL timer allows the STM32MP15x to be kept in standby and to assert a PWR_ON signal low for a minimum duration to allow the peripheral regulators to stop before restarting them. This is to ensure the peripherals restart properly if a wakeup event occurs just after application goes into standby. The STPMIC1B has a discharge resistors on each regulator output that allows all of the regulator output voltages to discharge in less than 3 ms. So, POPL can be set to a minimum of 3 ms or can be kept with default value (10 ms) if the wakeup duration from standby is not critical.

The STM32MP15x EADLY timer is dedicated to preventing boot ROM performing any access to the boot peripheral before it is ready when recovering from Standby mode. Typically waiting for a stable voltage on the Flash memory (eMMC or SD-card) to ensure the boot software is reliably read by the boot ROM . In this application, VDD_eMMC rises in less than 500 µs (LDO3). So EADLY can be set to 500 µs minimum or can be kept with default value (10 ms) if wakeup duration from standby is not critical.

      - Timers and so on.

   b. The STPMIC1B settings:
      i. BUCK1 (VDDCORE), LDO2 (VDD_eMMC): ON HP in main mode, OFF in alternate mode
      ii. LDO4 (VDD_USB), LDO5 (VDD_SD), LDO1 (VDD_AUDIO), LDO6 (VDD_LCD): OFF in main and alternate modes.
      iii. BUCK2 (VDD2_DDR), BUCK3 (VDD) ON HP in main mode and ON LP in alternate mode
3. The STM32MP15x resets the LPDS and LVDS bits to wait while entering Standby mode: PWR_ON signal is de-asserted when the STM32MP15x enters standby:
   a. The POPL timer is started to prevent the STM32MP15x leaving standby before POPL elapses.
   b. The STPMIC1B goes in Power-ON-alternate state:
      i. BUCK1 (VDDCORE), LDO2 (VDD_eMMC): regulators are powered OFF
      ii. BUCK2 (VDD2_DDR), BUCK3 (VDD) regulators go in LP mode
4. On a wakeup event, the STM32MP15x leaves Standby mode and asserts a PWR_ON signal:
   a. The STPMIC1B goes in Power-ON-main state:
      i. BUCK1 (VDDCORE), LDO2 (VDD_eMMC): regulators are powered ON
      ii. BUCK2 (VDD2_DDR), BUCK3 (VDD) regulators go into HP mode
   b. When VDDCORE voltage is above the VPVDCORE_0 min threshold , a tVDDCORETEMPO is started. As long as the tVDDCORETEMPO timer is not elapsed, the STM32MP15x is kept in reset internally.

Note: STM32MP15x internal voltage threshold. VPVDCORE_0 rising edge is 0.95 V min.

STM32MP15x internal delay. tVDDCORETEMPO is 200 µs min.

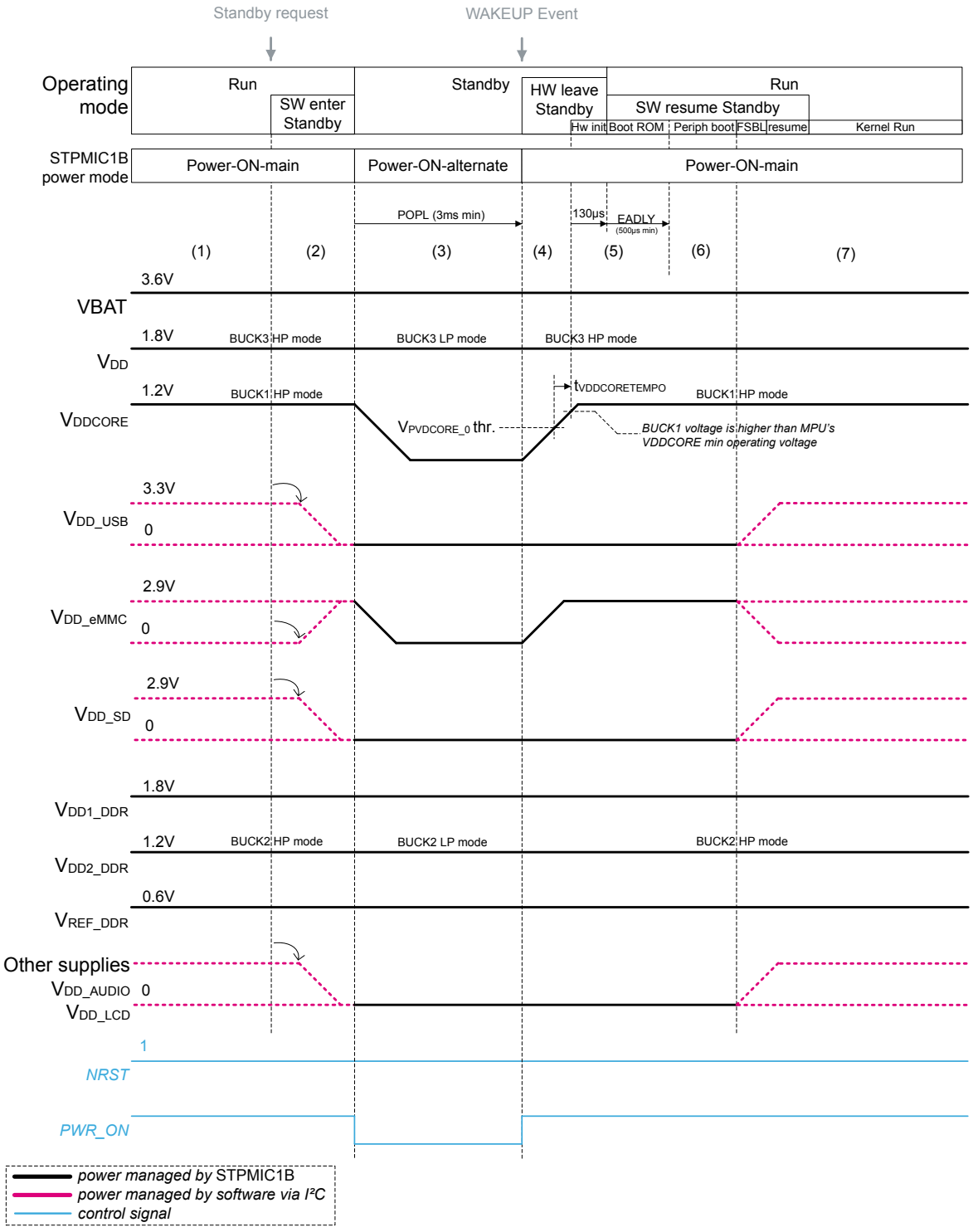
5. When tVDDCORETEMPO elapses, the STM32MP15x is taken out of internal reset (VDDCORE_OK):
   a. VDDCORE voltage is higher than STM32MP15x VDDCORE min operating voltage Section 5.3.2 .

Note: BUCK1 has a 2.3 mV/µs minimum slew rate guarantying a VDDCORE voltage is higher than the STM32MP15x VDDCORE min operating voltage when tVDDCORETEMPO elapses.

   b. The STM32MP15x performs an internal hardware initialization (enables the HSI and option bytes loading over a 130 µs duration) then enters in Run mode.
   c. EADLY delay timer is started.

6. When the EADLY delay timer is elapsed, the boot ROM starts accessing external peripherals (flash memory) to load and execute boot software. Implicitly, when EADLY has elapsed, VDD_eMMC voltage is stable:
   a. The boot ROM is read from the eMMC, verifies and executes the FSBL.
   b. From this step, the software can set the STPMIC1B via I²C interface to set any regulator.
7. The software detects an “exit from Standby mode” it then resumes and runs the Kernel software.

Figure 7. Standby mode sequence



5.4 User reset and crash recovery management

As introduced in Section 4.3 , the STM32MP15x and the STPMIC1B both have bidirectional active low reset pins interconnected (see Figure 1 signal NRST).

If an STM32MP15x crash occurs (iwdg1_out_rst or iwdg2_out_rst watchdog elapsing), a reset pulse is generated by the STM32MP15x on NRST signal. The reset pulse is caught by the STPMIC1B that triggers an immediate power cycle sequence: a STPMIC1B power-down sequence followed by a STPMIC1B power-up sequence.

A power cycle allows the peripherals to restart and reset properly after a crash occurs; especially for peripherals that do not have a reset input signal. Power cycling is mainly recommended for peripheral boot devices and Flash memory devices such as: eMMC, NAND, NOR, SD-Card. Power cycling is not performed on the STPMIC1B BUCK3 (VDD) that needs to be keep enabled during reset (see Section 5.1.2 for details of the STPMIC1B mask_reset option).

If the reset button is pressed by a user, the same power cycle sequence is performed by the STPMIC1B.

5.4.1 Crash recovery management or user reset sequence

The sequence in Figure 8 illustrates a crash recovery sequence according to the implementation shown in Figure 1.

1. The application is powered up and is in Run mode; the STPMIC1B is in Power-ON-main state. A crash occurs (iwdg1_out_rst or iwdg2_out_rst watchdog elapsing) or the reset button is pressed by the user generating a pulse on NRST signal.
2. The STPMIC1B detects the reset assertion (NRST pulse low) and starts a non-interruptible power cycle :
   a. The STPMIC1B asserts NRST low
   b. The STPMIC1B performs power-down sequence
   c. The STPMIC1B checks the conditions to restart (such as VBAT, temperature) and reloads the internal NVM
   d. The STPMIC1B performs power-up sequence
   e. The STPMIC1B releases NRST .

Note: If the reset signal (NRST) is still asserted at this step (such as the user is still pressing reset button), the STPMIC1B waits for the reset signal to be released before rearming the reset circuit. This is to avoid the STPMIC1B repeating a power cycle loop.

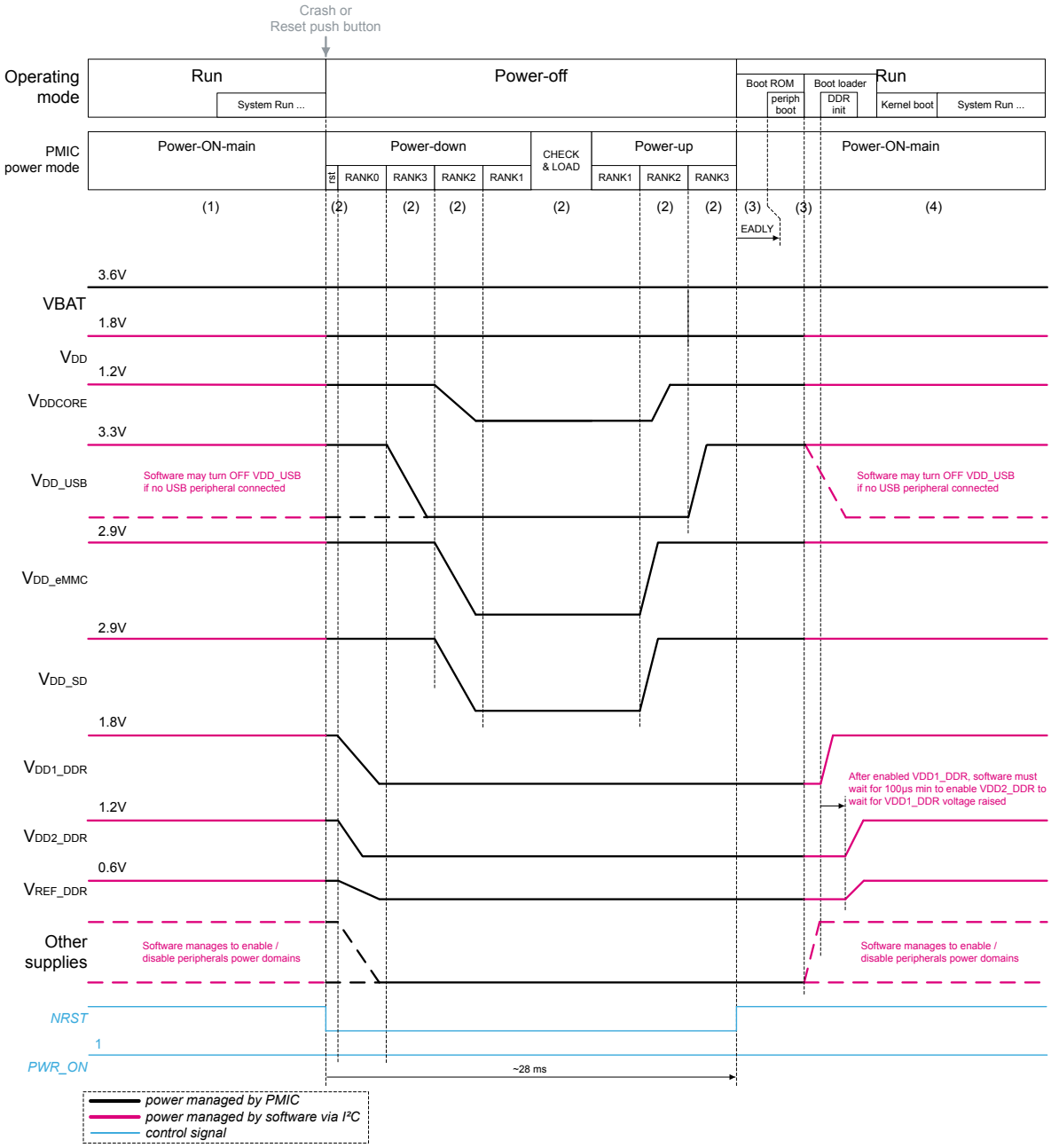
Note: STPMIC1B power cycle duration is ~28 ms.

3. The NRST signal rises as the STM32MP15x and the STPMIC1B release their respective reset pins (and reset button released):
   a. The STM32MP15x EADLY delay timer (10 ms) is started.

Note: The EADLY timer is dedicated to preventing boot ROM performing any access to the boot peripheral before it is ready. Typically waiting for a stable voltage of Flash memory (eMMC or SD-card) to ensure the boot software is reliably read by the Boot ROM. Default value after reset is 10ms delay. (see [5] for details).

   b. When EADLY is elapsed, the Boot ROM starts accessing external peripherals (either eMMC or SD-card depending on the STM32MP15x boot pins setting) to load and execute boot loader software.
   c. The Boot loader controls any STPMIC1B regulator (such as to initiate an LCD and plot splash screen).
4. The boot loader initializes the DDR then loads and executes the Kernel:
   a. LDO3 (VDD1_DDR) is enabled in bypass mode. The software waits at least 100 µs.
   b. DDR_VREF (VREF_DDR) is enabled.
   c. BUCK2 (VDD2_DDR) is enabled at 1.2 V. The software waits for at least 1.4 ms for BUCK2 to be ready.
   d. The software initializes the STM32MP15x DDR controller and lpDDR device.
   e. The Boot loader loads and executes the kernel and the kernel initializes.
   f. System is running.

Figure 8. Crash recovery sequence



5.5 Software management example

This section presents one possible software integration strategy of the STPMIC1B management by the STM32MP15x.

The OpenSTLinux software distribution integrates the way the STPMIC1B regulators are driven and configured by the STM32MP15x to match Operating mode presented in Section 5.1 .

To summarize the main OpenSTLinux integration point:

- Interface with the STPMIC1B is performed in the low level “secure monitor” part of the boot chain split between FSBL (for example : TF-A) and SSBL (for example: U-Boot).
- The STPMIC1B power management strategy presented in Table 4 below is configured in Secure Monitor dts file (typ TF-A) using Linux Regulator framework binding terminology.
- Each STPMIC1B power source is seen as a “regulator” on which software application and driver registers as a “consumer”. Typically, a regulator is enabled when it is requested by one consumer. The exception is made for the core supply which has to be kept alive whatever consumer registration state with the “Always-on” option.

To learn more about the Power management function in OpenSTLinux, refer to following online user guide articles:

https://wiki.st.com/stm32mpu/wiki/Power_overview

https://wiki.st.com/stm32mpu/wiki/Regulator_overview .

Also refer to the Linux® kernel binding documentation for regulator framework.

Table 4. STPMIC1B power management options

			Power State			
Application Power Domain	STPMIC1B supply source	Default Power On State (NVM)	STM32MP15x = Run / Stop PMIC= POWER_ON Main	STM32MP15x = LPLV-Stop PMIC= POWER_ON Alternate	STM32MP15x = STANDBY DDR OFF PMIC= POWER_ON Alternate	Options/ Comments
VDD	BUCK3	1.8V	1.8V / HP(1) / Always-on(2)	1.8V / LP	1.8V	Mask_reset(3), overcurrent protection (4)
VDDCORE	BUCK1	1.2V	1.2V/ HP / Always-on	0.9V / LP	Off	Overcurrent protection
VDD_USB	LDO4	3.3V	3.3V/ Consumer driven(5)	3.3V / Consumer driven	Off	-
VDD1_DDR	LDO3	OFF	1.8V/ Always-on	1.8V	Off	Overcurrent protection
VDD2_DDR	BUCK2	OFF	1.2V / HP / Always-on	1.2V / LP	Off	Overcurrent protection
VREF_DDR	DDR_REF	OFF	0.6V / Always-on	0.6V	Off	-
VDD_eMMC	LDO2	2.9V	2.9V/ Consumer driven	2.9V/ Consumer driven	Off / Boot on(6)	-
VDD_SD	LDO5	2.9V	2.9V/ Consumer Driven	2.9V/ Consumer driven	Off	-
VBUS_DR	PWR_USB_SW	OFF	ON	Off (suspend not supported on MP1 side)	Off	Pure software. No alternate registers
VDD_AUDIO	LDO1	OFF	1,8V / Consumer Driven	1,8V / Consumer driven	Off	-
VDD_LCD	LDO6	OFF	1,8V / Consumer Driven	1,8V / Consumer driven	Off	-

1. *HP/LP mode of the STPMIC1B regulator. Refer to [2]*
2. *Always-on: Keeps the core voltage on even if there is no software consumer.*
3. *Mask_reset: Specify the STPMIC1B mask_reset option to this regulator not to be impacted by a reset power cycle. (see Section 5.1.2)*
4. *Overcurrrent protection: Specify the STPMIC1B option OCPOFF on this regulator. Overcurrent detection leads to a STPMIC1B shutdown. Refer to [2]*
5. *Consumer driven: Linux driver turns the regulator on/off following consumer demand. When entering low power mode, the last Run status is applied (Main mode duplicate in Alternate mode). The user has to consider the required status before entering low power mode. For example when the powered peripheral is set as wake-up source.*
6. *Boot on: Software must set this regulator on in POWER_ON Main mode before entering into Low power mode (switch to STPMIC1B Alternate mode) in order to turn on immediately on wake-up (switch back to Main mode).*

6 USB port management

6.1 USB port using legacy micro-AB (OTG) or micro-B receptacle (DR)

Figure 1, Figure 2 and Section 4.2.7 illustrate a micro-AB receptacle for USB OTG extension support. Alternatively, a micro-B receptacle (having an ID pin) may be used only if USB DR mode is expected. Using a micro-B receptacle, the application natively supports the USB peripheral mode and it also supports the USB host mode if a “non USB compliant” adapter Section 6.1 pin ID is set to ground.

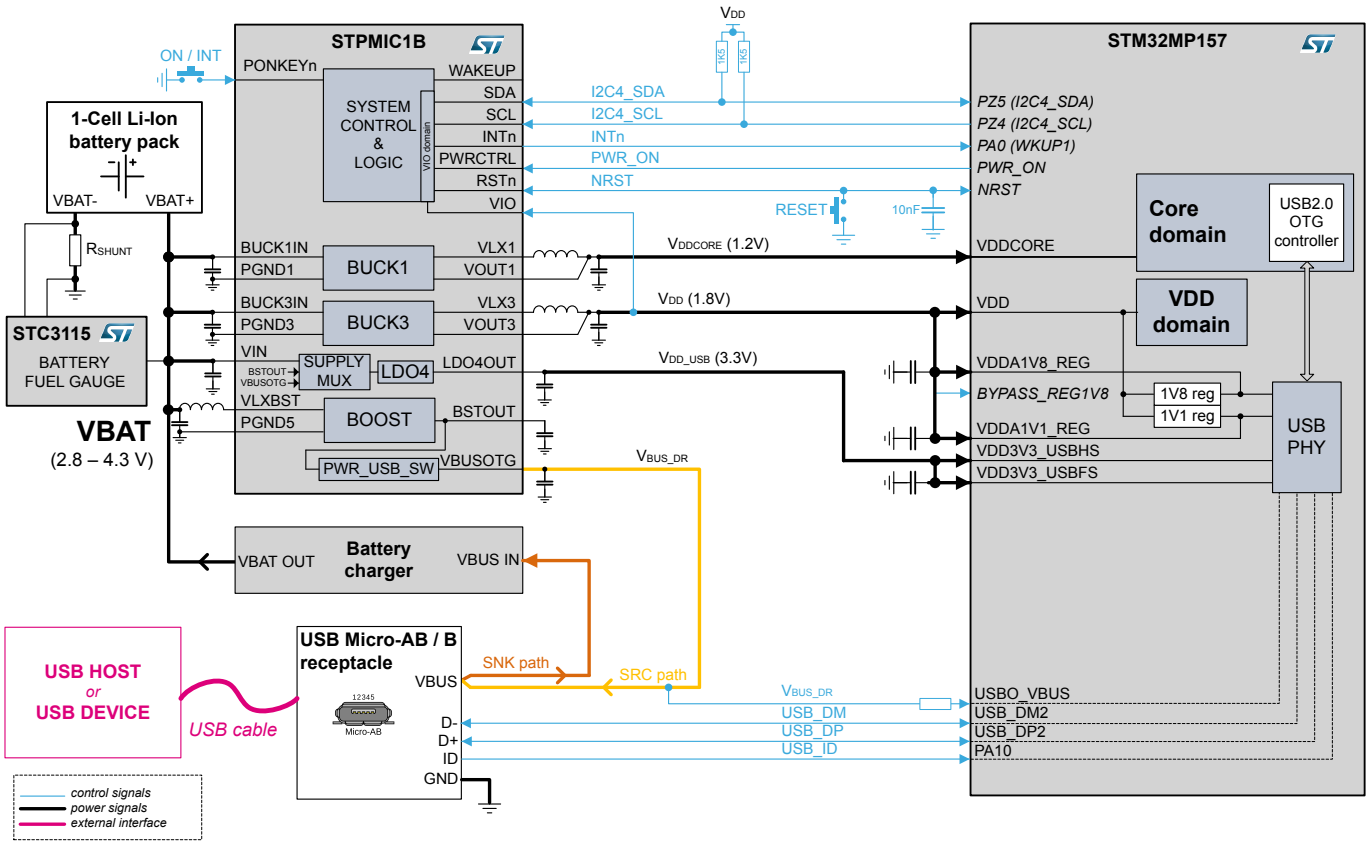
Note: Adaptor stands for a USB micro-B plug with ID pin grounded to USB type-A receptacle.

Figure 9 provides details on the USB port integration with a Type micro-B or a Type micro-AB receptacle.

The VBUS pin of a USB receptacle is connected to VBUS_DR power domain and it is interconnected to:

- The STPMIC1B VBUSOTG pin: PWR_USB_SW is a power switch which provides VBUS to the USB receptacle from Boost SMPS converter for a A-session or to detect a valid VBUS from USB receptacle for a B-session.
- The STM32MP15x USBO_VBUS pin to detect the presence of VBUS_DR voltage for a B-session
- The Battery charger input to charge the Li-Ion battery pack from the USB receptacle in B-session.

Figure 9. USB port management using a USB micro-AB or micro-B receptacle



Note:

1. *STM32MP15x and STPMIC1B are simplified figures focusing on USB feature integration.*
2. *STM32MP15x decoupling scheme not shown (see [1] related chapter).*
3. *STPMIC1B discrete component values not shown (see [2]).*
4. *Additional protection, such as ESD, EMI filtering, overvoltage, not shown.*

6.1.1 USB peripheral plug detection and peripheral removal detection management

Power-up by USB host peripheral plug:

This document assumes the application in Figure 9 is in Power-off mode with the STPMIC1B in Off mode (See Table 3 for details on the operating modes). The battery voltage VBAT is higher than the STPMIC1B VINOK_rise allowing STPMIC1B to Power-ON. A cable from USB host peripheral or USB charger peripheral is plugged into the USB receptacle causing the VBUS_DR voltage to rise. This triggers a STPMIC1B Turn-ON conditions (STPMIC1B VBUSOTG pin connected to VBUS_DR).

Note: A USB device peripheral cable plug is not a STPMIC1B Turn-ON condition as USB_ID signal is not connected to the STPMIC1B to detect a Turn-ON event.

Wake up by USB peripheral plug:

This document assumes the application in Figure 9 is in Stop, LPLV-Stop or Standby operating mode. Implicitly, the STPMIC1B is in Power-ON mode.

A cable from the USB host peripheral or USB charger peripheral is plugged into the USB receptacle causing the VBUS_DR voltage to rise. The rising voltage on the STPMIC1B VBUSOTG pin triggers an interrupt to the STM32MP15x via INTn signal that wakes up the STM32MP15x.

Note: STPMIC1B interrupts need to be enabled by software before being activated by setting the STPMIC1B INT_MASK_R1[VBUSOTG_RI] bit. See [2] for details.

Wake up by USB peripheral removal:

This document assumes the application in Figure 9 is in Stop or LPLV-Stop or Standby operating mode. Implicitly, the STPMIC1B is in Power-ON mode.

A cable from the USB host peripheral or USB charger peripheral is removed from the USB receptacle causing the VBUS_DR voltage to fall. The falling voltage on the STPMIC1B VBUSOTG pin triggers an interrupt on the STM32MP15x via INTn signal that wakes up the STM32MP15x.

Note: STPMIC1B interrupts need to be enabled by software before being activated by setting the STPMIC1B INT_MASK_R1[VBUSOTG_FA] bit. See [2] for details.

Run time detection by USB peripheral plug:

The application in Figure 9 is in Run operating mode. Implicitly, the STPMIC1B is in Power-ON mode.

- **USB host or charger peripheral case:**

  A cable from USB host peripheral or USB charger peripheral is plugged into USB receptacle causing the VBUS_DR voltage to rise. In Run operating mode, there are two detection sources:

  – Voltage rising on the STPMIC1B VBUSOTG pin which triggers an interrupt to STM32MP15x via INTn signal.

  – The rising voltage on the STM32MP15x USBO_VBUS pin can trigger an interrupt to the STM32MP15x USB OTG controller.

Note: STM32MP15x USB OTG controller interrupts need to be enabled. Status can be read in STM32MP15x GINTSTS[SRQINT]. See [5] for details.

- **USB device peripheral case:**

  A cable from a USB device peripheral is inserted into the USB receptacle grounding the USB_ID signal and pulling the STM32MP15x PA10 signal low. This triggers an interrupt in the STM32MP15x USB OTG controller.

Note: STM32MP15x USB OTG controller interrupts need to be enabled. Status can be read in STM32MP15x GOTGINT [IDCHNG]. See [5] for details.

Run time detection by USB peripheral removal:

The application in Figure 9 is in Run operating mode. Implicitly, the STPMIC1B is in Power-ON mode.

- **USB host or charger peripheral case:**

  A cable from USB host peripheral or USB charger peripheral is removed from USB receptacle causing the VBUS_DR voltage to drop. In Run operating mode, there are two detection sources:

  Voltage drop on the STPMIC1B VBUSOTG pin which triggers an interrupt to the STM32MP15x via INTn signal.

  Voltage drop on the STM32MP15x USBO_VBUS pin triggers an interrupt in the STM32MP15x USB OTG controller.

Note: STM32MP15x USB OTG controller interrupts need to be enabled. Status can be read in STM32MP15x GINTSTS[DISCINT]. See [5] for details.

- **USB device peripheral case:**

  A cable from a USB peripheral device is removed from USB receptacle causing the USB_ID signal to go high and so the STM32MP15x PA10 signal to go high also. This triggers an interrupt in STM32MP15x USB OTG controller.

Note: STM32MP15x USB OTG controller interrupts need to be enabled. Status can be read in STM32MP15x GOTGINT [IDCHNG]. See [5] for details.

6.1.2 Generate VBUS_DR voltage from battery voltage in USB host mode

When a USB device peripheral is plugged in and is detected by the STM32MP15x (see Section 6.1.1), the STM32MP15x initiates an A-session:

1. Software initializes USB OTG controller in A-device mode (host mode)
2. Software sets the STPMIC1B to generate VBUS_DR from the battery:
   a. Enable the STPMIC1B's Boost converter (to convert VBAT to 5.2 V)
   b. Wait for at least 1.5 ms for the BSTOUT voltage to stabilize
   c. Enable STPMIC1B PWR_USB_SW power switch: VBUS_DR rises to 5.2 V

Conversely, when the USB device peripheral is removed and a detached state is detected by the STM32MP15x, it ends the A-session and disable VBUS_DR:

1. Disable the STPMIC1B PWR_USB_SW power switch: VBUS_DR is tied low by discharge pull down (see Section 6.1.3)
2. Disable the STPMIC1B boost converter

6.1.3 VBUS_DR discharge pull-down management with STPMIC1B

The STPMIC1B PWR_USB_SW power switch has a programmable pull-down to discharge the decoupling capacitor on the VBUS_DR path.

After the STPMIC1B Power-ON, the pull-down needs to be enabled by software (disabled by default).

When PWR_USB_SW is enabled (A-session valid), the pull-down is automatically disabled by the STPMIC1B. When the USB device peripheral is removed and a detached signal has been detected by the STM32MP1 Series microprocessor, the STM32MP1 Series microprocessor disables the STPMIC1B PWR_USB_SW. The pull-down is automatically enabled by the STPMIC1B tying VBUS_DR low (discharge decoupling capacitor on VBUS_DR).

When a USB host peripheral is connected to a USB receptacle and a B-session valid voltage is detected, disable the STPMIC1B PWR_USB_SW pull-down by software to avoid continued current consumption. When the USB host peripheral is removed and a detached signal has been detected by the STM32MP1 Series microprocessor, the STPMIC1B PWR_USB_SW is set to pull-down by software to completely discharge VBUS_DR.

6.1.4 The STM32MP15x USB PHY supply with battery discharged below 3.3 V

The STM32MP15x USB PHY has three voltage domains 1.1 V, 1.8 V and 3.3 V (see Figure 9):

The 1.8 V (VDDA1V8_REG) is supplied from VDD power source from the STPMIC1B BUCK3.

The 1.1 V (VDDA1V1_REG) is supplied by the STM32MP15x 1V1_reg (internal STM32MP15x LDO) having VDD as power source.

The 3.3 V (VDD3V3_USBHS and VDD3V3_USBFS) are supplied from VDD_USB from STPMIC1B LDO4.

When the battery is discharged (VBAT is under than 3.3 V), the STPMIC1B LDO4 cannot generate VDD_USB voltage at 3.3 V from VBAT (PMIC's VIN pin). So the voltage is too low to supply 3.3 V the USB PHY power domain.

To continue supplying the STM32MP15x USB PHY with the right voltage (VDD_USB at 3.3 V) from a partially discharged battery, it uses one of the STPMIC1B LDO4 input: VIN, BSTOUT or VBUSOTG. The STPMIC1B automatically manages the LDO4 input from the highest voltage source.

USB host peripheral case:

When a USB host peripheral (like a Personal Computer) is plugged, it provides VBUS (~5 V) to the application: the VBUS_DR (and the STPMIC1B VBUSOTG) voltage rises. The LDO4 input is automatically switch to VBUSOTG power source as it is higher than battery voltage: VBUSOTG = VBUS ~5 V and VBAT <= 3.3 V.

With ~5 V as supply input source, the LDO4 is able to generate VDD_USB at 3.3 V.

USB device peripheral case:

When an USB device peripheral (like a USB memory stick) is plugged in and has been detected by the STM32MP15x (see Section 6.1.1), the STM32MP15x initiates a A-session and the STPMIC1B generates VBUS_DR voltage from the battery voltage (VBAT) via the boost converter (see section Section 6.1.2). The boost converter output (BSTOUT) generates a 5.2 V supply. The LDO4 input is automatically switch to BSTOUT power source as BSTOUT voltage is higher than battery voltage: BSTOUT = 5.2 V and VBAT <= 3.3 V.

With 5.2 V as supply input source, the LDO4 is able to generate VDD_USB at 3.3 V.

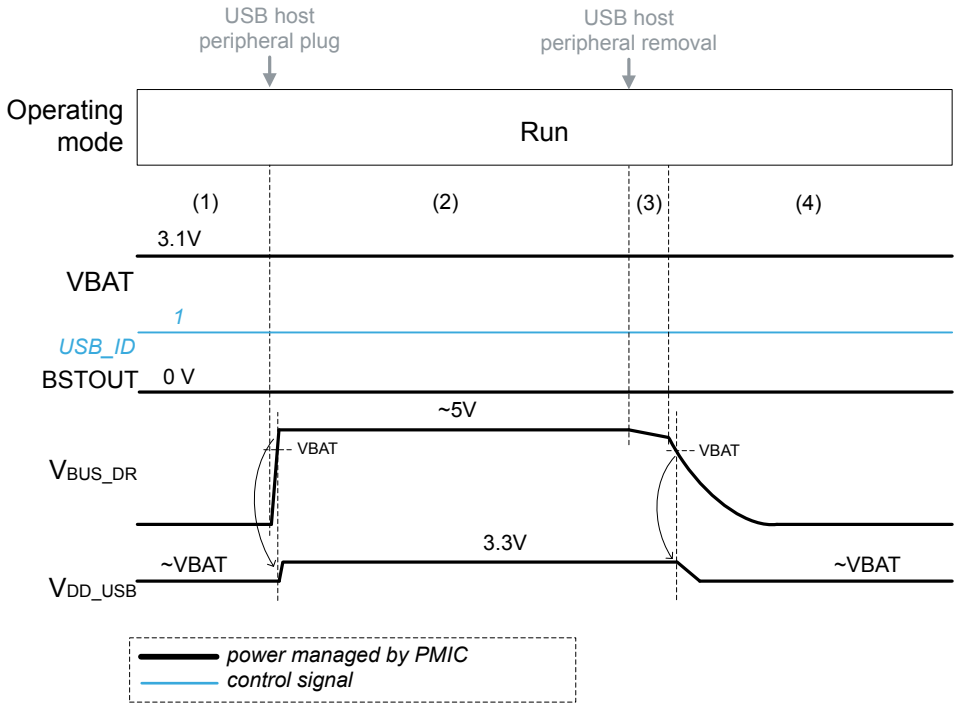
6.1.5 USB host peripheral supply management summary

The sequence in Figure 10 is dedicated to USB host peripheral management and provides a summary of Section 6.1.1 to Section 6.1.4 according to the implementation shown in Section 6.1 . For simplification, it is assumed that battery voltage is constant during the whole sequence (battery charging not shown).

Note: Example: a personal computer host.

1. The application is powered by a discharged battery (VBAT = 3.1 V) and is in Run operating mode. The STPMIC1B is in Power-ON-main state. The STPMIC1B PWR_USB_SW discharge pull down is initially enabled by software (see Section 6.1.3)
2. A USB host peripheral is plugged into the USB receptacle:
   a. VBUS_DR rises (as VBUS voltage is provided by host peripheral) pulls both the STPMIC1B VBUSOTG and STM32MP15x USBO_VBUS up. Both signals can trigger an interrupt (see Section 6.1.1).
   b. Once the interrupt is caught by the software, the software initializes the STM32MP15x USB OTG controller in B-device mode (device mode) and initiates USB communication.
   c. Once VBUS_DR is above VBAT, the STPMIC1B LDO4 is powered from VBUS_DR (see Section 6.1.4) and LDO4 output (VDD_USB) returns to 3.3 V.
   d. The software disables the STPMIC1B PWR_USB_SW discharge pull down (see Section 6.1.3)
3. The USB host peripheral is unplugged from the USB receptacle:
   a. VBUS_DR decreases slowly.
   b. Once VBUS_DR is below a defined voltage threshold, the STPMIC1B PWR_USB_SW (via the VBUSOTG pin) or the STM32MP15x USB OTG controller (via the USBO_VBUS pin) triggers an interrupt (see Section 6.1.1).
   c. The software ends the USB B-device session.
4. The software enables the STPMIC1B PWR_USB_SW discharge pull down:
   a. VBUS_DR is tied low .
   b. Once VBUS_DR is below VBAT, the STPMIC1B LDO4 is powered back from VBAT and LDO4 output (VDD_USB) returns to a value under 3.3 V. (see Section 6.1.4).

Figure 10. USB host peripheral supply management summary



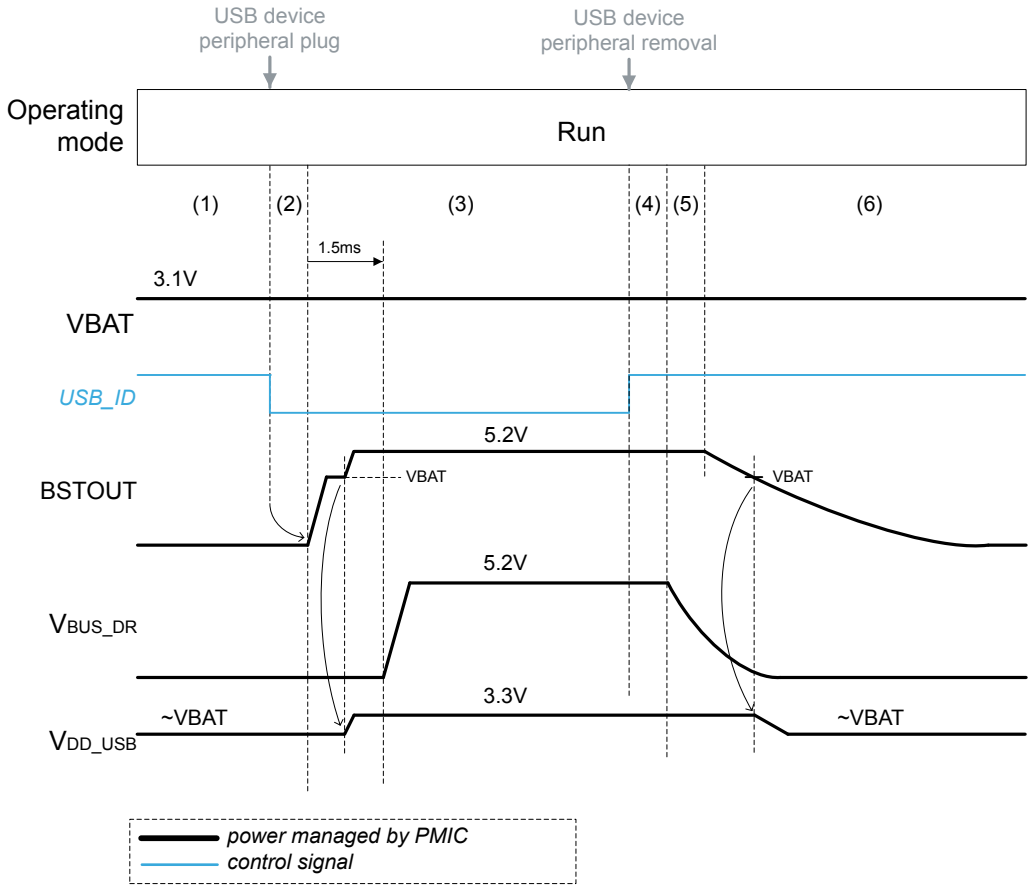
6.1.6 USB device peripheral supply management summary

The sequence in Figure 11 is dedicated to USB device peripheral management and provides a summary for Section 6.1.1 to Section 6.1.4 according to the implementation shown in Section 6.1 . For simplification, it is assumed that battery voltage is kept constant during the whole sequence (battery discharge not shown).

Note: Example: a USB mass storage dongle.

1. The application is powered by a discharged battery (VBAT = 3.1 V) and is in Run operating mode. The STPMIC1B is in Power-ON-main state.
2. A USB device peripheral is plugged into the USB receptacle:
   a. The USB_ID signal is grounded when the USB cable is inserted into USB receptacle. When the STM32MP1 Series microprocessor PA10 signal (USB_ID) goes low, an interrupt into STM32MP1 Series microprocessor's USB OTG controller is triggered (see Section 6.1.1).
   b. Software initializes the STM32MP1 Series microprocessor's USB OTG controller in A-device mode (host mode).
3. The software enables the STPMIC1B boost converter:
   a. Once the BSTOUT boost output voltage is above VBAT, the STPMIC1B LDO4 is powered from BSTOUT (see Section 6.1.4) and LDO4 output (VDD_USB) returns to 3.3 V.
   b. After a minimum delay of 1.5 ms (see Section 6.1.2), the software enables the STPMIC1B PWR_USB_SW power switch: VBUS_DR voltage rises. The USB device peripheral is powered. The USB OTG controller detects the USB device presence and initiates a USB A-session.
4. The USB device peripheral is unplugged from the USB receptacle:
   a. The USB_ID signal is released and it goes high. When STM32MP1 Series microprocessor PA10 signal (USB_ID) goes high, it triggers an interrupt to the STM32MP1 Series microprocessor's USB OTG controller (see Section 6.1.1).
   b. When the STM32MP1 Series microprocessor USB OTG controller detects a detach, it ends the USB A-device session.
5. The software disables the STPMIC1B PWR_USB_SW power switch. The PWR_USB_SW's internal pull-down is automatically enabled by the STPMIC1B, causing VBUS_DR to be tied low: VBUS_DR drops (see Section 6.1.3
6. While VBUS_DR is discharging, the software disables the STPMIC1B boost converter:
   a. BSTOUT voltage decreases
   b. Once BSTOUT boost output voltage is below VBAT, the STPMIC1B LDO4 is powered back from VBAT and LDO4 output (VDD_USB) returns to a value under 3.3 V. (see Section 6.1.4).

Figure 11. USB device peripheral supply management summary



6.1.7 Battery charging and battery monitoring with USB micro-B receptacle application

This sections is referring to Figure 9. It highlights the main interactions between the STM32MP15x, the STPMIC1B and the battery charger when a charger is connected to the USB Type micro-B receptacle.

Note: Battery charging algorithms and battery charging safety is out of this application note's scope

The application in Figure 9 natively supports single-cell or parallel-cell 3.6 V Li-Ion or Li-Polymer batteries. Also so called low cut-off batteries are supported thanks to STPMIC1B BOOST SMPS. Other battery types may also be supported but limited to the STPMIC1B minimum and maximum operating voltages.

In this application, the USB charger is a generic IC. It requires the support of both:

- Hardware charging: Autonomous dead battery pre-charge.
- Software charging: The charging process and the charge monitoring are managed by the STM32MP15x software algorithm

For the following functions refer to the appropriate sections:

- Battery interfaces and presents detection see Section 6.3.1 .
- Battery monitoring see Section 6.3.2 .
- Power-up / power-down from / to dead battery mode and battery charging see Section 6.3.3 .

6.2 USB port using USB Type-C™ receptacle

Figure 12 provides details about USB USB Type-C™ Port integration with a USB Type-C™ receptacle. This application example supports:

- USB 2.0 High Speed interface
- Source (SRC) power role: Supplies an attached sink peripheral with power up to 500 mA and operates in downstream-facing-port data role (USB host mode).
- Sink (SNK) power role: Get energy from an attached source peripheral (typically to charge the battery up to 5 V / 3 A depending of the source peripheral and battery charger IC capabilities) and operates in upstream-facing-port data role (USB device mode).
- USB Power-Delivery is not supported (including Dual-Role-Power)

The USB Type-C™ controller (STUSB1600) manages peripheral plug detection and attach recognition. It supports dead battery mode (for battery charging when battery is fully discharged) or to wake up the application when a peripheral attach. The USB Type-C™ controller is configured to manage either source power role or sink power role.

When a source peripheral attaches, the application operates as power sink port and takes upstream-facing-port data role (USB device mode).

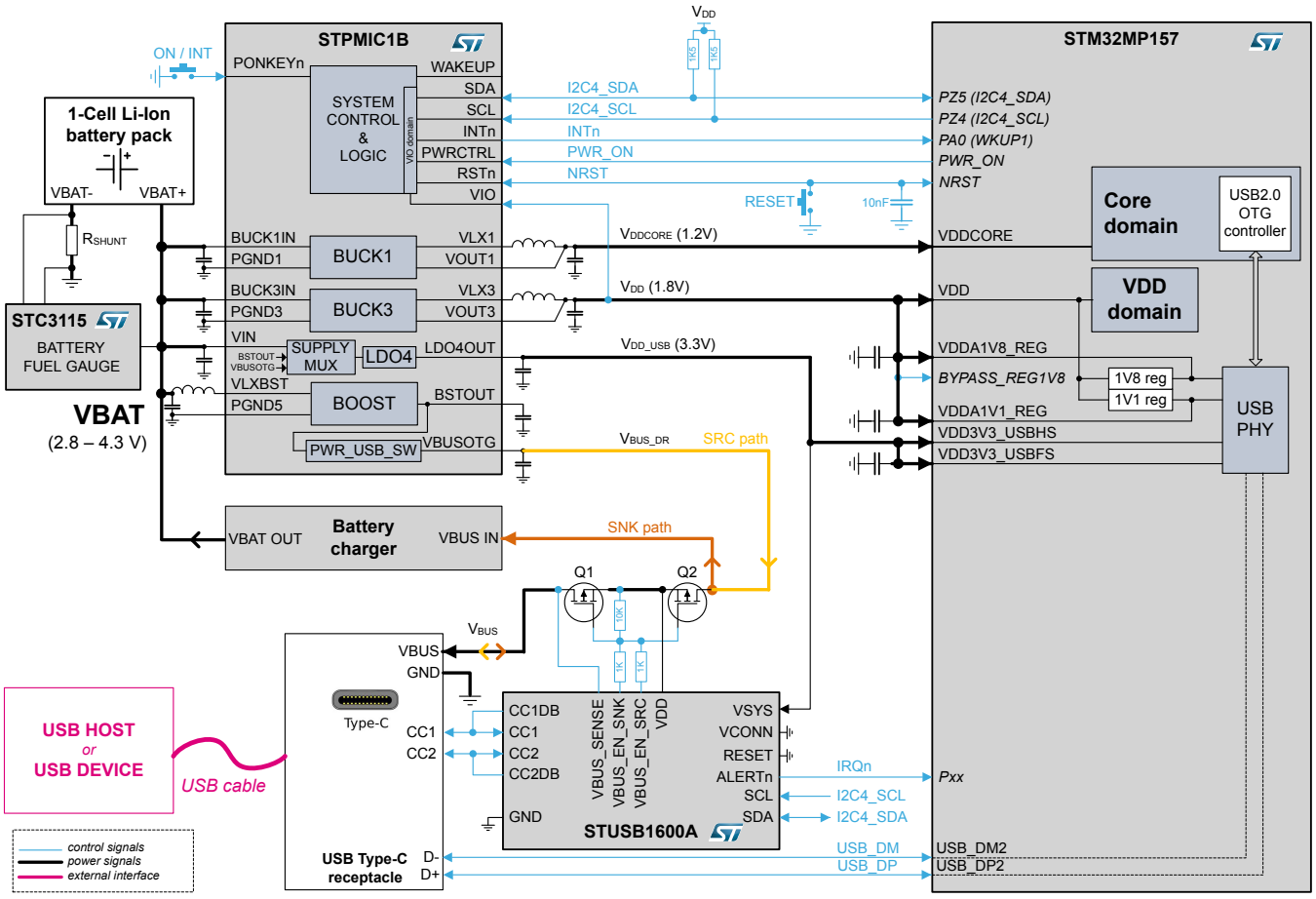
When a sink peripheral attaches, the application operates as power source port and takes downstream-facing-port data role (USB host mode).

The application is able to source up to 2.5 W (5 V / 500 mA) via the STPMIC1B boost converter and the PWR_USB_SW power switch. So, when sink peripheral attaches, the application advertises the sink peripheral with “Default USB Power”. At attach, the sink peripheral must follow USB 2.0 specification to allow sinking 500 mA budget from the application.

The VBUS pin of the USB Type-C™ receptacle is connected to a bidirectional power switch composed of two P-channel power mosfet in back-to-back. The bidirectional power switch is controlled by the USB Type-C™ controller (STUSB1600A) to manage either: Source or sink power path.

- Source power path: The battery voltage VBAT is converted to 5.2 V by the STPMIC1B boost converter (from VLXBST to BSTOUT). BSTOUT is internally connected to PWR_USB_SW power switch input. PWR_USB_SW output (VBUSOTG) is connected to VBUS_DR to provide voltage to the bidirectional power switch that bypass VBUS_DR to VBUS of USB Type-C™ receptacle.
- Sink power path: The USB Type-C™ provides VBUS voltage to the bidirectional power switch that bypass VBUS from the USB Type-C™ to VBUS_DR. The VBUS_DR is transferred to battery charger for battery charging and to STPMIC1B VBUSOTG pin (PWR_USB_SW) for VBUS detection to power-up the application.

Figure 12. **USB port management using a USB Type-C™ receptacle**



Note:

1. *STM32MP15x, STPMIC1B and USB USB Type-C™ controller (STUSB1600A) are simplified figures focusing on USB feature integration.*
2. *STM32MP15x decoupling scheme not shown (see [1] related chapter).*
3. *STPMIC1B discrete components value not shown (see [2][2]).*
4. *Additional protection, such as ESD, EMI filtering, overvoltage, not shown.*

6.2.1 USB peripheral plug detection and peripheral removal detection management

Power-up by USB source peripheral plug (source-to-sink connection in Dead-battery mode):

The application in Figure 12 is in Power-off mode with STPMIC1B in Off mode . The battery voltage VBAT is higher than STPMIC1B VINOK_rise allowing STPMIC1B to Power-on.

Note: See Table 3 for details about operating modes.

A source peripheral connects into USB Type-C™ receptacle. The USB Type-C™ controller (STUSB1600A) asserts a pull-down termination (Rd) on its CC pins and advertises as a sink (Dead Battery support). The source peripheral detects the pull-down termination, establish the source-to-sink connection, and provide the VBUS. Once the USB Type-C™ controller detects a valid voltage (VBUS_SENSE pin), it closes power switches (Q1/Q2) making VBUS_DR voltage rising. This trigs a STPMIC1B Turn-ON conditions (STPMIC1B VBUSOTG pin connected to VBUS_DR).

Note: A USB sink peripheral connection (sink-to-source connection) is not a Turn-ON condition of the application: When the application is in Power-off mode with STPMIC1B in Off mode, theUSB Type-C™ controller is not powered and cannot detect an attachment as it has no energy to asserts a pull-up termination (Rp) on its CC pins.

Wake up by USB source peripheral plug:

The application in Figure 12 is in Stop or LPLV-Stop or Standby operating mode. Implicitly, STPMIC1B is in Power-on mode. The USB Type-C™ controller initialized by the STM32MP15x in “power sink role” (see [7] for detail).

A source peripheral connects into USB Type-C™ receptacle. As the USB Type-C™ controller is advertising as a sink (the USB Type-C™ controller asserts a pull-down termination (Rd) on its CC pins), the source peripheral detects the pull-down termination, establish the source-to-sink connection, and provide the VBUS. Once the USB Type-C™ controller detects a valid voltage (from VBUS_SENSE pin), it closes power switches (Q1/Q2) making VBUS_DR voltage rising. There are two detection source possible to wake up the STM32MP15x:

- An attachment detected by the USB Type-C™ controller triggers an interrupt to STM32MP15x via IRQn signal.
- The voltage rising onto STPMIC1B VBUSOTG pin triggers an interrupt to STM32MP15x via INTn signal

Wake up by USB sink peripheral plug:

The application in Figure 12 is in Stop or LPLV-Stop or Standby operating mode. Implicitly, STPMIC1B is in Power-on mode. The USB Type-C™ controller initialized by the STM32MP15x in “power SRC role” (see [7] for detail).

A sink peripheral connects into USB Type-C™ receptacle. As the USB Type-C™ controller is advertising as a source (the USB Type-C™ controller asserts a pull-up termination (Rp) on its CC pins), it detects the attached sink then it trigs an interrupt to STM32MP15x via IRQn signal that wake up the STM32MP15x.

See Section 6.2.2 for details about VBUS voltage generation.

Wake up by USB source peripheral removal:

The application in Figure 12 is in Stop or LPLV-Stop or Standby operating mode. Implicitly, STPMIC1B is in Power-on mode. A USB source peripheral is connected onto the USB Type-C™ receptacle and the application works as “Attached.SNK” (see Section 2.1)

The source peripheral is removed from USB Type-C™ receptacle:

- The USB Type-C™ controller detects the removal. The USB Type-C™ controller open power switches (Q1/Q2) and triggers an interrupt to the STM32MP15x via IRQn that wakes up the STM32MP15x. Once the STM32MP15x awake, it reads the interrupt status into the USB Type-C™ controller and detects a USB source removal.
- The VBUS_DR voltage falls causing the voltage drop to the STPMIC1B VBUSOTG pin that triggers an interrupt to STM32MP15x via INTn signal that wakes up the STM32MP15x.

Wake up by USB sink peripheral removal:

The application in Figure 12 is in Stop or LPLV-Stop or Standby operating mode. Implicitly, the STPMIC1B is in Power-on mode. A USB sink peripheral is connected to the USB Type-C™ receptacle and the application works as “Attached.SRC” (see Section 2.1)

The sink peripheral is removed from USB Type-C™ receptacle. The USB Type-C™ controller detects the removal. The USB Type-C™ controller opens the power switches (Q1/Q2) and triggers an interrupt to the STM32MP15x via IRQn that wakes up the STM32MP15x. Once the STM32MP15x is woken-up, it reads interrupt status in the USB Type-C™ controller and detects an USB sink removal. The STM32MP15x disables the STPMIC1B PWR_USB_SW then the STPMIC1B Boost converter.

Note: To wake up the STM32MP15x from Standby state, the interrupt signal from the STPMIC1B or from USB Type-C™ controller must be connected to a wake-up source (WKUPx) input of the STM32MP15x.

Run time attached or un-attached detection of a USB sink peripheral or a USB source peripheral:

The application in Figure 12 is in Run operating mode. Implicitly, the STPMIC1B is in Power-on mode.

At run time, a sink or a source USB peripheral plug or removal uses the same detection mechanism as the wake-up described previously.

6.2.2 Generate VBUS voltage from battery voltage in source power mode

When a USB sink peripheral is plugged and the connection has been detected by the USB Type-C controller (see Section 6.2.1), it triggers an interrupt to STM32MP15x via IRQn:

1. The STM32MP15x software reads interrupt status register and detects a USB sink peripheral connection.
2. The STM32MP15x software sets the STPMIC1B to generate VBUS_DR from battery:
   a. Enable STPMIC1B Boost converter (to convert VBAT to 5.2 V).
   b. Wait for a 1.5ms minimum delay to stabilize the BSTOUT voltage (recommended 5ms or higher).
   c. Enable the STPMIC1B PWR_USB_SW power switch: VBUS_DR rises to 5.2 V.
3. The USB Type-C controller closes the power switches (Q1/Q2) which causes VBUS to rise. The application is in “Attached.SRC” state (see Section 2.1).
4. The STM32MP15x software can initiate a downstream facing port session (USB host mode).

6.2.3 STM32MP15x USB PHY supply with battery discharged below 3.3 V

See Section 6.2.4 for an overview about the STM32MP15x USB PHY supply constraints. This section refers to the application in Figure 12.

USB Source peripheral case:

When a USB source peripheral is connected to the application, the USB Type-C™ controller, initially set as “power SNK role”, advertises as a sink peripheral making the source peripheral provide VBUS (~5 V). If the VBUS voltage is in the valid range (see [7]) the USB Type-C™ controller closes power switches (Q1/Q2): the VBUS_DR (and the STPMIC1B VBUSOTG) voltage rises. The LDO4 input automatically switchs to VBUSOTG power source as VBUSOTG voltage is higher than battery voltage (see [2]):

- VBUSOTG = VBUS ~5 V
- VBAT <= 3.3 V.

With ~5 V as supply input source, the LDO4 is able to generate VDD_USB at 3.3 V.

USB sink peripheral case:

When a USB sink peripheral is attached to the application, the application generates the VBUS voltage to the sink peripheral thanks to boost converter that converts battery voltage to VBUS voltage (see Section 6.2.2 for details): The boost converter output (BSTOUT) generates a 5.2 V. The LDO4 input automatically switchs to BSTOUT power source as BSTOUT voltage is higher than battery voltage:

- BSTOUT = 5.2 V
- VBAT <= 3.3 V.

With 5.2 V as supply input source, the LDO4 is able to generate VDD_USB at 3.3 V.

6.2.4 USB source peripheral supply management summary

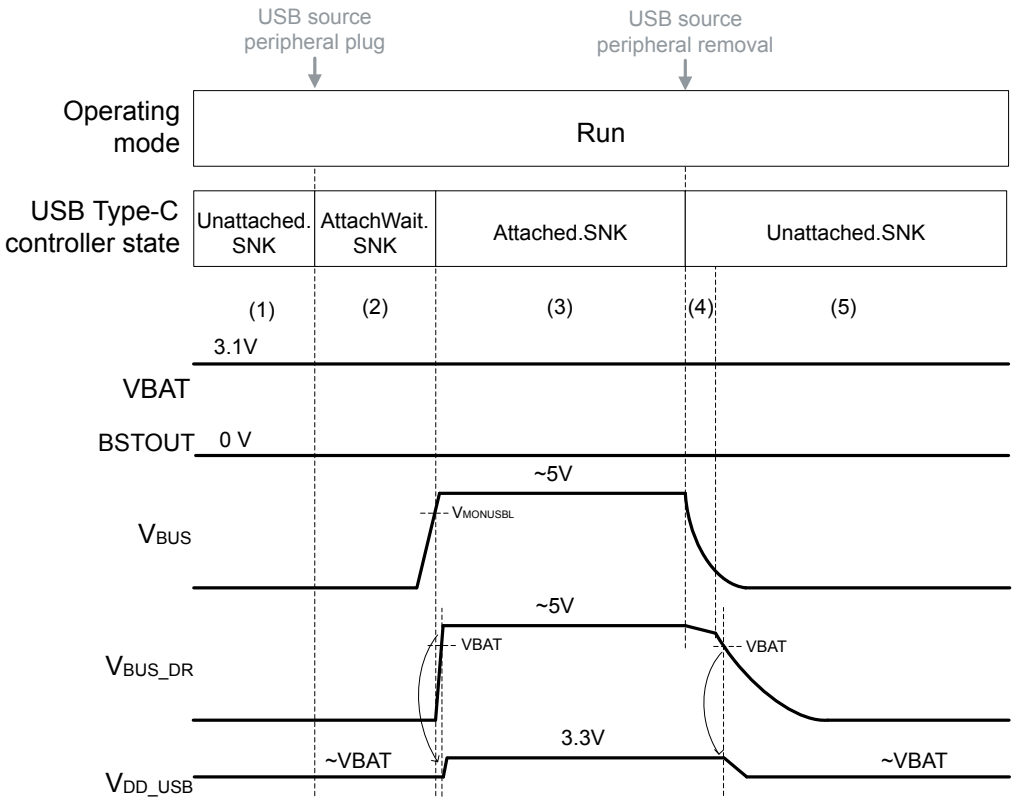
The sequence in Figure 13 is dedicated to USB source peripheral management (example: a Personal Computer host) and provides a summary from Section 6.2.1 to Section 6.2.3 according to the implementation shown in Figure 12. In this case, it is assumed that battery voltage is constant during the whole sequence (battery charging not shown).

Note:

1. Application is powered by a discharged battery (VBAT = 3.1V) and works in Run operating mode. The STPMIC1B is in Power-ON-main state. The USB Type-C™ controller is initially set by the STM32MP15x in “power SNK role” and it is performing in an unattached.SNK state (see [7] for detail).
2. A USB source peripheral is plugged into the USB receptacle:
   a. As the USB Type-C™ controller advertises as Unattached.SNK, it detects the source peripheral connection and it goes to AttachWait.SNK state.
   b. Once the USB source peripheral has detected the connection to the application, it supplies VBUS.

3. Once VBUS is above the VMONUSBL threshold (see [7] for detail):
   a. The USB Type-C™ controller goes in Attached.SNK state, it closes power switches (Q1/Q2) and it trigs an interrupt to the STM32MP15x (see Section 6.2.1).
   b. VBUS_DR rises causing the STPMIC1B VBUSOTG to rise and triggers an interrupt to the STM32MP15x (see Section 6.2.1).
   c. Once the interrupt is caught by the software, the software initializes the STM32MP15x USB OTG controller in B-device mode (device mode) and initiates USB communication.
   d. Once VBUS_DR is above VBAT, the STPMIC1B LDO4 is powered from VBUS_DR (see Section 6.2.2) and LDO4 output (VDD_USB) recovers to 3.3 V.
   e. The software disables STPMIC1B PWR_USB_SW discharge pull down (see Section 6.1.3).
4. The USB source peripheral is unplugged from the USB receptacle. Once the USB Type-C™ controller detects peripheral detachment (see [7] for detail):
   a. USB Type-C™ controller opens power switches (Q1/Q2), it triggers an interrupt to the STM32MP15x, it activates VBUS discharge path and it goes in Unattached.SNK state.
   b. Once VBUS discharge path is activated, VBUS voltage is tied low.
   c. Once power switches (Q1/Q2) are open, VBUS_DR starts to discharge slowly.
5. Once the interrupt from the USB Type-C™ controller is caught by the software:
   a. The software ends the USB B-device session.
   b. The software enables the STPMIC1B PWR_USB_SW discharge pull down pulling VBUS_DR low.
   c. Once VBUS_DR is below VBAT, the STPMIC1B LDO4 is powered back from VBAT and LDO4 output (VDD_USB) goes back below 3.3 V. (see Section 6.2.3).

Figure 13. USB source peripheral supply management summary



6.2.5 USB sink peripheral supply management summary

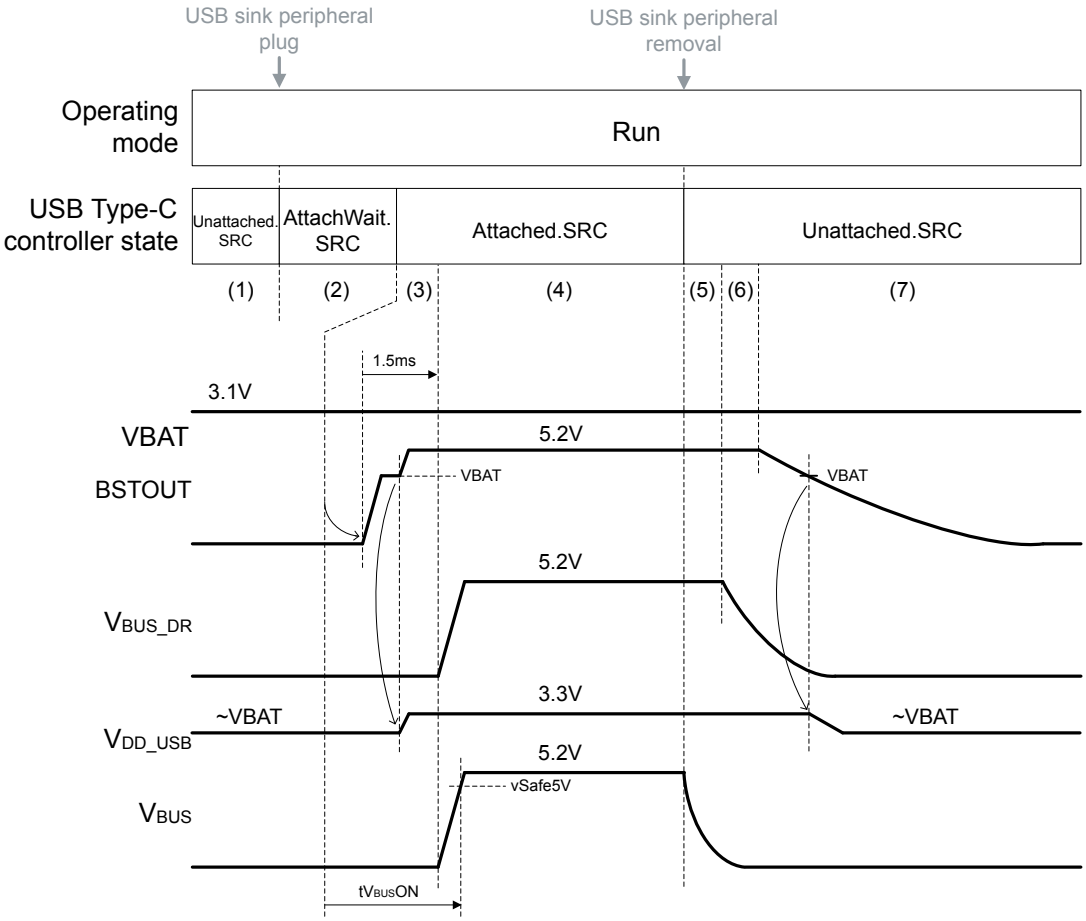
The sequence in Figure 14 is dedicated to USB sink peripheral management (Example: an USB mass storage dongle) and provides a summary for Section 6.2.1 to Section 6.2.3 according to the implementation shown in Figure 12. In the case of this procedure, it is assumed that battery voltage is constant during the whole sequence (battery discharge not shown).

1. Application is powered by a discharged battery (VBAT = 3.1 V) and works in Run operating mode. The STPMIC1B is in Power-ON-main state. The USB Type-C™ controller is initially set by the STM32MP15x in “power SNK role” and it is performing in Unattached.SRC state (see [7] for detail).
2. A USB sink peripheral is plugged into the USB receptacle. As the USB Type-C™ controller advertises as Unattached.SRC:
   a. The USB Type-C™ controller detects the sink peripheral connection.
   b. The USB Type-C™ controller goes to AttachWait.SNK state.
   c. After a debounce duration to guaranty sink peripheral connection (see tCCDebounce in Section 2.1), the USB Type-C™ controller goes in Attached.SRC state Section 6.2.5 , it closes power switches (Q1/Q2) and it triggers an interrupt to the STM32MP15x.

Note: When the USB Type-C™ controller enters Attached.SRC state, the VBUS voltage must be above vSafe5V voltage no longer than tVBUSON delay (275ms maximum). See Section 2.1 for details.

3. Once the interrupt is caught by the STM32MP15x software:
   a. The software initializes the STM32MP15x USB OTG controller in A-device mode (host mode).
   b. The software enables the STPMIC1B boost converter and sets the timer to a 1.5 ms minimum duration (see Section 6.2.3)
   c. Once BSTOUT boost output voltage is above VBAT, the STPMIC1B LDO4 is powered from BSTOUT and LDO4 output (VDD_USB) recovers to 3.3 V.
4. Once the 1.5ms delay timer is elapsed, the software enable the STPMIC1B PWR_USB_SW power switch:
   a. VBUS_DR voltage rise causes VBUS to rise.
   b. The USB device peripheral is powered. The USB OTG controller detects the presence of the USB sink peripheral and initiates an USB A-device session.
5. The USB sink peripheral is unplugged from the USB receptacle. Once the USB Type-C™ controller detects a peripheral detachment (see [7] for detail):
   a. USB Type-C™ controller opens the power switches (Q1/Q2), it triggers an interrupt to STM32MP15x, it activates VBUS discharge path and it goes in Unattached.SRC state.
   b. Once VBUS discharge path is activated, VBUS voltage is pulled low.
6. Once the interrupt from the USB Type-C™ controller is caught by the software:
   a. The software ends the USB A-device session.
   b. The software disables the STPMIC1B PWR_USB_SW power switch. The PWR_USB_SW’s internal pull-down is automatically enabled by the STPMIC1B causing VBUS_DR to be pulled low: VBUS_DR drops (see Section 6.1.3)
7. While VBUS_DR is discharging, the software disables the STPMIC1B boost converter:
   a. BSTOUT voltage decreases.
   b. Once BSTOUT boost output voltage is below VBAT, the STPMIC1B LDO4 is powered back from VBAT and LDO4 output (VDD_USB) goes back below 3.3 V. (see Section 6.2.3).

Figure 14. USB sink peripheral supply management summary



6.2.6 Battery charging and battery monitoring with USB Type-C™ receptacle application

This sections refers to Figure 12. This sections aims to highlight main interactions between the STM32MP15x, the STPMIC1B, the USB Type-C™ controller and the battery charger when a charger is connected to the USB Type-C™ receptacle.

Note: Battery charging algorithms and battery charging safety is out of scope of this application note.

Application in Figure 12 natively supports single-cell or parallel-cell 3.6 V Li-Ion or Li-Polymer batteries. Also so called low cut-off batteries are supported thanks to STPMIC1B BOOST SMPS. Other battery chemistries may also be supported but are limited to the STPMIC1B minimum and maximum operating voltages.

In this application, the USB charger is a generic IC. It requires to support both:

1. Hardware charging: Autonomous dead battery pre-charge.
2. Software charging: The charge and the charge monitoring are managed by STM32MP15x software algorithm.

The USB Type-C™ controller CC1DB pin and CC2DB pin are connected respectively to CC1 pin and CC2 pin to enable the Dead-Battery feature.

For battery interfaces and presents detection see Section 6.3.1

For battery monitoring see Section 6.3.2

For power-up / power down from / to dead battery mode and battery charging see Power-up / power-down from / to dead battery mode and battery charging.

Battery interfaces and presents detection

See Section 6.3.1

Battery monitoring

See Section 6.3.2

Power-up / power-down from / to dead battery mode and battery charging for USB Type-C™ receptacle application

See Section 6.3.4

6.3 Battery management

This section details the methods to manage the battery. It includes;

- Battery presence detection Section 6.3.1 .
- Battery monitoring (see Section 6.3.2 .
- Powering up and powering down from a dead battery and battery charging for:
  - Legacy micro-AB (OTG) or micro B receptacle (DR) (see Section 6.3.3 .
  - Type C receptacle (see Section 6.3.4).

6.3.1 Battery interfaces and presence detection

This application does not natively support a battery interface such as the Dallas Semiconductor's 1-wire™ or the Texas Instrument's Single-Wire SDQ™.

The battery presence detection and temperature monitoring must managed by the battery charger IC and / or a battery monitoring IC such as the STC3115.

Typically, battery pre-charging must not be allowed if a battery is not present. If this is the case, the charger is set to pre-charge (as no battery is present and so VBAT = 0, equivalent to a dead battery) and the VBAT voltage rises above STPMIC1B VINOK_rise threshold. The STPMIC1B therefore powers-up as describes in Section 5.2.1 . As the pre-charge current is usually very low, the VBAT voltage drops as soon as the STM32MP15x boots-up and starts consuming power causing the instability system and possibly crash.

6.3.2 Battery monitoring

This application embeds a STC3115 gas gauge IC to accurately monitor the battery state-of-charge (SOC).

To monitor the temperature during battery charge, the STC3115 embeds a temperature sensor. Alternatively, if battery pack embeds à NTC resistor (negative temperature coefficient), an STM32MP15x ADC input can be used to convert the NTC resistance to temperature.

6.3.3 Legacy micro-AB (OTG) or micro B receptacle (DR) power-up / power-down to or from dead battery mode and battery charging

When the battery pack is fully discharged with VBAT = 0 (battery pack embedded protection circuit is opened putting VBAT in high impedance) or with a discharged battery with VBAT < STPMIC1B VINOK_rise threshold, the application cannot Turn-ON (see Section 5.2.1). A pre-charge cycle with low charging current is required to preserve the battery chemistry. This cycle is fully managed by the battery charger IC until the application is able to Turn-ON (VBAT > STPMIC1B VINOK_rise). When the application is running, the STM32MP15x software manages and monitors battery charging. The process is illustrated in Figure 15

When the system is running with a discharged battery and without charger, the STPMIC1B VINLOW_rise threshold can be programmed to send an interrupt to the STM32MP15x when VBAT crosses STPMIC1B VINLOW_rise threshold. The STM32MP15x software must manage this interrupt before battery voltage gets too low. To protect battery chemistry against excessively low voltage, the STPMIC1B VINOK_Fall threshold can be programmed to automatically power down the application (see Section 5.2.1).

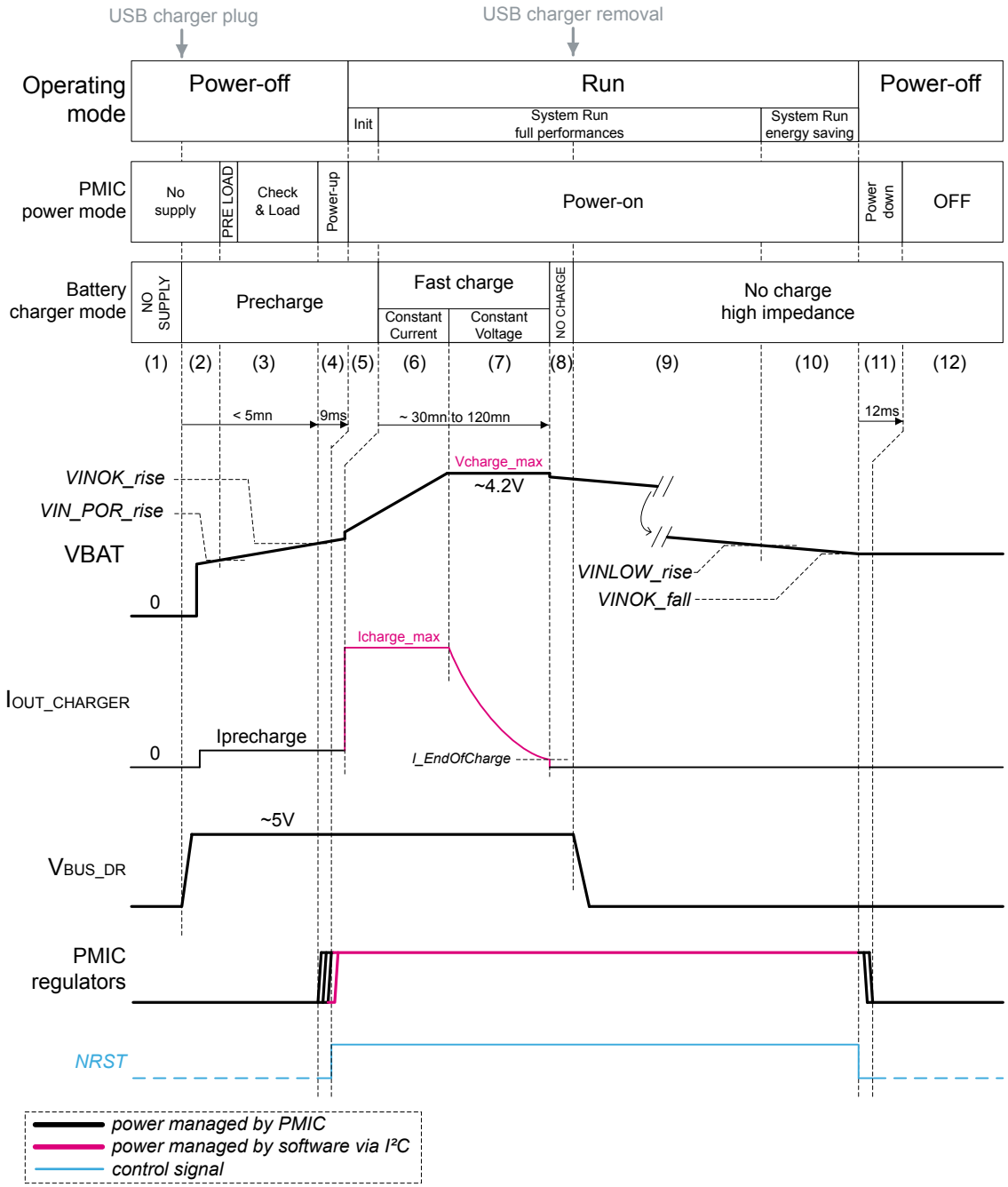
1. Application has no power: battery pack is present but it is fully discharged: VBAT = 0 (dead battery: battery pack embedded protection circuit is opened).

2. A cable from USB host peripheral (for example from a USB standard downstream port of a personal computer (PC)) or USB charger peripheral is plugged into the USB receptacle:
   a. VBUS_DR voltage rises to ~5V
   b. Once VBUS_DR voltage is stable, the battery charger starts a pre-charge cycle with a very low charging current. Battery voltage (VBAT) increases
   c. Once the battery pack embedded protection circuit detects a VBAT voltage higher than the battery cell voltage, it disables protection making battery cell connection to VBAT. The battery pack starts charging making VBAT voltage rising slowly.
3. Once VBAT voltage is above VIN_POR_rise:
   a. The STPMIC1B initializes, asserts NRST and pre-load its NVM contents
   b. As the STPMIC1B has “AUTO_TURN_ON” bit set by default in NVM, the STPMIC1B goes directly into “CHECK & LOAD” state as long as VBAT < VINOK_rise (see [2] for details).
4. Once VBAT voltage is above VINOK_rise, the STPMIC1B performs a power-up sequence then it releases the NRST signal (see Figure 6 for details about power-up sequence)
5. Once STM32MP15x NRST is rise, the STM32MP15x software starts execution and it initializes the application.
6. The STM32MP15x software initializes the USB charger IC:
   a. Stop the precharge
   b. Set the fast charge like max charging current, max charging voltage and so on. Those settings depend on the USB source current capability, the battery capacity and so on.
   c. As the battery is discharged (VBAT < max charging voltage setting) the USB charger IC charges the battery in constant current mode. The battery voltage increases slowly.

Note: Battery monitoring during charge is out of this document scope.

7. Once the VBAT voltage is above the max charging voltage setting:
   a. The USB charger IC switches to constant voltage charging mode.
   b. The battery voltage keeps stable and the charging current decrease during charge until the end of charge detection.
8. Once the charging current is below I_EndOfCharge threshold, the USB charger IC stop the charge and it goes in NO CHARGE state.
9. The cable from USB host peripheral or USB charger peripheral is removed from the USB receptacle:
   a. The VBUS_DR voltage drops and the USB charger IC goes in high impedance state.
   b. The application runs and battery is discharging
10. Once the VBAT voltage is below the STPMIC1B VINLOW_rise thresholds:
    a. The STPMIC1B trigs and interrupt to STM32MP15x to alert the STM32MP15x that battery has low voltage.
    b. The STM32MP15x software may enter in energy saving mode
11. Once the VBAT voltage is below the STPMIC1B VINOK_fall threshold:
    a. A turn-off condition occurs in STPMIC1B
    b. The STPMIC1B asserts NRST and performs a power-down sequence (see Figure 6 for details about power-down sequence)
12. STPMIC1B is in OFF mode:
    a. Application is in Power-off mode.
    b. As long as VBAT voltage is below VINOK_rise, the STPMIC1B cannot power-up

Figure 15. Power-up / power-down from/to dead battery mode



6.3.4 Type C receptacle application power-up / power-down to or from dead battery mode and battery charging

When the battery pack is fully discharged with VBAT = 0 (the battery pack embedded protection circuit is opened putting VBAT in high impedance) or with very discharged battery having VBAT < STPMIC1B VINOK_rise threshold, the application cannot Turn-ON (see Section 5.1.1). A pre-charge cycle with low charging current is required to preserve the battery chemistry. This cycle is completely managed by the battery charger IC until the application is able to Turn-ON (VBAT > STPMIC1B VINOK_rise). When the application is running, the STM32MP15x software manages and monitors battery charging.

When the system is running with a discharged battery and without charger, the STPMIC1B VINLOW_rise threshold can be programmed to send interrupt to the STM32MP15x when VBAT crosses the STPMIC1B VINLOW_rise threshold. The STM32MP15x software must manage this interrupt before the battery voltage gets too low. To protect battery chemistry against excessively low voltage, the STPMIC1B VINOK_Fall threshold can be programmed to automatically power down the application (see Section 5.1.1).

1. Application has no power: battery pack is present but it is fully discharged: VBAT = 0 (dead battery: battery pack embedded protection circuit is opened).
2. A cable from a USB Source peripheral is plugged into the USB receptacle:
   a. The USB Type-C™ controller enters in dead battery mode and it advertises as Unattached.SNK.
   b. Once the USB source peripheral has detected the connection to the application, it supplies VBUS.

Note: Example of a USB Source peripheral could be a USB Type-C™ port of a personal computer (PC) or a USB Type-C™ charger.

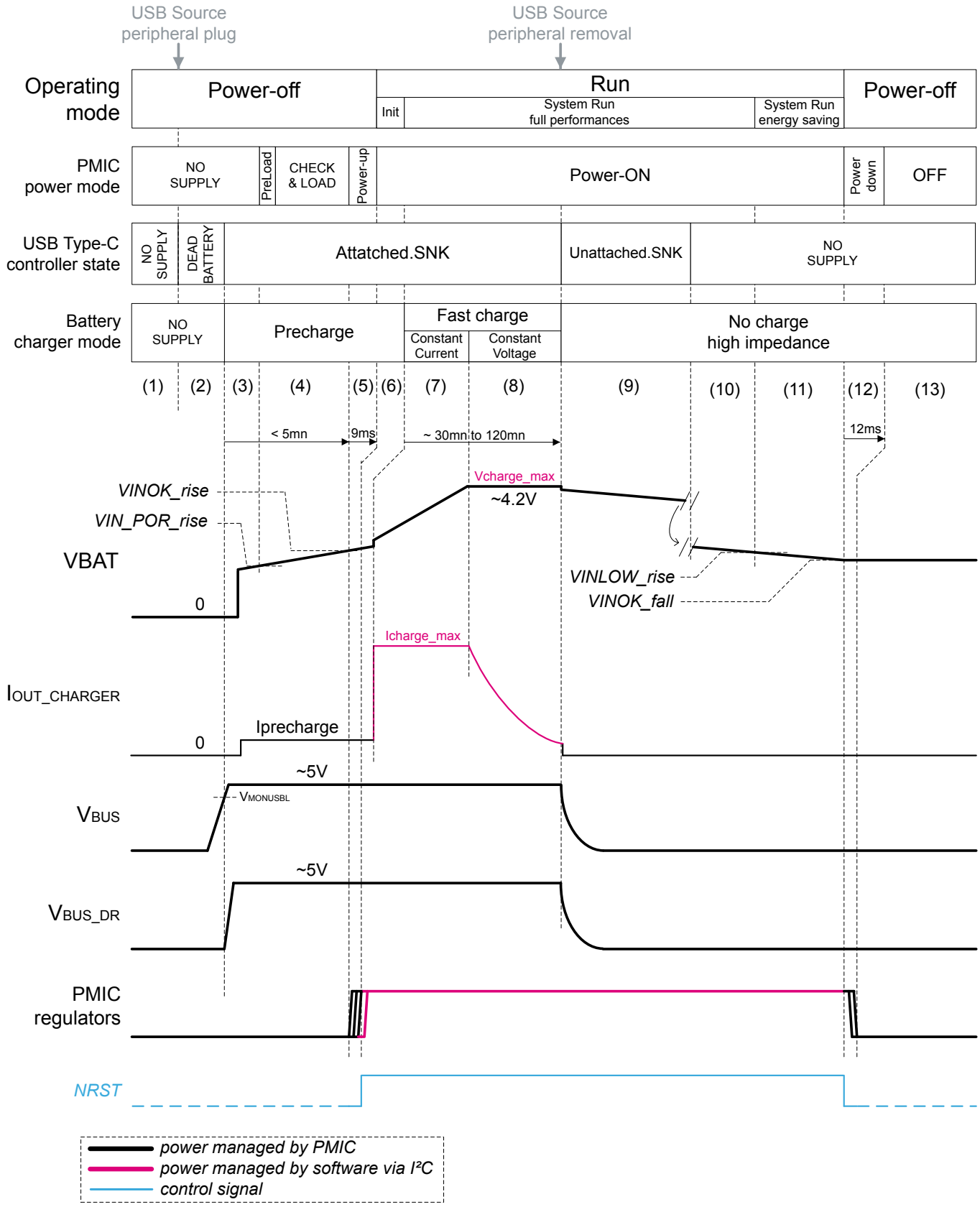
3. Once VBUS is above the VMONUSBL threshold (see [7] for detail):
   a. The USB Type-C™ controller goes in Attached.SNK state, it closes power switches (Q1/Q2) and it triggers an interrupt to the STM32MP15x (see Section 6.2.1).
   b. VBUS_DR voltage rises to ~5V.
   c. Once VBUS_DR voltage is stable, the battery charger starts a pre-charge cycle with a very low charging current. Battery voltage (VBAT) increases.
   d. Once the battery pack embedded protection circuit detects a VBAT voltage higher than the battery cell voltage, it disables protection connecting the battery cell to VBAT. The battery pack starts charging causing VBAT voltage to rise slowly.
4. Once VBAT voltage is above VIN_POR_rise:
   a. The STPMIC1B initializes, asserts NRST and pre-load its NVM contents .
   b. As the STPMIC1B has “AUTO_TURN_ON” bit set by default in NVM, the STPMIC1B goes directly into “CHECK & LOAD” state as long as VBAT < VINOK_rise (see [2] for details).
5. Once VBAT voltage is above VINOK_rise, the STPMIC1B performs a power-up sequence then it releases the NRST signal (see Figure 5 for details about power-up sequence)
6. Once the STM32MP15x NRST is set, the STM32MP15x software starts execution and it initializes the application:
7. The STM32MP15x software initializes the USB charger IC:
   a. Stops the precharge
   b. Sets the fast charge with maximum charging current, maximum charging voltage and so on. The settings depend on the USB source current capability, the battery capacity and so on.
   c. As the battery is discharged (VBAT < max charging voltage setting) the USB charger IC charges the battery in constant current mode. The battery voltage increases slowly.

Note: Battery monitoring during charge is out of this document scope

8. Once the VBAT voltage is above the max charging voltage setting:
   a. The USB charger IC switches to constant voltage charging mode.
   b. The battery voltage kept stable and the charging current decreases during charge until it reaches the end of charge detection.

9. The cable from the USB Source peripheral is removed from the USB receptacle. Once the USB Type-C™ controller detects peripheral removal (see [7] for detail):
   a. The USB Type-C™ controller opens power switches (Q1/Q2), it triggers an interrupt to the STM32MP15x, it activates the VBUS discharge path and it goes into Unattached.SNK state.
   b. Once VBUS discharge path is activated, VBUS voltage is tied low.
   c. The VBUS_DR voltage drops. If battery was charging, the USB charger IC goes in high impedance state.
   d. The application runs and battery is discharging
10. Once the VBAT voltage is below the USB Type-C™ controller VSYS min voltage (see [7]), the USB Type-C™ controller is unpowered
11. Once the VBAT voltage is below the STPMIC1B VINLOW_rise thresholds:
   a. The STPMIC1B triggers an interrupt to the STM32MP15x to alert the STM32MP15x that battery voltage is low.
   b. The STM32MP15x software may enter in energy saving mode.
12. Once the VBAT voltage is below the STPMIC1B VINOK_fall threshold:
   a. A Turn-OFF condition occurs in the STPMIC1B.
   b. The STPMIC1B asserts an NRST and performs a power-down sequence (see Figure 5 for details about power-down sequence)
13. STPMIC1B is in OFF mode:
   a. Application is in Power-off mode.
   b. As long as VBAT voltage is below VINOK_rise, the STPMIC1B cannot power-up.

Figure 16. Power-up / power-down from/to dead battery mode



Revision history

Table 5. **Document revision history**

Date	Version	Changes
03-Feb-2020	1	Initial release.

Contents

List of figures

List of tables