STM32MP151/153/157 MPU lines and STPMIC1 integration on a wall adapter supply

Introduction

This application note applies to the STM32MP151/153/157 MPU line devices and are now referred to as STM32MP15x. The STPMIC1 device used for this application is the STPMIC1APQR.

This application note provides a sample hardware reference design based on STM32MP15x and STPMIC1APQR power management IC, powered from a 5V wall adapter.

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.
- Enhanced CPU frequency supply 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 STPMIC1APQR with DDR and flash memory and together with the following peripherals:

- DC input power source from main power supply: 5 V typ. (4V to 5.5V)
- Peripheral I/O interface voltage at 3.3 V supplied by the STPMIC1APQR.
- DDR3/DDR3L with x32 bits bus width
- A boot device which can be either eMMC, NAND or NOR powered from a 3.3 V power source and 3.3V I/O (for eMMC)
- 2 x USB HS Host port

Not covered by this application note:

- lpDDR2 and lpDDR3
- Peripheral interface with an I/O voltage of 1.8 V.

The battery-powered application using STPMIC1B are addressed in the *STM32MP151/153/157 MPU lines and STPMIC1 integration on a battery powered application* (AN5260).

- Type-C receptacle with dual role USB

The dual role feature requires power delivery support which is not in the scope of this application note.

In this document, MPU terminology refers to the STM32MP15x and the PMIC terminology is referring to STPMIC1APQR device.

2.1 Reference documents

Table 1. Reference documents

Document number	Title
STMicroelectronics documents	
[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 MPU lines and STPMIC1 integration on a battery powered application* (AN5260)
[5]	*STM32MP157 advanced Arm®-based 32-bit MPUs* (RM0436)
[6]	*Arm® dual Cortex®-A7 800 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)
[8]	*STM32MP1 Series lifetime estimates* (AN5438)
[9]	*STM32MP151, STM32MP153 and STM32MP157 discrete power supply hardware integration* (AN5256)
USB specification	
[10]	*USB Type-C cable and connector specification* release 1.4 or later available from USB implementation forum web site

Note: For STMicroelectronics documents refer to www.st.com.

3 Glossary

Table 2. **Glossary**

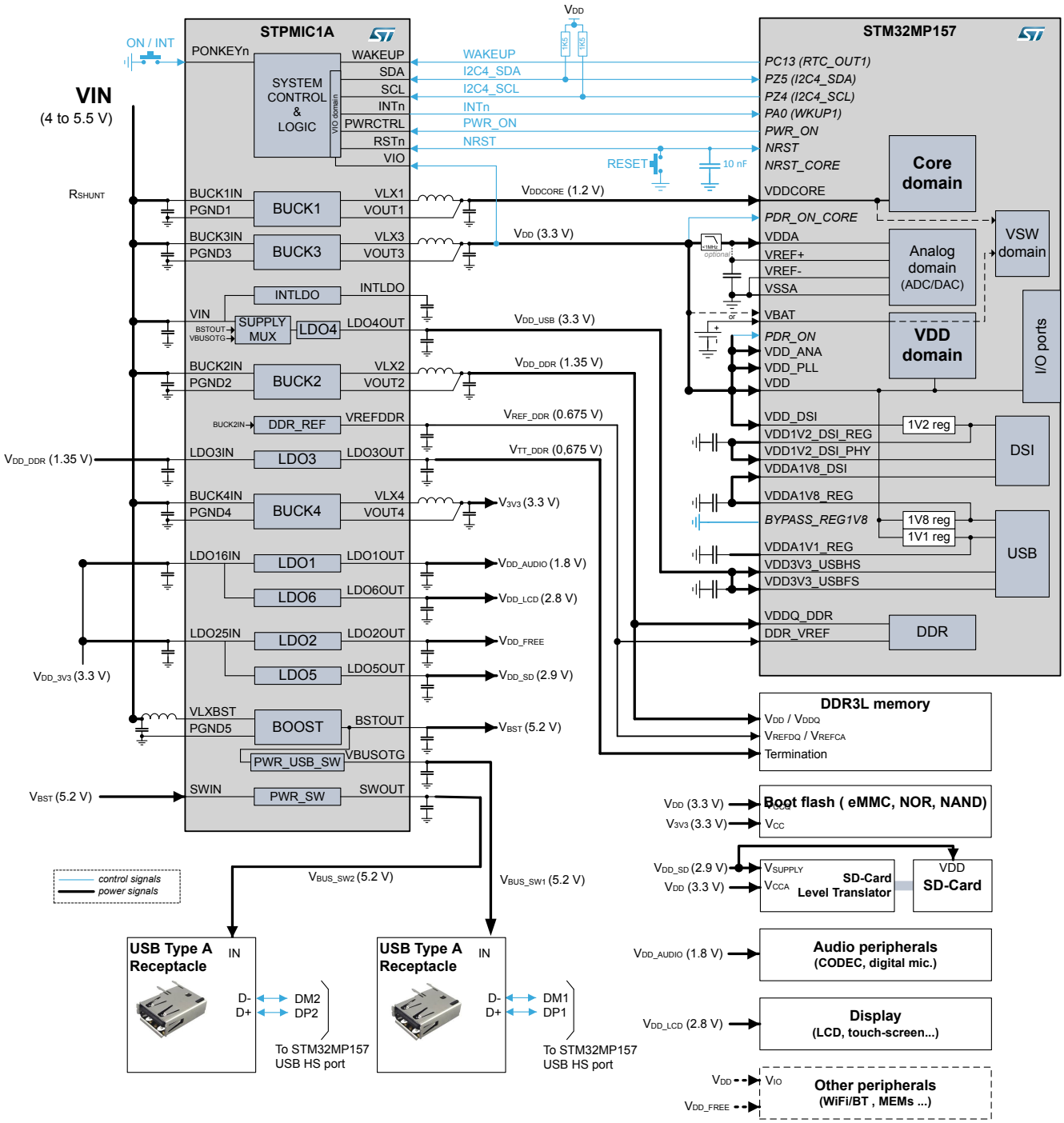
Term	Definition
FSBL	First stage boot loader
HSI	High speed internal oscillator
IC	Integrated circuit
LDO	Low drop out linear regulator
MPU	Microprocessor unit
PMIC	Power management integrated circuit
RTC	Real-time clock
SMPS	Switching mode power supply
SW	Software

4 Wall adapter supply application reference design

This reference design targets an application powered from a mains supply adaptor with DDR3L, a boot flash (either eMMC, NAND, NOR), an SD-card and two USB 2.0 HS host ports. Other peripherals like audio, display, wireless, MEMs are added to illustrate the application.

The main peripheral interfaces work with an I/O voltage of 3.3 V. The overall system is illustrated in Figure 1.

Figure 1. STM32MP15x and STPMIC1APQR with DDR3L, boot flash, SD-card and 2x USB HS



Note: The following are not shown in the diagram:

- *STM32MP15x decoupling scheme (see [1] related chapter),*
- *STPMIC1APQR discrete components value (see [2]),*
- *Additional protection, such as ESD, EMI filtering, overvoltage.*

4.1 Power distribution

The STPMIC1APQR integrates the regulators required to supply all the STM32MP15x in addition to a set of regulators to supply the application peripherals.

4.1.1 V_{DD} power domain (3.3 V)

V_{DD} is the reference design main I/O voltage domain used by the STM32MP15x, STPMIC1APQR and peripherals.

It is powered from the STPMIC1APQR BUCK 3 step-down SMPS, which has high efficiency and low quiescent current across all load conditions.

The V_{DD} voltage domain is the first voltage available during the power-up sequence (STPMIC1APQR Rank 1) and the last disabled during the power down sequence.

V_{DD} is enabled in run, LP-stop, LPLV-stop, and standby modes. V_{DD} is disabled in off mode.

STPMIC1APQR has dedicated NVM settings to set BUCK 3 (I/Os) at 3.3 V during the STPMIC1APQR power up to fit with a battery reference design in Figure 1. See [2] for details.

STM32MP15x V_{DD} power supply:

Connect V_{DD_PLL}, V_{DD_ANA}, and V_{DD_DSI} to V_{DD} in addition to supplying the STM32MP1 Series microprocessor V_{DD} I/O voltage domain.

ADC/DAC analogue voltage (V_{DDA}) and related analogue reference voltage (VREF+) may be powered from V_{DD} depending on expected ADC performances. If high ADC performances are expected or to ensure the DAC functions correctly, power V_{DDA} (and VREF+) from a low noise power source with a voltage greater than 1.8 V. In that case, a STPMIC1APQR LDO may be dedicated to supply the STM32MP15x V_{DDA} voltage domain.

4.1.2 V_{DDCORE} power domain (1.2 V / 1.35 V)

V_{DDCORE} is the main STM32MP15x digital power domain.

It is powered from the STPMIC1APQR BUCK1 step-down SMPS. This voltage domain is the next power domain to be available during power up sequence (STPMIC1APQR Rank 2) and the penultimate to be disabled during power down sequence.

V_{DDCORE} is enabled in run, LP-stop and LPLV-stop modes. V_{DDCORE} is reduced in LPLV-stop to save power. V_{DDCORE} is disabled in standby and in off mode.

STPMIC1APQR has NVM settings to configure BUCK1 (V_{DDCORE}) to 1.2 V when STPMIC1APQR is in power up. The STM32MP15xD and STM32MP15xF devices have an enhanced consumer mission profile (see [8]). This profile allows the Arm® dual Cortex®-A7 CPUs to the clock frequency run up to 800MHz (see [6] for details and limitations).

Accordingly, the V_{DDCORE} supply voltage must be increased to 1.35 V when the CPU frequency (Fmpuss_ck) operates up to 800 MHz . When it does not operate in run mode (800 MHz), the V_{DDCORE} supply voltage must be set back to its nominal voltage (1.2 V typ).

4.1.3 V_{DD_USB} power domain (3.3 V)

V_{DD_USB} is dedicated to supplying power to the STM32MP15x USB PHYs (V_{DD3V3_USBHS} and V_{DD3V3_USBFS}).

It is powered from the STPMIC1APQR LDO4 linear regulator, which has been specifically designed for this feature.

This voltage domain is the last domain available during power up sequence (STPMIC1APQR Rank 3) and the first to be disabled during power down sequence (except regulators enabled by software that are disabled before LDO4 in Rank 0. See [2] for details).

V_{DD_USB} regulator is managed by software and is turned on or off at run time. V_{DD_USB}, V_{BUS_SW1}, and V_{BUS_SW2} are disabled in standby, power off, and VBAT mode.

For the USB flashing use case (STM32MP13x peripheral boot from ROM), LDO4 (V_{DD_USB}) is needed at power up to supply the USB PHY. LDO4 has a 3.3 V fixed voltage.

4.1.4 V_{DD_DDR} (1.35 V), V_{REF_DDR} (0.675 V), V_{TT_DDR} (0.675 V) power domains

- V_{DD_DDR} is dedicated for DDR3/DDR3L core power supply (VDD) at 1.35 V.
- V_{REF_DDR} is dedicated for DDR3/DDR3L reference voltage (VREFQ / VREFCA) and for STM32MP15x DDR reference voltage (DDR_VREF) at V_{DD2_DDR} / 2 (0.675 V).
- V_{TT_DDR} is dedicated for fly by topology termination, mandatory for 32 bits dual DDR3. (optional for single 16 bits)
- V_{DD_DDR} (1.35 V) is powered from the STPMIC1APQR BUCK2 step down SMPS having high efficiency and low quiescent current in any load conditions. BUCK2 is powered from V_{IN}.
- V_{REF_DDR} (0.675 V) is powered from the STPMIC1APQR REFDDR sink/source LDO. When enabled REFDDR output voltage is equal to V_{DD_DDR} / 2 (BUCK2 output voltage / 2).
- V_{TT_DDR} is powered from the STPMIC1APQR sink/source LDO3. When enabled LDO3 output voltage is equal to V_{DD_DDR} / 2 (BUCK2 output voltage / 2).

The STPMIC1APQR does not start V_{DD_DDR}, V_{REF_DDR} and V_{TT_DDR} at power up. They must be powered up and powered down by STM32MP15x software, respectively at STM32MP15x boot up and shutdown.

DDR3 boot sequence recommended by JEDEC is the following:

1. Enable DDR_REF LDO (V_{REF_DDR})
2. Set BUCK2 at 1.35 V and enable BUCK2 (V_{DD_DDR})
3. After 100 µs, set LDO3 in sink/source ($V_{OUT2/2}$) mode and enable LDO3.(V_{TT_DDR})

Steps 1, 2 and 3 can be inverted since LDO3 and DDR_REF LDO output rise to VDD_DDR/2 once BUCK2 is enabled.

4.1.5 V_{3V3} power domain (3.3 V)

V_{3V3} is powered from the BUCK4 step-down SMPS.

It is a generic purpose power supply which intend to be used instead of V_{DD} for following usage:

1. Power supply boot flash device (eMMC, NAND, NOR) core domain (V_{CC}).
   V_{3V3} power domain :
   - is disabled during reset power cycle, complying with flash reboot requirement.
   - is automatically enabled at boot
   - is disabled in standby and off mode.

   V_{3V3} can be disabled anytime by software in run, LP stop and LPLV stop if no access to powered peripheral is expected (power saving).
   V_{3V3} must be enabled prior to Standby (in the STPMIC1APQR BUCK4_MAIN_CR register) as detailed in [2], so that it can be disabled in standby (PWR_ON signal low) and be enabled back when the MPU leaves standby mode (PWR_ON signal high).
2. LDO16IN, LDO25IN pre-regulation
   In case LDOs are not expected to provide more than 3.0 V as output, supplying them with pre-regulated 3.3 V enhances the LDOs power efficiency by saving thermal dissipation.
   In that case, V_{3V3} can be disabled to save power, but only when all the connected peripherals are disabled also.
   This voltage domain is the second one to be enabled during power up sequence (STPMIC1APQR Rank2) and it is the before-last disabled during the power down sequence.
   The STPMIC1APQR has NVM settings to set BUCK4 at 3.3 V when it is powered up.

4.1.6 V_{DD_SD} power domain (2.9 V)

V_{DD_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 V_{DD} voltage to supply its IOs power domain (VCCA).

It is powered from the STPMIC1APQR LDO5 linear regulator.

This voltage domain is the second domain to be enabled during the power up sequence (STPMIC1APQR Rank2) and it is the penultimate to be disabled during power down sequence.

V_{DD_SD} regulator is managed by software and can be turned on or turned off at run time. The software must disable V_{DD_SD} if no R/W access is expected. V_{DD_SD} is disabled in standby and in off modes.

STPMIC1APQR has NVM settings to set LDO5 (V_{DD_SD}) to 2.9 V when the STPMIC1APQR powers up. LDO5 (V_{DD_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.

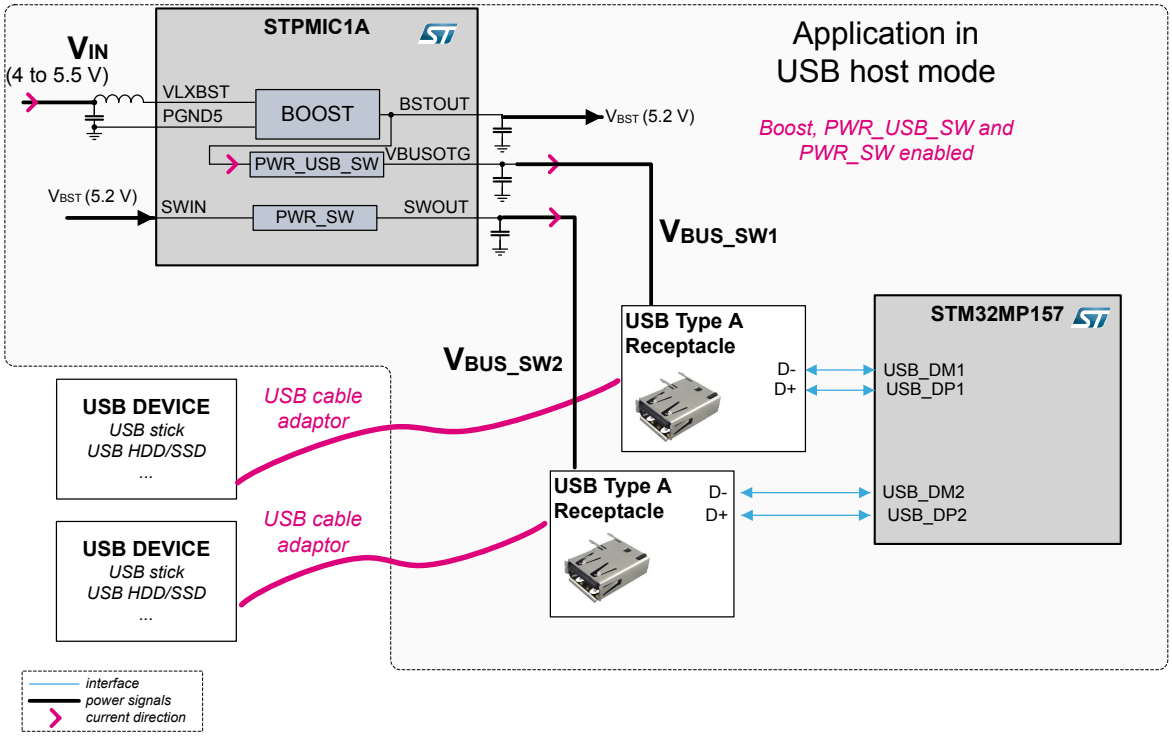
If the SD-card device is the boot flash peripheral, the application software must program the STPMIC1APQR to power off the SD-card in standby mode (PWR_ON signal low) and power on the SD-card in run mode (PWR_ON signal high) before the application goes into standby mode. In such case, when the application recovers from standby mode to run mode, the SD-card is powered and ready to be accessed by the STM32MP15x bootROM (peripheral boot).

4.1.7 V_{BUS_SW1} and V_{BUS_SW2} power domain (5.2V)

V_{BUS_SW1} and V_{BUS_SW2} are dedicated power domains for the two USB high speed host interfaces. They are connected to the IN pin of the USB type-A receptacles.

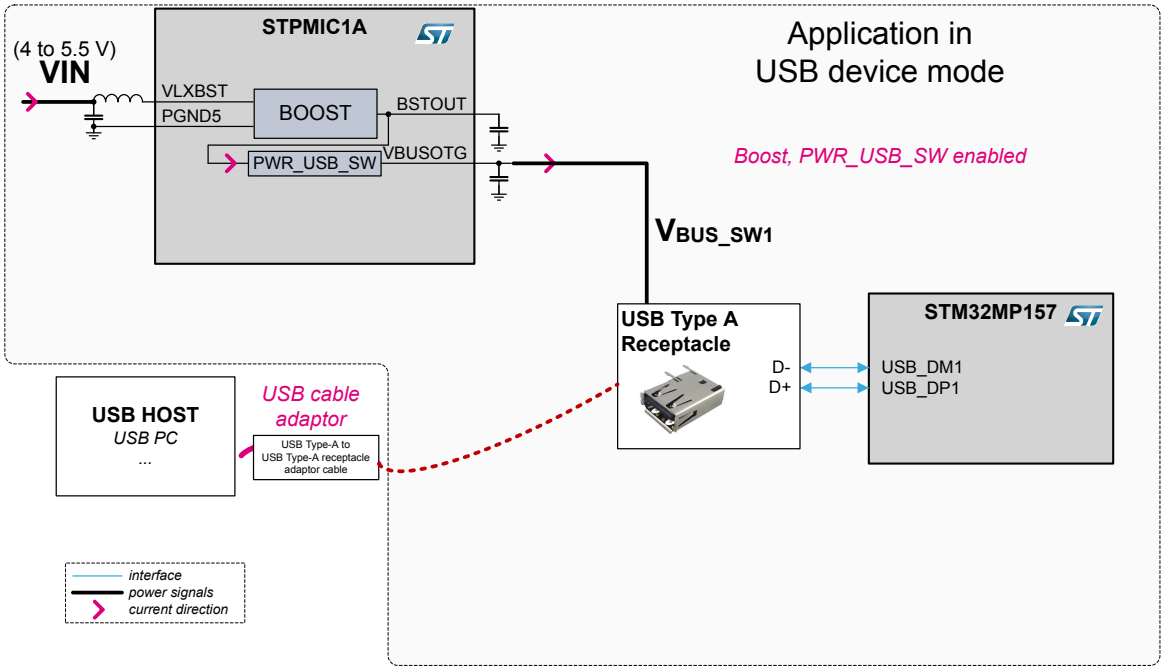
- Two USB device peripherals are connected to the application and are powered from V_{BUS_SW1}/V_{BUS_SW2} from the STPMIC1A (see Figure 2)
- There is no need for external power supplies, the STPMIC1A delivers compliant USB VBUS voltages from the VIN wall supply (4 V to 5.5 V). This is done thanks to the PMIC boost SMPS that regulates V_{BUS} to 5.2 V with V_{IN} from 4 V to 5.5 V.
- When at least one of the V_{BUS_SW1}/V_{BUS_SW2} is enabled, V_{DD_USB} supply must also be enabled in order to enable USB PHY operation.
- Software manages boost at run time.

Figure 2. V_{BUS_SW1}, V_{BUS_SW2} power path in USB host mode



Flashing through USB with Type-A connector

Figure 3. Flashing through USB with Type-A connector



For applications that have only Type-A receptacle USB port, it is still possible to perform serial boot over the USB on the STM32MP15x system. This can be done in USB device mode.

This specific boot mode is different from the classical boot method that uses the USB Type-B receptacle. To support this specific boot mode, there are two requirements:

- A USB Type-A receptacle is used in USB device mode
- A dedicated non USB-compliant Type-A to Type-A plug cable is required.

The host PC USB Type-A receptacle must be connected on one side of the dedicated cable while the other side is connected to the STM32MP15x device USB Type-A receptacle.

The STM32Cube Programmer is used on the host PC to flash the Linux distribution on the target STM32MP15x device.

The STM32MP15x must be ready for USB/UART boot:

- Either the flash memory is empty and the device is automatically set to USB/UART boot
- Or boot pins must be set to USB/UART boot mode (BOOT[2:0]=0b000 or 0b110)

In this specific use case, the VBUS_SW1 signal is not connected to the STM32MP15x hence the boot ROM does not probe V_{BUS} to detect a host PC connection.

During the flashing operation, the boost and PWR_USB_SW must be kept de-activated.

1. Initial conditions:
   - PC ready to enumerate USB DFU
   - Boot pin set to USB/UART mode if flash is not empty
   - Board power supply is OFF
   - USB cable connection between PC host and STM32MP15x board
2. Power supply is switched ON (or reset of the STM32MP15x device)
3. PC enumerates USB DFU
4. Flashing starts with STM32Cube programmer

Once flashing is complete, the USB cable between the host PC and the STM32MP15x device must be disconnected. This must be done before booting the system in application mode from the flash.

4.1.8 V_{DD_AUDIO} (1.8 V), V_{DD_LCD} (2.8 V) power domains

V_{DD_AUDIO} and V_{DD_LCD} are provided to illustrate the reference design in Figure 1. They are respectively powered from LDO1 and LDO6 linear regulators.

V_{DD_AUDIO} and V_{DD_LCD} are not enabled by the STPMIC1APQR at power-up. They are enabled and set at the right voltage after power-up by software when the related peripheral requires them.

V_{DD_AUDIO} and V_{DD_LCD} are managed by software and can be turned on or off at runtime. Software can disable V_{DD_AUDIO} and V_{DD_LCD} if no peripheral access is expected. V_{DD_AUDIO} and V_{DD_LCD} are disabled in standby, and in off modes.

4.2 Control signals and interface between STM32MP157 and STPMIC1APQR

This section outlines the way the STM32MP157 microprocessor communicates with the STPMIC1APQR device. There are several interface choices which can be used depending on the application requirements. Each interface is described in this first part.

I^2C interface:

The STPMIC1APQR can be controlled by the STM32MP15x via the I^2C 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.

The STPMIC1APQR has special default NVM settings that allows it to boot an STM32MP15x application with 3.3 V I/Os 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 STPMIC1APQR NVM on the fly to fine tune the application.

ON / INT push button:

The user “ON / INT” push button is connected to the STPMIC1APQR PONKEYn pin (active low). This button allows:

- To power up the STPMIC1APQR.
- 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 STPMIC1APQR with a long press (16 s by default).

NRST signal:

The NRST is a bidirectional active low signal for the STM32MP15x and the STPMIC1APQR. The STM32MP15x NRST pin and STPMIC1APQR RSTn pin are of digital input / open drain output topology:

- When STPMIC1APQR 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 STPMIC1APQR releases the NRST.
- When the STM32MP15x asserts an NRST signal (such as an STM32MP15x watchdog reset) or a user presses on the “RESET” button, STPMIC1APQR immediately asserts the RSTn pin and performs a non-interruptible power-cycle: the STPMIC1APQR performs a power down sequence followed by a power up sequence and finally releases the RSTn .
  At the end of power-cycle sequence, STPMIC1APQR waits for the NRST signal to go high before rearming the reset to avoid infinite reset loop.

INTn signal:

The INTn is a STPMIC1APQR output active low interrupt line connected to the STM32MP15x PA0 input pin. PA0 has both interrupt and wakeup capability:

- To manage interrupts from the STPMIC1APQR 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 STPMIC1APQR PWRCTRL pin.

This allows the STM32MP15x to switch the STPMIC1APQR 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 STPMIC1APQR)

After a power-up or a reset, the STPMIC1APQR PWRCTRL pin is disabled. Before going in low-power mode, the STM32MP15x sets the STPMIC1APQR via I^2C to program the expected power behavior according to the PWR_ON signal state.

EADLY timer

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)

POPL timer

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 STPMIC1APQR 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.

WAKEUP signal (optional):

The WAKEUP signal is driven by the STM32MP15x PC13 (RTC_OUT) pin to control the STPMIC1APQR WAKEUP pin. It allows the STM32MP15x to power up the STPMIC1APQR; typically when real time clock timer elapses.

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

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 STM32MP15x and they control the power management and the clock distribution (see details in [3]).

Table 3 summarizes the application level operating modes. The STPMIC1APQR power modes depend on the application operating mode.

LPLV stop mode is not covered in this document. Refer to reference [2] for its application.

Table 3. Application operating modes

Operating mode	STPMIC1APQR Power mode	V_{IN}(1)	PWR_ON	Description	Notes
Run	Power-on-main	> VINOK_fall	1	V_{DD} power on V_{DDCORE} power on, system clock on DDR3 active / auto refresh Peripherals power on / off	(2)
Stop	Power-on-main	> VINOK_fall	1	V_{DD} power on V_{DDCORE} power on, system clock off DDR3 auto refesh/ self refresh V_{TT_DDR} off Peripherals power on / off	(2)
LPLV stop	Power-on-alternate	> VINOK_fall	0	V_{DD} power on V_{DDCORE} power on at lower voltage, system clock off DDR3 auto refesh/ self refresh Peripherals power on / off	(3)
Standby	Power-on-alternate	> VINOK_fall	0	V_{DD} power on V_{DDCORE} power off, system clock off DDR3 self refesh/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. *STPMIC1APQR 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 STPMIC1APQR via I^2C 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 V_{DD} 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 STPMIC1APQR into run mode. Similarly , if the application needs to go into power-off mode, a turn-off condition is required to power-down the STPMIC1APQR.

The STPMIC1APQR 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).

By definition:

- Power-up is the STPMIC1APQR transitional power up state where the regulators start sequentially in a predefined order (rank) and voltage, and ends with the release of the NRST signal. After this state, the STPMIC1APQR goes into power-ON state and remains there, the application is now able to run. This state is reached from off mode with a turn-on condition or from NO_SUPPLY with V_{IN} voltage rising higher than VINOK_rise (AUTO turn-on).

- Power-down is the STPMIC1APQR 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 STPMIC1APQR 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 STPMIC1APQR 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 STPMIC1APQR automatically powers up when the supply voltage (V_{IN}) rises above VINOK_rise (an AUTO turn-on feature enabled by default in STPMIC1APQR NVM). If the STPMIC1APQR is in the off state (and V_{IN} > VINOK_rise), it can be powered up by one of three external triggers:

- "ON / INT" user button press: PONKEYn pin voltage falling 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.

On power-up, the STPMIC1APQR undergoes a 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 STPMIC1APQR 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 V_{IN} voltage rising higher than VINOK_rise (AUTO turn-on)

Turn-off conditions

A turn-off condition leads the STPMIC1APQR to power down and go into the off state. In the off state, all regulators are turned off. If the STPMIC1APQR is in the on state, it can be powered down by one of six conditions:

- Software switch-off: I^2C command sent by the STM32MP15x to the STPMIC1APQR.
- "ON / INT" user button long press: when the reset button is pressed for 16 s, the STPMIC1APQR is turned off (the delay is programmable).
- Thermal shutdown: if overheating, STPMIC1APQR 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 STPMIC1APQR shut down.
- Watchdog: if enabled by software, when the countdown timer reaches 0, the STPMIC1APQR goes to the off state.
- VINOK_fall: if V_{IN} goes below VINOK_fall threshold, the STPMIC1APQR goes to the off state.

On power-down, the STPMIC1APQR undergoes a 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 STPMIC1APQR 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.

It is important to point out that as soon as a STPMIC1APQR 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).

An application can set the STPMIC1APQR "restart request" feature to automatically restart the application after a turn-off condition (see [2] for details).

5.1.2 STPMIC1APQR restart_request and mask_reset options

Before a turn-off condition occurs, the STM32MP15x software can program the STPMIC1APQR to restart instead of turning it off by setting the restart_request feature in the STPMIC1APQR. 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 STPMIC1APQR (RREQ_EN = 1) then to program a software switch off (SWOFF = 1). The STPMIC1APQR 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 STPMIC1APQR regulators to be kept enabled during a power cycle, the STM32MP15x software can program the STPMIC1APQR mask_reset option by setting the STPMIC1APQR BUCKS_MRST_CR register to target the buck converter and LDOS_MRST_CR register to target LDOs (see [2] for details on the STPMIC1APQR 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 V_{DD} power domains. The power cycle on V_{DD} 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 User reset and crash recovery management) 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 STPMIC1APQR 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 4 based on the reference design in .

5.2.1 Power-up by main supply plugin

The application is in off condition and has no power connected to it. A supply is connected up and the application starts automatically when VIN rises (the STPMIC1APQR has the auto-turn ON enabled by default in its NVM). When the STPMIC1APQR is powered-up, the application boots (including the DDR initialization) and finally the system reaches run mode. When a turn-off condition occurs, the STPMIC1APQR 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 4:

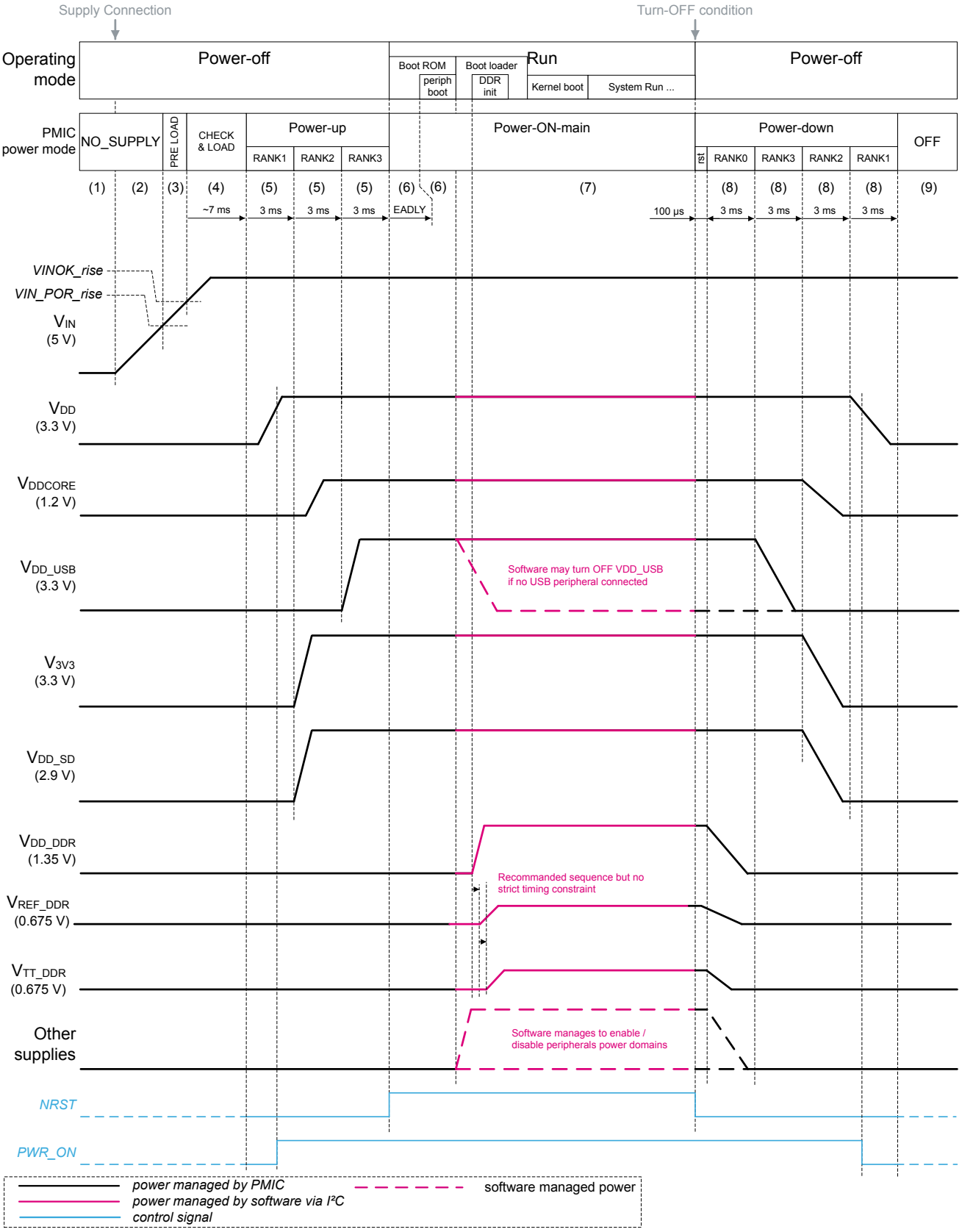
1. Application has no power or the STM32MP15x is in coin-cell-VBAT mode (powered from coin cell battery to supply the STM32MP15x V_{SW}).
2. A compliant Power supply (V_{IN} > VINOK_rise) is connected to the application. V_{IN} voltage rises.
3. Once V_{IN} supply is above VIN_POR_rise (STPMIC1APQR VIN_POR_rise threshold is initially set to 2.3 V):
   a. The STPMIC1APQR initializes and pre-loads its NVM contents.
   b. The STPMIC1APQR asserts the NRST.
4. V_{IN} supply rises above VINOK_rise, the STPMIC1APQR checks the turn-on condition (auto turn-on is enabled in the STPMIC1APQR NVM). The STPMIC1APQR starts a power up sequence as a valid turn-on condition is detected.
5. The STPMIC1APQR follows the power-up sequence:
   a. Rank1: BUCK3 (V_{DD}) is enabled at 3.3 V and waits for 3 ms.
   b. Rank2: BUCK1 (V_{DDCORE}) is enabled at 1.2 V. LDO5 (V_{DD_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 STPMIC1APQR is still asserting its NRST signal.
   c. Rank3: LDO4 (V_{DD_USB}) is enabled at 3.3 V (hard setting). After 3 ms, the STPMIC1APQR releases the NRST signal.
6. As the NRST signal rises, both the STM32MP15x and the STPMIC1APQR release their respective reset pins:
   a. The STM32MP15x EADLY delay timer (10 ms) starts. Refer to EADLY timer section for more information.
   b. When the EADLY delay 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 STPMIC1APQR 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. BUCK2 (V_{DD_DDR}) is enabled at 1.35 V.
   b. DDR_VREF (V_{REF_DDR}) is enabled.
   c. LDO3 (V_{TT_DDR}) is enabled at 0.675 V. The software waits for at least 1.4 ms for BUCK2 ready.
   d. The software initializes the STM32MP15x DDR controller and DDR3 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 STPMIC1APQR performs a power-down sequence:
   a. The STPMIC1APQR asserts the NRST (STM32MP15x reset) and waits for 100 µs.
   b. Rank0: the STPMIC1APQR 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. See also Turn-off conditions section.

Note: As soon as a STPMIC1A 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 (V_{DD_USB}) is disabled and waits for 3 ms.
   d. Rank2: BUCK1 (V_{DDCORE}), LDO2 (V_{3V3}) and LDO5 (V_{DD_SD}) are disabled and waits for 3 ms.
   e. Rank1: BUCK3 (V_{DD}) is disabled waits for 3 ms.
9. The STPMIC1APQR is now in off mode: the application is in power-off.

Figure 4. Power-up / power-down sequence



5.2.2 Power-up from the STPMIC1APQR off mode

The application in Figure 4 is powered up from NO_SUPPLY state, where the wall supply V_{IN} connection is the turn-on condition (auto turn-on is enabled in the STPMIC1APQR NVM).

A power-up from the STPMIC1APQR off mode follows a similar sequence as in Figure 4 from off mode. The difference is when a turn-on condition occurs, the sequence starts from step (4) “ CHECK & LOAD” instead of waiting for V_{IN} to rise.

The differences are detailed below:

- The STPMIC1APQR is initially powered from a V_{IN} voltage higher than VINOK_rise allowing the STPMIC1APQR to power-up (instead of a supply plug-in which triggers V_{IN} rise in Figure 4).
- Steps (1), (2), (3) of Figure 4 are replaced by a single one; merging “NO-SUPPLY” and “PRE-LOAD” of the STPMIC1APQR power modes to “OFF”.
- The “Supply connection” event is replaced by the “power-ON condition” event and is placed between “OFF” and “CHECK & LOAD” of the STPMIC1APQR state.

5.2.3 Power-down by mains supply removal

The application in Figure 4 is powered off by a turn-off condition with V_{IN} maintaining a valid voltage.

If the application is powered off by a supply removal, the turn-off condition is V_{IN} dropping below VINOK_fall. Once V_{IN} supply is below VINOK_fall, the STPMIC1APQR asserts an NRST for 100µs then powers-down as shown in step (8) onwards in Figure 4.

Limitation: When the supply is removed, V_{IN} 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 STPMIC1APQR asserts an NRST, system activity is immediately stopped and power consumption drops, slowing V_{IN} drop. Nevertheless, V_{IN} may drop below VIN_POR_fall threshold before the power-down sequence ends. In this case, the STPMIC1APQR regulators pull-down discharge resistors are no longer controlled by the STPMIC1APQR. A bulk decoupling capacitor (a few hundred µF) may be inserted on V_{IN} path to limit V_{IN} dropping speed.

5.3 Low-power mode management

The STM32MP15x supports several operating modes to reduce power consumption (see Section 5.1). This section describes the LP 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 STPMIC1APQR PWRCTRL input pin. The STPMIC1APQR states can then be switched: power-on-main to power-on-alternate and vice versa.

After power-up, the STPMIC1APQR goes into power-on-main until the STM32MP15x tells the STPMIC1APQR to activate PWRCTRL pin in active low by setting PWRCTRL_POL = 0 and PWRCTRL = 1 in STPMIC1APQR MAIN_CR register.

Table 3 summarizes the STPMIC1APQR states according to PWR_ON signal with related operating modes.

When the STPMIC1APQR 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).

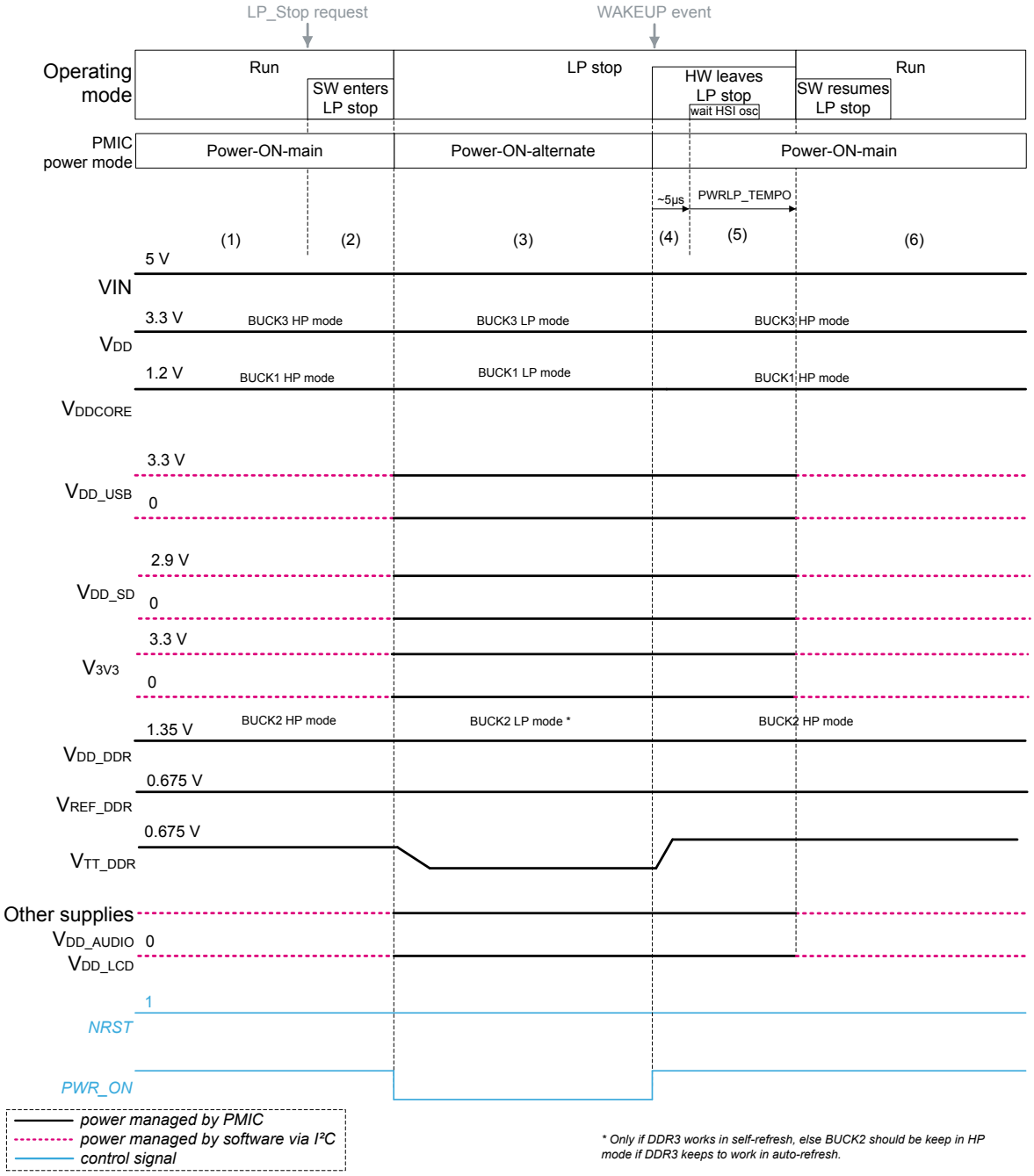
Before entering in low-power mode, the STM32MP15x must set the STPMIC1APQR ALTERNATE control registers in line with the expected STPMIC1APQR regulator settings for low-power mode behavior. If needed, the STM32MP15x must set the STPMIC1APQR MAIN control registers to guarantee that the application leaves the low-power mode.

5.3.1 LP-stop mode

The application LP-stop mode sequence is shown in Figure 5 based on to the implementation shown in Figure 1.

1. The application is powered up and works in run operating mode; the STPMIC1APQR is in power-on-main state.
2. When the LP-stop operating mode is requested, the software prepares LP-stop process:
   a. the STM32MP15x settings such as: stopping some clocks, setting DDR to Self-Refresh, setting PWRLP_TEMPO.
   b. STPMIC1APQR settings:
      - BUCK1 (V_{DDCORE}), BUCK2 (V_{DD_DDR}), BUCK3 (V_{DD}): ON HP in main mode and ON LP in alternate mode.
      - LDO3 (V_{TT_DDR}) ON HP in main mode and OFF LP in alternate mode.
      - LDO5 (V_{DD_SD}), LDO4 (V_{DD_USB}), BUCK4 (V_{3V3}) 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 LP stop: PWR_ON signal is de-asserted when the STM32MP15x enters LP stop. The STPMIC1APQR goes in Power-on-alternate state:
   – LDO3 voltage is turned off
   – All BUCK regulators go into low-power mode.
4. On a wakeup event, the STM32MP15x leaves LP stop mode and asserts a PWR_ON signal:
   a. The STPMIC1APQR goes in power-on-main state:
   b. LDO3 (V_{TT_DDR}) turns-on in max 20 µs. In parallel, all BUCK converters go from low-power to high-power mode (100 µs).
5. Once the STM32MP15x HSI clock oscillator is stable (~5 µs), the PWRLP_TEMPO timer is timed out to waiting for peripheral to be stable.
   Where PWRLP_TEMPO is an STM32MP15x dedicated timer designed to wait for the regulator recovery when the application goes from LP mode to run mode. PWRLP_TEMPO must be set to 100 µs to let the STPMIC1APQR regulators recover from LP mode to HP mode.
6. When PWRLP_TEMPO elapses, the application goes into run mode. The software resumes LP stop (restores clocks, resumes lpDDR from self-refresh, and so on).

Figure 5. LP stop mode sequence



5.3.2 Standby mode

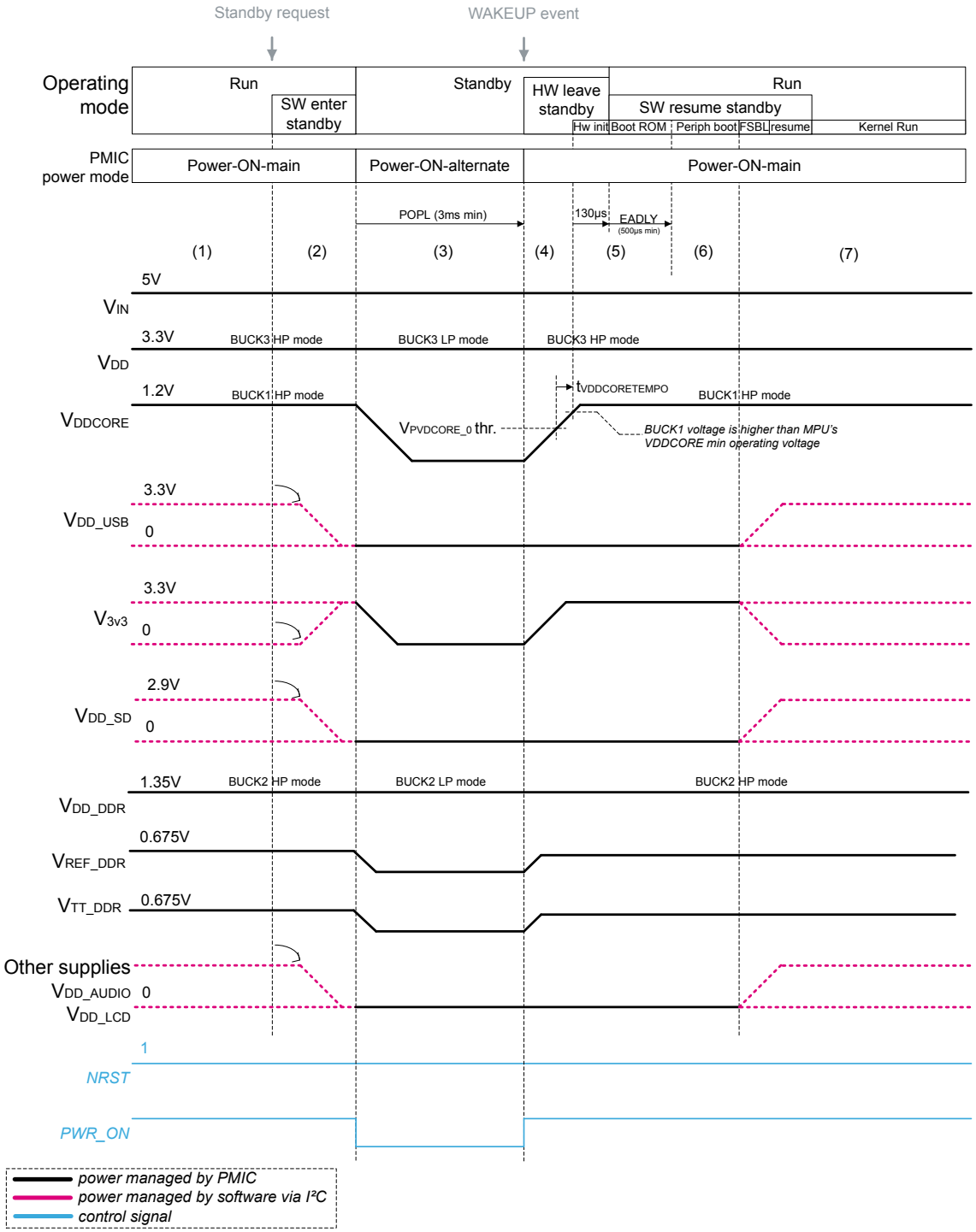
The application standby mode sequence is shown in Figure 5 according to the implementation shown in Figure 1.

In this application, the eMMC flash memory voltage (V_{DD_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 STPMIC1APQR 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
        For details on the POPL refer to POPL timer.
        For details on the EADLY timer refer to the EADLY timer.
      - Timers and so on.

   b. The STPMIC1APQR settings:
      i. BUCK1 (V_{DDCORE}), BUCK4 (V_{3V3}), LDO3 (V_{TT_DDR}): ON HP in main mode, OFF in alternate mode
      ii. LDO4 (V_{DD_USB}), LDO5 (V_{DD_SD}), LDO1 (V_{DD_AUDIO}), LDO6 (V_{DD_LCD}): OFF in main and alternate modes.
      iii. BUCK2 (V_{DD_DDR}), BUCK3 (V_{DD}) 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 STPMIC1APQR goes in Power-on-alternate state:
      i. BUCK1 (V_{DDCORE}), BUCK4 (V_{3V3}), LOD3 (V_{TT_DDR}): regulators are powered OFF
      ii. BUCK2 (V_{DD_DDR}), BUCK3 (V_{DD}) regulators go in LP mode
4. On a wakeup event, the STM32MP15x leaves standby mode and asserts a PWR_ON signal:
   a. The STPMIC1APQR goes in power-on-main state:
      i. BUCK1 (V_{DDCORE}), BUCK4 (V_{3V3}), LOD3 (V_{TT_DDR}): regulators are powered ON
      ii. BUCK2 (V_{DD_DDR}), BUCK3 (V_{DD}) regulators go into HP mode
   b. When V_{DDCORE} voltage is above the VPVDCORE_0 min threshold , a $t_{VDDCORETEMPO}$ is started. As long as the $t_{VDDCORETEMPO}$ timer is not elapsed, the STM32MP15x is kept in reset internally.
      The STM32MP15x internal voltage threshold, VPVDCORE_0 rising edge, is 0.95 V min.
      The STM32MP15x internal delay. $t_{VDDCORETEMPO}$ is 200 µs min.
5. When $t_{VDDCORETEMPO}$ elapses, the STM32MP15x is taken out of internal reset (V_{DDCORE_OK}):
   a. V_{DDCORE} voltage is higher than STM32MP15x V_{DDCORE} min operating voltage.
      BUCK1 has a 2.3 mV/µs minimum slew rate guarantying the V_{DDCORE} voltage is higher than the STM32MP15x V_{DDCORE} min operating voltage when $t_{VDDCORETEMPO}$ 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, V_{3V3} voltage is stable:
   a. The boot ROM is read from the flash memory, verifies and executes the FSBL.
   b. From this step, the software can set the STPMIC1APQR via I^2C interface to set any regulator.
7. The software detects an “exit from standby mode” it then resumes and runs the Kernel software.

Figure 6. Standby mode sequence



5.4 User reset and crash recovery management

As introduced in Section 4.2 , the STM32MP15x and the STPMIC1APQR 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 STPMIC1APQR that triggers an immediate power cycle sequence: a STPMIC1APQR power-down sequence followed by a STPMIC1APQR 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 STPMIC1APQR BUCK3 (VDD) that needs to be remain enabled during reset (see Section 5.1.2 STPMIC1APQR restart_request and mask_reset options for details of the STPMIC1APQR mask_reset option).

If the reset button is pressed by a user, the same power cycle sequence is performed by the STPMIC1APQR.

5.4.1 Crash recovery management or user reset sequence

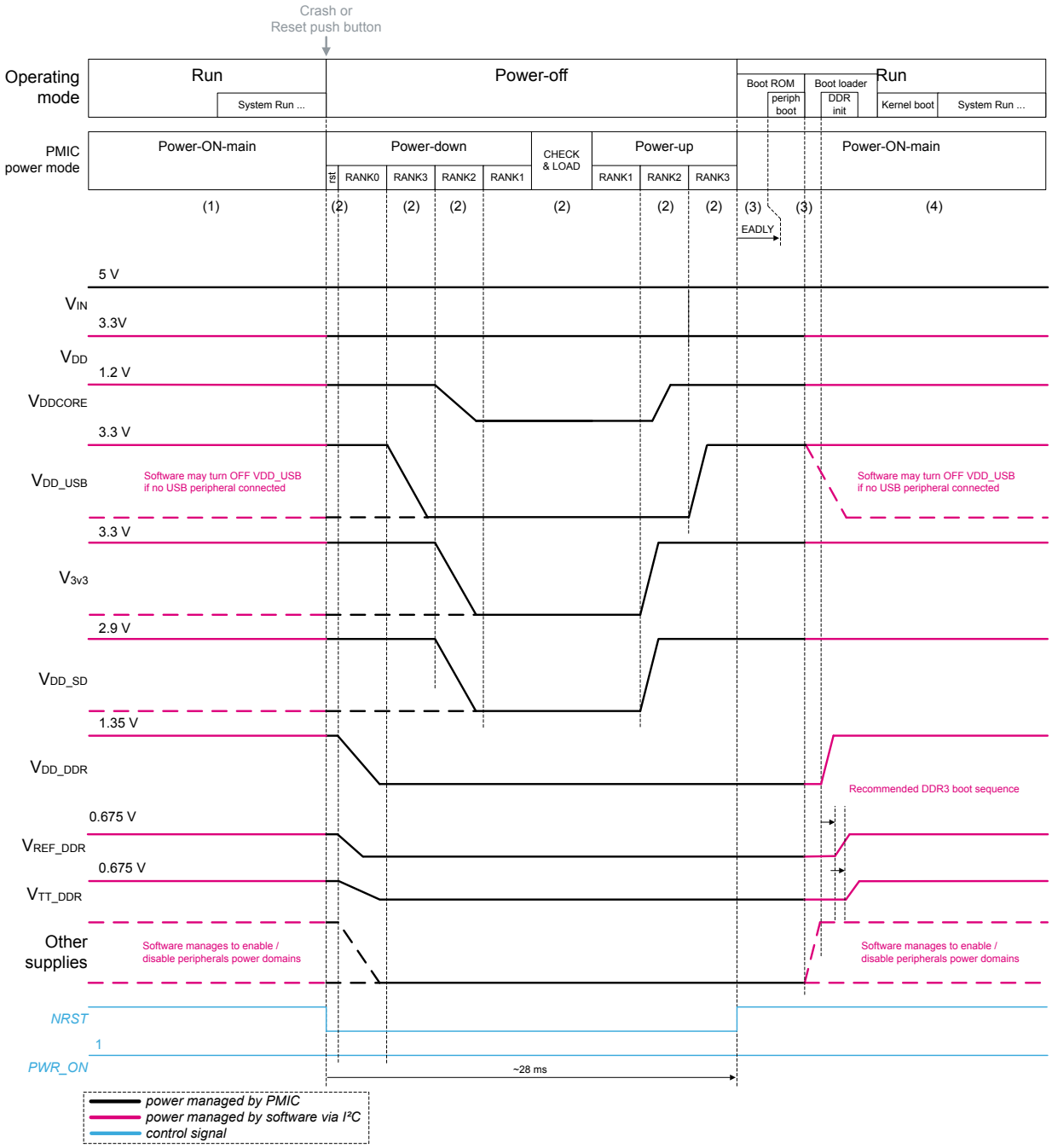
The sequence in Figure 7 illustrates a crash recovery sequence according to the implementation shown in Figure 1. In this sequence, the crash happens in run mode (by IWDG reset). Neverthless, a IWDG reset could occur in all modes, including: run, stop, LP stop, LPLV stop, and standby modes.

1. The application is powered up and is in run mode; the STPMIC1APQR 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 STPMIC1APQR detects the reset assertion (NRST pulse low) and starts a non-interruptible power cycle :
   a. The STPMIC1APQR asserts NRST low.
   b. The STPMIC1APQR performs power-down sequence.
   c. The STPMIC1APQR checks the conditions to restart (such as V_{IN}, temperature and so on) and reloads the internal NVM.
   d. The STPMIC1APQR performs power-up sequence.
   e. The STPMIC1APQR releases NRST.
   If the reset signal (NRST) is still asserted at this step (such as the user is still pressing reset button), the STPMIC1APQR waits for the reset signal to be released before rearming the reset circuit. This is to avoid the STPMIC1APQR repeating a power cycle loop.

Note: STPMIC1APQR power cycle duration is ~28 ms.

3. The NRST signal rises as the STM32MP15x and the STPMIC1APQR release their respective reset pins (and reset button released):
   a. The STM32MP15x EADLY delay timer (10 ms) is started. For information on the EADLY timer refer to EADLY timer.
   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 STPMIC1APQR 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. BUCK2 (V_{DD_DDR}) is enabled at 1.35 V.
   b. DDR_VREF (V_{REF_DDR}) is enabled.
   c. LDO3 (V_{TT_DDR}) is enabled at 0.6 V. The software waits for at least 1.4 ms for BUCK2 to be ready.
   d. The software initializes the STM32MP15x DDR controller and DDR3 device.
   e. The boot loader loads and executes the kernel and the kernel initializes.
   f. System is running.

Figure 7. Crash recovery sequence



5.5 Software management example

This section presents one possible software integration strategy of the STPMIC1APQR management by the STM32MP15x.

The OpenSTLinux software distribution integrates the way the STPMIC1APQR regulators are driven and configured by the STM32MP15x to match operating mode presented in Section 5.1 Operating modes.

To summarize the main OpenSTLinux integration points:

- Interface with the STPMIC1APQR 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 STPMIC1APQR power management strategy presented in Table 4. STPMIC1APQR power management options below is configured in Secure Monitor dts file (typ TF-A) using Linux Regulator framework binding terminology.
- Each STPMIC1APQR 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 .

Table 4. STPMIC1APQR power management options

Application Power Domain	STPMIC1A supply source	Power State				Options/ Comments
		Default Power On State (NVM)	STM32MP15x = Run / Stop PMIC= POWER_ON Main	STM32MP15x = LP-Stop PMIC= POWER_ON Alternate	STM32MP15x = STANDBY DDR OFF PMIC= POWER_ON Alternate	
V_{DD}	BUCK3	3.3 V	3.3 V / HP(1)/ Always-on(2)	3.3 V / LP	3.3 V	Mask_reset(3), overcurrent protection (4)
V_{DDCORE}	BUCK1	1.2 V	1.2 V/ HP / Always-on	0.9 V / LP	Off	Overcurrent protection
V_{DD_USB}	LDO4	3.3 V	3.3V/ Consumer driven(5)	3.3 V / Consumer driven	Off	-
V_{TT_DDR}	LDO3	OFF	0.675 V/ Always-on	Off	Off	Overcurrent protection
V_{DD_DDR}	BUCK2	OFF	1.35 V / HP / Always-on	1.35 V / LP	Off	Overcurrent protection
V_{REF_DDR}	DDR_REF	OFF	0.675 V / Always-on	0.675 V	Off	-
V_{DD_SD}	LDO5	2.9 V	2.9 V/ Consumer driven	2.9 V/ Consumer driven	Off / boot on(6)	-
V_{3V3}	BUCK4	3.3 V	3.3 V/ Consumer driven	3.3 V/ Consumer driven	Off	-
V_{BUS_DR}	PWR_USB_SW	OFF	ON	Off (suspend not supported on MP1 side)	Off	Pure software. No alternate registers
V_{DD_AUDIO}	LDO1	OFF	1,8 V / Consumer driven	1,8V / Consumer driven	Off	-
V_{DD_LCD}	LDO6	OFF	1,8 V / Consumer Driven	1,8 V / Consumer driven	Off	-

1. *HP/LP mode of the STPMIC1A regulator. Refer to [2]*
2. *Always-on: Keeps the core voltage on even if there is no software consumer.*
3. *Mask_reset: Specify the STPMIC1A mask_reset option to this regulator not to be impacted by a reset power cycle. (see Section 5.1.2 STPMIC1APQR restart_request and mask_reset options)*
4. *Overcurrrent protection: Specify the STPMIC1A option OCPOFF on this regulator. Overcurrent detection leads to a STPMIC1A 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 STPMIC1A alternate mode) in order to turn on immediately on wake-up (switch back to main mode)*

6 USB port management

The STM32MP15xD and STM32MP15xF devices have an enhanced consumer mission profile (see [8]). This profile allows the Arm® dual Cortex®-A7 CPUs to run at a higher clock frequency (see [7] for details and limitations).

Accordingly, the V_{DDCORE} supply voltage must be increased when the CPU frequency (Fmpuss_ck) operates above 650 MHz. When it does not operate in run mode above 650 MHz, the V_{DDCORE} supply voltage must be set back to its nominal voltage (1.2 V typ.).

- The V_{DDCORE} voltage increase is done by setting the BUCK1 of the STPMIC1A at the desired voltage value. It requires to set in advance the xxxx_MAIN_CR (HP), xxxx_ALT_CR (LP) power mode registers to the desired voltages.

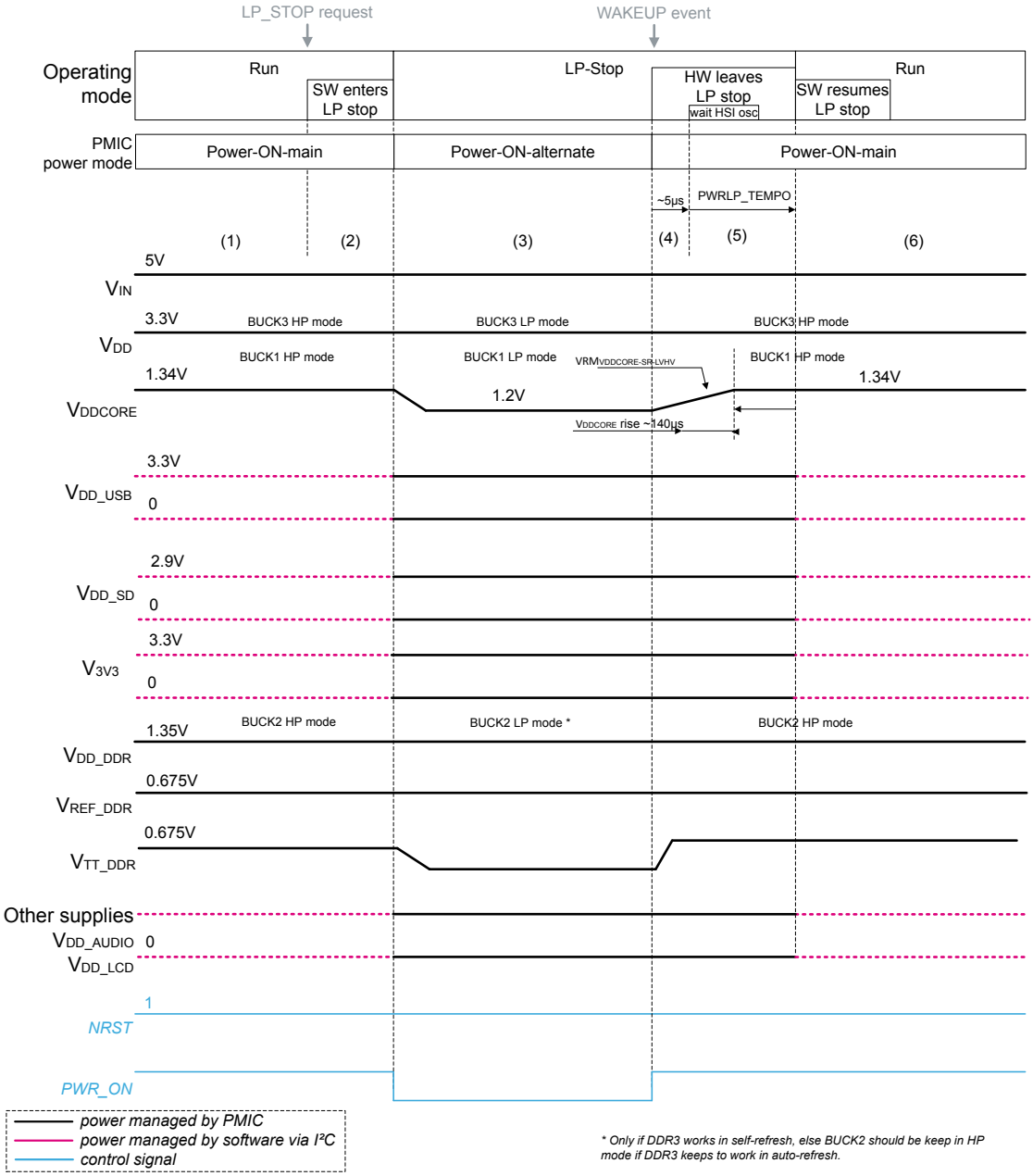
When programming from run to run above 650Mhz, V_{DDCORE} increases.

When going from run above 650 Mhz back to run mode, the frequency decreases before the switch.

The voltage switch between (HP) and (LP) is activated by the PWR_ON signal.

Figure 8 shows the same application as in Figure 5, except for the V_{DDCORE} supply power source.

Figure 8. Example with IOs at 3.3 V, DDR3L and V_{DDCORE} voltage scaling



Revision history

Table 5. **Document revision history**

Date	Version	Changes
09-Aug-2022	1	Initial version
01-Nov-2022	2	Updated: • Introduction • Table 3. Application operating modes • Section 5.1.1 Application turn-on / turn-off conditions • Section 5.2.1 Power-up by main supply plugin • Section 5.3.1 LP-stop mode • Section 5.2.3 Power-down by mains supply removal

Contents

List of figures

List of tables

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

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

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

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

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

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

© 2022 STMicroelectronics – All rights reserved