Infotainment Systems Product Development

P06 - Power Management Specification

SYNC Generation 4

Version 4.22 WIP

Version Date: August 5, 2020

UNCONTROLLED COPY IF PRINTED

FORD CONFIDENTIAL

Revision History

Date	Version	Created/Modified By	Notes
11/29/2017	4.00 WIP	Stephan Schueerholz / SSchueer	Initial Work in Progress for SYNC Gen4: • Cleanup of history. • Cleanup of not supported functionality/features (TBD). • Removed HS4 references (TBD). • Removed 1394 references (TBD).
4/25/2018	4.01 WIP	Amit Mandhane / Amandhan	Cleanup and modify CAN signals as per new database.
6/13/2018	4.02 WIP	Amit Mandhane / Amandhan	Add CAN signals as per DB_SyncGen4_4_02_Upd4_ICAN.dbc
7/13/2018	4.03 WIP	Amit Mandhane / Amandhan	Update CAN message name for Battery management signals in 2.1.4.1. Remove reference to 10-minute clock mode.
7/31/2018	4.04 WIP	Chris Tumas / ctumas	Add the VMCU Reset/Reboot power mode. CCPU_RESET no longer supported after DV 1. Added statement for support of code without Gyro/Accel. Redefined the CCPU_SENSOR_OE pin. Redefined the PMICX_PWR_GOOD pins to analog.
8/9/2018	4.05 WIP	Amit Mandhane / Amandhan	Add predictiove trigger signal to 2.1.4.1 and 2.1.5
8/20/2018	4.06 WIP	Amit Mandhane / Amandhan	Update extended play timer value to 1 min when LifeCycMde is factory in section 2.4.4
8/22/2018	4.07 WIP	Stephan Schueerholz / SSchueer	Added USB Ground Protection requirements (2.4.1.19)
8/23/2018	4.08 WIP	Stephan Schueerholz / SSchueer	Improvements to USB Ground Protection requirements (2.4.1.19)
8/27/2018	4.09 WIP	Amit Mandhane / Amandhan	Added Key OFF Power Moding requirements in 2.1.4.1, 2.1.4.2, 2.1.5, 2.4.1.5, 2.4.1.6, 2.4.2.3, 2.4.2.9, 2.4.2.12, 2.4.2.15, 2.4.3.1
9/18/2018	4.10 WIP	Amit Mandhane / Amandhan	Added "approach detection predictive trigger" requirements in 2.1.5, 2.4.2.2, 2.4.2.8, 2.4.2.11, 2.4.2.14
9/26/2018	4.11 WIP	Amit Mandhane / Amandhan	Added power transition restrictions based on OTA related CAN signal VehOnSrc_D_Stat in 2.1.4.1, 2.1.5, 2.4.2.1, 2.4.2.2, 2.4.2.7, 2.4.2.10, 2.4.2.11, 2.4.2.13, 2.4.2.14
11/7/2018	4.12 WIP	Chris Tumas / ctumas	Changed VMCU_E_SHUTDOWN to VMCU_E_SHTDWN in multiple locations. Changed section 2.4.1.2 to indicate a Low or High voltage condition. Also changed to remove the VCC_16V_OUT shutdown for this condition. Section 2.4.1.1 changed the delay to 310msec in step 6 and 530msec in step 36. Section 2.4.1.4 delay time changed to 510msec.
02/26/2019	4.13 WIP	Amit Mandhane / Amandhan	Add 3-Minute timer requirements related to Touchscreen HMI On Button present configuration to 2.4.2.5 and 2.4.2.8
03/28/2019	4.14 WIP	Amit Mandhane / Amandhan	Updated T02, T11 and T14 in section 2.4.2 – No transition to DisplayOnly when KeyOut is active and Transport is active. No change in T05 for this requirement to allow it to transition from Info to DisplayOnly and exit decision is made in DisplayOnly. Added <<TransportActive to Inactive>> event in 2.1.5 Updated T08 to exit DisplayOnly when <<TransportActive to Inactive>> event occurs.
06/28/2019	4.15 WIP	Amit Mandhane / Amandhan	Added GWM_Send_Signals_18_HS3 CAN message and modified EFP2_Button_Press CAN message in 2.1.4.1.

08/28/2019	4.16 WIP	Amit Mandhane / Amandhan	Updated «LoadShed Event» condition in 2.1.5. Added power transition restrictions based CAN signal KeyOffMde_D_Actl in 2.4.2.1, 2.4.2.2, 2.4.2.7, 2.4.2.10, 2.4.2.11, 2.4.2.13, 2.4.2.14. Added Scenario H for KeyOffMde_D_Actl missing in 2.4.3.1.
01/14/2020	4.17 WIP	Amit Mandhane / Amandhan	Updated «HMI On/Off» transition event for DE0X Touchscreen HMI On Button present configured on.
02/20/2020	4.18 WIP	Amit Mandhane / Amandhan	Add new predictive trigger requirements for rear doors, Lift gate, Tail gate, Vehicle remote start, Charging cord unplug in 2.1.4, 2.1.5, 2.4.2.2, 2.4.2.8, 2.4.2.11, 2.4.2.14
03/05/2020	4.19 WIP	Amit Mandhane / Amandhan	VMCU keeps DSPL_EN low during PowerOn, Wakeup or CCPU reset. VMCU follows logic from CCPU to control DSPL_EN
03/12/2020	4.20 WIP	Chris Tumas / ctumas	Change the definition of the VMCU_LVI signal to CCPU wake up from stop mode. Add CCPU STOP/Sleep mode state and CCPU START/Resume mode state.
		Isidro Garcia / igarci20	Updated DisplayOnly to Wait-Shutdown & VHM to Wait-Shutdown transitions to include STOP state requirements.
			Added R061 for CCPU reboot timer requirements.
			Updated R057 to add the CCPU reboot timer.
		Chris Tumas / ctumas	Changed the definition of the VMCU_IPC_SPARE tp GNSS_TIME_PPS to support Trimble DR on the CCPU.
		Isidro Garica / igarci20	Added CCPU watchdog timeout to T08 and T12 tranistions for completeness
			Updated T08 and T12 transitions to reflect CCPU requesting SUSPEND to enter stop mode
			Updated section references
			Added CCPU Sleep Timer
			Added Stop Mode Timer
03/24/2020	4.21 WIP	Isidro Garcia / igarci20	Changed 10-Minute Clock Timer to DisplayOnly Timer.
			Removed CCPU Sleep Timer
			Modified CCPU Stop/Sleep sequence to have VMCU follow CCPU for the DSPL_EN pin.
			Removed requirement to wakeup and suspend CCPU after the Stop Mode Timer expires.
04/27/2020	4.22 WIP	Isidro Garcia / igarci20	Changed Reverse Gear Active/Inactive conditions in Table 2.1.5

Table of Contents

1 Introduction

1.1 Purpose

This specification defines the overall power management scheme for the SYNC 4 device. This document is intended to capture the following components related to overall power management on the SYNC 4:

- Hardware-level power features
- Power states
- Power state management and communication between the CCPU and VMCU

1.2 Requirements Traceability

For the purpose of this document, a requirement is any a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user. Also, all stated requirements shall have a validation procedure (i.e. a test or tests) to verify that they have been met.

This document shall be considered the requirements "master". While other requirements systems may be used to facilitate traceability (such as Reqtify, DOORS, or RequisitePro), this document remains the final authority on approved requirements.

- Requirements shall be enclosed within tags that are formatted as follows:
  - [Xxx.R001.01] Some Requirement text [END] – this allows the requirements traceability tool to extract requirements from the document and test cases created to ensure test coverage.
  - The first three letters of the starting tag represent the specification file code. This code is represented as the first three letters of the file name.
  - Examples of starting tags include:
    - [S01.R020.02]
    - [A03.R132.01]
    - [P12.R025.12]
- [Xxx.R001.01] is decoded such that the second number (in this case R001) is the requirement number. The requirement number may not be duplicated in a document, but the number is not required to be contiguous. It shall be a zero-padded three digit number in all cases (starting with 001).
- [Xxx.R001.01] is decoded such that the final section specifies the version of the requirement. Prior to official sign-off and delivery, the version number shall always be 01 and incremented for any change to the text of the requirement thereafter. It shall be a zero-padded two-digit number in all cases (starting with 01).

1.2.1 Changes to requirements:

1.2.1.1 Additions

Requirements that are added to the document shall be given a unique number and the document version shall be incremented (after initial release).

1.2.1.2 References

Requirements that are deleted shall be removed from the document and the document version shall be incremented (after initial release).

1.2.1.3 Revisions

Requirements that are revised shall indicate the revision by changed text and an incremented revision number in the requirements start tag (after initial release). An example: [S01.R020.02] becomes [S01.R020.03].

High-level overviews or further explanations will not generally be enclosed in requirements tags. While these sections are not considered specifications directly (because they are not enclosed in tags), they shall be used to clarify the scope and intent of stated requirements.

1.3 References

This section contains references to documents which affect the requirements presented in this requirement specification.

Reference Title	Document Location
S23 IPC Remote Services	
S23b IPC Remote Services PM-NM	
Auxiliary Protocol Module (APIM) Infotainment Subsystem Part Specific Specification (SPSS) v???	
Operations Mode Management – Ford North America CGEA v1.3 / FNV2 V???	
A22G – "Welcoming/Program Screenflows"	

1.4 Terminology and Abbreviations

Term	Description
ACK	Acknowledgement of a particular event, typically through a defined communication protocol
Cold Boot	Device restart where the operational state is loaded entirely from Flash (all RAM contents cleared)
Infotainment System	Set of devices/ECUs in the vehicle that provide entertainment and HMI functionality
ECU	Electronic Control Unit – a node on one of the vehicle CAN busses
VP	Vehicle Prototype
BSP	Board Support Package – the software deliverable that includes all device drivers and system-level software (OEM adaptation layer)
CAN	Controller Area Network – a data-link-layer protocol used by all ECUs
LIN	Local Interconnect Network – a data-link-layer protocol
HMI	Human Machine Interaction
SPSS	
Cranking	
VBattState	Current Voltage State (NORM_VOLT, OVER_VOLT, UNDER_VOLT, etc.) of the system as defined by the Voltage Range Monitor.doc
PaaK	Phone as a Key – Feature to start the car with a phone instead of a key FOB present
LBI	Lincoln Backup Ignition – Feature to start the car when a PaaK device is unavailable or empty.

1.5 Schedule and Dependencies

The Power Management Specification is a priority component to SYNC, as all features are required to be implemented for the VP deliverable. Additionally, several key components (such as VMCU requirements and software design, as well as BSP software) are dependent upon the Power Management Specification.

2 Feature Design Requirements

2.1 Architecture

2.1.1 Hardware Power Features

The power management scheme implemented by the SYNC defines several hardware-level features that enable power control and communication. The hardware design strategy associates every power enable with feedback to indicate the power is regulated and stable.

The VMCU subsystem consists of the VMCU, the GPS, the CAN, the LIN, IPC interface and the Gyro/Accelerometer. All of these devices are powered from the always on VCC_3P3VBAT power. The VMCU low power state is required to put all of these interfaces in the standby or sleep mode. The VMCU must disable the VCC_5V_EN, the GNSS_ANT_PWR_EN and the MAIN_PWR_EN. The VMCU will then go into the LPU/Standby mode, with the 32Khtz clock supplied to the GPS, and the 256Kbytes of RAM being powered.

There will be SYNC 4 design variants without the Gyro/Acceleromter. VMCU code must handle these variants.

2.1.1.1 Power Supply Control

[P06.R029.03]The following overall power supply control signals are defined:

Name	I/O	Type	Description
VMCU_BAT_ADC	Input	Analog	Signal to monitor incoming battery level
VMCU_ADC_EN	Output	Active high	Signal to enable the ADC measurement. This signal must be enabled to measure battery voltage.
MAIN_PWR_EN	Output	Active high	Signal to enable the main buck/boost power supply to generate the 16V for the MCM, the Display, and the DRVC (9V).
VCC_16V_PG	Input	Active high	Signal active when the buck/boost regulator is stable.
VCC_5V_USB_EN	Output	Active high	Signal to enable the 5V USB voltage rail.
VCC_5V_USB_PG		Active high	Signal active when the 5V USB voltage rail is stable and regulated.
VCC_5V_SW_EN	Output	Active high	Signal to enable the 5V voltage rail. This needs to be enabled for CAN communication.
VCC_5V_SW_PG	Input	Active high	Signal active when the 5V voltage rail is stable and regulated.
GNSS_ANT_PWR_EN	Output	Active high	Signal to enable GNSS Antenna power.
GNSS_ANT_NFLT	Input	Active high	Signal indicates no Fault. H = No Antenna Fault: L= Antenna Fault.
GNSS_ANT_FFLAG	Input	Active high	Signal indicates Fault H = Fault Current over 250ma. L= No Fault
GNSS_ANT_SENSE	Input	Analog	Analog measurement of current used to detect Antenna.
VCC_8V_EN	Output	Active high	Signal to enable the 8V voltage rail.
VCC_8V_DIAG	Input	Analog	Voltage monitor on the +8V power rail. Divide by 6 for the voltage
VCC_9V_EN	Output	Active high	Signal to enable the 9V voltage rail.
VCC_9V_PG	Input	Active high	Signal active when the 9V rail is regulated and stable.
CCPU_PMIC_EN_N	Output	Active Low	Signal to enable the PMIC on CCPU board. L = Enable PMIC H = Disable PMIC
PMIC1_PWR_GOOD	Input	Analog	Signal active when the PMIC voltage is stable >1V = Power good <1V = Power RegulatorFault

PMIC2_PWR_GOOD	Input	Analog	Signal active when the PMIC voltage is stable >1V = Power good <1V = Power RegulatorFault
MCM_EN	Output	Active high	Signal to enable the MCM Power.
MCM_PWR_DIAG	Input	Analog	Voltage Monitor on the MCM power line. Divide by 8 for the voltage. Note: Check signal level of input prior to turning on MCM power if high then MCM short to battery.
MCM_SENSE	Input	Analog	MCM module current sense.
DSPL_EN	Output	Active high	Signal to enable the Display Power.
DSPL_PWR_DIAG	Input	Analog	Voltage Monitor on the Display power line. Divide by 8 for the voltage. Note: Check signal level of input prior to turning on Display power if high then Display short to battery.
DSPL_SENSE	Input	Analog	DSPL module current sense.
VMCU_2MHZ_OUT		PWM	Future 2MHz PWM signal – 50% Duty Cycle for power supplies. This signal is currently not used.
VMCU_400KHZ_OUT		PWM	Future Switching Frequency for BB SMPS set to 400KHz. Only operates when "Main Power enable" is active. This signal is currently not used.
USB_GND_DET	Input	Active High	L= No short to Battery H=Short to Battery This measurement is only valid if USB_GND_EN_N is high.
USB_GND_SNS_OUT	Input	Analog	Analog measurement of current through the USB GND path. The intent is to use the CMP 2 input with a threshold set to detect a battery short to GND.
USB_GND_EN_N	Output		L = USB GND connected H = USB GND disconnected Note: USB_5V_EN needs to be high to enable the pull up for the GND.

[END]

2.1.1.2 VMCU/CCPU Interface

[P06.R030.05]These signals enable direct communication/control between the VMCU and CCPU:

Name	Type	Description	Direction
UART_IPC_RXD/ UART_IPC_TXD	UART	Async IPC serial comm. Channel between VMCU and CCPU	Bidirectional
UART_IPC_CCPU_CTS	Digital	Future UART Handshake signal. This is a GPIO on VMCU.	VMCU->CCPU
UART_IPC_CCPU_RTS	Digital	Future UART Handshake signal. This is a GPIO on VMCU. The DV revision of the PCB moves this signal to pin that allows interrupt.	CCPU->VMCU
UART_IPC_SPARE_RXD/ UART_IPC_SPARE_TXD	UART	Async IPC spare serial comm. Channel between VMCU and CCPU	Bidirectional
CCPU_SCU_CONFIG2	Digital	Used to set the CCPU boot option. Must be set prior to booting the CCPU. USB: CCPU_SCU_CONFIG2 = L eMMC: CCPU_SCU_CONFIG2 = H	VMCU->CCPU
CCPU_SCU_CONFIG3	Digital	Used to set the CCPU boot option. Must be set prior to booting the CCPU.	VMCU->CCPU

		USB: CCPU_SCU_CONFIG2 = H eMMC: CCPU_SCU_CONFIG2 = L	
CCPU_RESET	Digital	High=Reset the CCPU : Low = Allow the CCPU out of reset. This would be the first option to try an recover a CCPU. The PMIC's can also hold the CCPU in reset. **This signal is not supported beyond DV 1 hardware.**	VMCU->CCPU
CCPU_SW_UPDATE	Digital	H = Allows the VMCU to direct the CCPU to do a software update. L = Default state no update.	VMCU->CCPU
CCPU_RECOVERY_N	Digital	L = Software update thru recovery IFS H = Normal Operation	VMCU->CCPU
DISPLAY_EN_N	Digital	L = Turn on the display power H = Turn off display power	CCPU->VMCU
BOOT_DTC_1	Digital	00 = Bad Boot loader 01 = Bad IFS 10 = Bad IFS and Recovery 11 = Normal Boot	CCPU->VMCU
BOOT_DTC_2	Digital	See above for DTC decode.	CCPU->VMCU
VMCU_LVI	Digital	GPIO to indicate Stop mode state Low – Normal High – CCPU can transition to STOP mode High to Low – CCPU wake up (when in STOP mode)	VMCU->CCPU
REVERSE_MODE	Digital	GPIO to indicate Reverse Gear is detected on CAN: 0 – Vehicle Not In Reverse 1 – Vehicle In Reverse	VMCU->CCPU
DOOR_OPEN_IND	Digital	GPIO to indicate Driver/Passenger Door open detected on CAN: 0 – Door Not Open 1 – Door Open	VMCU->CCPU
START_UP_ANIMATION	Digital	GPIO to indicate that the startup animation is to be played 0 – Key out of ignition 1 – Key in ignition	VMCU->CCPU
VMCU_E_SHTDWN	Digital	GPIO to indicate an emergency shutdown to the CCPU: 0 – Normal operation 1 – Emengency Shutdown	VMCU->CCPU
IPC_SPARE	Digital	GPIO to indicate a GNSS_TIME_PPS from the VMCU. This is used by the CCPU for the Trimble DR running on the CCPU. VMCU must recreate this from the GNSS signal.	VMCU->CCPU

[END]

2.1.1.3 CCPU Power and Reset Control signals

[P06.R031.04]These signals enable the CCPU to control specific power supplies and peripheral power rails:

Name	Type	I/O	Description
ENET0_WAKE	GPIO	O	H = ENET WAKE enabled L = ENET WAKE disabled
ENET0_RST_N	GPIO	O	H = ENET PHY out of reset L = ENET PHY in reset
ENET0_EN	GPIO	O	H = ENET PHY enabled

			L = ENET PHY disabled
DISPLAY_PDB_N	GPIO	O	L = FPDLINKIII Phy in power down H = FPDLINKII Phy enabled
DISPLAY_L_PDB_N	GPIO	O	L = FPDLINKIII Phy in power down H = FPDLINKII Phy enabled
DRVC_OSC_OE	GPIO	O	H = 25Mhz OSC enabled L = 25Mhz OSC disabled
DRVC_PDB	GPIO	O	L = FPDLINKIII Phy in power down H = FPDLINKII Phy enabled
IPOD_RESET	GPIO	O	H = iPOD out of reset L = iPOD in reset (Note: This IC is always going to sleep mode.)
WIFI_BT_PD_N	GPIO	O	L = WIFI/BT module in powerdown state H = WIFI/BT module powered
USB_PWR_EN_OTG2	GPIO	O	H = GNSS USB port enabled L = GNSS USB port disabled
CODEC_RST_N	GPIO	O	H = Audio Codec out of reset L = Audio Codec in reset
ARVC_PD_N	GPIO	O	L = Video Codec in power down H = Video Codec Powered
ARVC_RST_N	GPIO	O	L = Video Codec in reset H = Video Codec out of reset
PMIC_I2C_SCL	I2C		PMIC I2C bus to control iMX8 power supplies
PMIC_I2C_SDA	I2C		
USB_SWITCH_CTRL1	GPIO	O	L = No power on the USB connector H = +5V on the USB connector

[END]

2.1.1.4 CAN Interface

[P06.R032.02]NOTE: The 3 CAN interface signals are duplicated for each CAN interface: HS1-CAN, HS3-CAN and HS4-CAN. The HSxCAN_RXD pin on each interface is used for the CAN wakeup. The CAN transceiver only requires the VIO voltage to support the CAN wakeup event. The VCC_5V_SW power will be disabled in the LPU/Standby State. The HSxCANRXD signal needs to be defined as a Wakeup pin in order to support LP/Standby wake up.

Name	Type	Description
HSxCAN_RXD / HSxCAN_TXD	FlexCAN	CAN serial communication channel.
HSxCAN_STB	Active High	CAN standby H = Standby: L= Normal Operation

[END]

2.1.1.5 LIN Interface

[P06.R032.04]. The LIN interface wakeup is the LIN_RX pin. The LIN_RX signal needs to be defined as a Wakeup pin in order to support LP/Standby wakeup.

Name	Type	Description
LIN_RX / LIN_TX	Lin	LIN serial communication channel. The DV revision of the PCB moves the LIN_RX signal to pin that allows wakeup.
LIN _SLP_N	Active Low	LIN bus sleep enable L = Sleep: H = Normal operation

[END]

2.1.1.6 GNSS Interface

The VMCU interfaces to the GNSS IC over the SPI bus. The GNSS is powered from the same voltage as the VMCU. It requires a command to put it in low power standby mode. The GNSS will then use the SPI_CS going low to wakeup the GNSS IC.

Name	Type	I/O	Description
GNSS_RESET_N	GPIO	O	H = GPS out of Reset : L= GPS in reset. Must be set for 10msec to be recognized. (Should not use the reset in normal operation.)
GNSS_SAFE_BOOT_N	GPIO	O	H = Boot Normal : L = Boot into Safe mode. Note: Output must be set high prior to enabling as an output. We do not want this pin to momentarily go low. Note: SAFEBOOT is only sampled during a GNSS_RESET_N low.
GNSS_TIME_PPS	GPIO	I	GNSS Pulse Per Second
CLKOUT1_32K_GPS	CLK	O	32Khz Clock Always enabled even during KOL
GNSS_SPI0_SCLK	SPI	O	GNSS clock
GNSS_SPI0_CS0	SPI	O	GNSS chipselect active low. (This is will wake the GNSS IC)
GNSS_SPI0_MOSI	SPI	O	GNSS Data Out
GNSS_SPI0_MISO	SPI	I	GNSS Data In

2.1.1.7 Gyro/Accelerometer sensor

The VMCU interfaces to a 6-Axis Motion Tracking device over I2C. The Gyro/Accelerometer sensor is powered from the same voltage as the VMCU. It requires a command to put it into sleep mode. The hardware architecture supports an interrupt to allow a wakeup event from the sensor.

Name	Type	I/O	Description
SENSOR_INT_N	GPIO	I	Sensor Interrupt (3D Accel and 3D Gyro) L = INT H = No INT
SENSOR_FSYNC	GPIO		Not yet used or defined
CCPU_SENSOR_OE	GPIO	O	Enable I2C connection to CCPU. H = VMCU can access the CCPU I2C slave L = CCPU I2C is isolated from the VMCU Should only be enabled (high) after CCPU section is powered and for CCPU access.
SENSOR_I2C_SDA	I2C		I2C sensor Data (3D Accel and 3D Gyro)
SENSOR_I2C_SCL	I2C		I2C sensor Clock (3D Accel and 3D Gyro)

2.1.1.8 Debug Interface and Diagnostics

The following signals are used for development, debug and diagnostics:

Name	Type	I/O	
UART_DEBUG_VMCU_TX UART_DEBUG_VMCU_RX	UART		Dedicated UART debug port. These pins route to the debug connector.
MIC_SENSE1_P	Analog	I	Microphone 1 (+) Diagnostic Input
MIC_SENSE1_N	Analog	I	Microphone 1 (-) Diagnostic Input
MIC_SENSE2_P	Analog	I	Microphone 2 (+) Diagnostic Input (DNP)
MIC_SENSE2_N	Analog	I	Microphone 2 (-) Diagnostic Input (DNP)
VMCU_SW_UPDATE_N	GPIO	I	H = Normal (No update): L = Force the VMCU to update code over the UART. There are jumpers on the debug board to support pulling this signal low.
CCPU_FORCE_ON_N	GPIO	I	H = Normal : L = Force the CCPU on and ignore any IPC or Watchdog events. There are jumpers on the debug board to support pulling this signal low.
FORCE_CCPU_USB_BOOT_N	GPIO	I	Low = Debug board jumper installed to instruct VMCU to put CCPU in USB_BOOT High = Normal
IPC_ISO_CONTROL	GPIO	I	H = IPC UART interface enabled to the CCPU:

			L = IPC UART is routed thru the debug board. The debug PCB has a jumper to allow the IPC UART to route thru the PC. This signal is only valid if the PMIC is powered.
VMCU_LED	GPIO	O	L = LED off : H = LED on

2.1.2 CCPU Power Modes

The Power Management Specification defines the CCPU Power Modes, which are power modes implemented by the SYNC BSP. The CCPU Power Manager acts as a slave to the VMCU; that is, all power mode transitions are commanded by the VMCU (by sending a SET_PM_STATE message to the CCPU, defined by "S23b IPC Remote Services PM-NM"). The CCPU may request certain power mode transitions (by sending a SET_PM_STATE to the VMCU), but the VMCU is not required to command the requested state transition.

It is important to note that while CCPU Power Modes enable the SYNC to meet the SPSS power mode requirements, they do not directly relate to individual SPSS-defined power mode states (Sleep, Standby, and Functional) or SPSS-defined System State (OFF, Display only, MMActive, MMInactive, Extended Play, and Phone). Rather, the SPSS power requirements are implemented via a combination of VMCU actions, CCPU power modes, and SYNC application-specific behavior.

The following CCPU Power Modes are defined:

CCPU Power Mode	Description	Applicable SPSS Power Mode(s)	Application Functions
Unpowered	No power applied to system	Unpowered	N/A
Wait-On	Minimal CPU peripherals are powered, CPU is running (transitional state only)	N/A	N/A
Infotainment	SYNC is fully operational, all CPU peripherals powered, Infotainment components are ON	MMActive Extended Play Phone	eCall (911)
DisplayOnly	SYNC is partially operational, display/HMI available, limited CPU peripherals powered, Infotainment components are OFF	10-Minute Clock MMInactive	"Welcome" Screen Transport Mode CD Load/Eject with ignition off Load Shed notification VIN Locking
VHM	SYNC is partially operational, display/HMI unavailable, limited CPU peripherals powered, Infotainment components are OFF	MMInactive	Background tasks
Wait-Shutdown	CCPU is running, but transitioning to Unpowered mode – some peripherals may begin to power down. E-Shutdown process is started	N/A	N/A

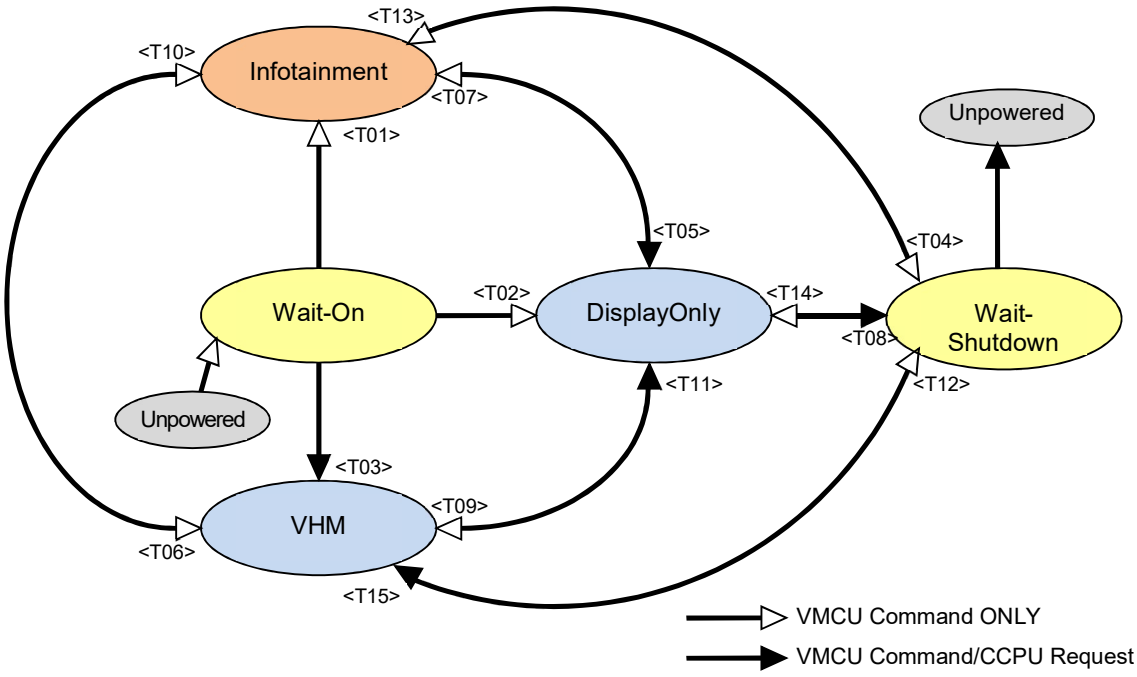

Figure 2-1. Power Modes

[P06.R033.08]The following describes the power state of each CCPU-controlled peripheral in each of the power modes:

	Infotainment System (HMIAudioMode)	Display (VMCU & CCPU)	CCPU Power	Audio CODEC	Video CODEC	Audio/Video Amps	Bluetooth/WiFi	Ethernet AOBR	USB	MCM
Unpowered	-	-	-	-	-	-	-	-	-	-
Wait-On	-	-	X	X	X	X	X	X	X	X
Infotainment	X	X	X	X	X	X	X	X	X	X
DisplayOnly	-	X	X	X	X	X	X	X	X	X
VHM	-	-	X	X	X	X	X	X	X	-
Wait-Shutdown	-	X	X	X	X	X	X	X	Z	X

- \- Peripheral/feature unpowered
- L Peripheral/feature in low power state
- X Peripheral/feature fully powered
- Z Peripheral/feature in transition / indeterminate

Table 2.1 - Peripheral Power States

[END]

[P06.R058.08]
The following describes the activity state of each CAN bus in each of the power modes:

	I-CAN (HS3)	LIN
Unpowered	X	X
Wait-On	R	S
Infotainment	A	S
DisplayOnly	H	S
VHM	L	S
Wait-Shutdown	H	S

Table 2.2 – CAN Bus Power States

X - VMCU leaves bus inactive (i.e., does not request activity and should vote to go to sleep if bus is active) and does not allow CCPU activity via signal layer or diagnostics

A - VMCU keeps bus active (i.e., maintains bus activity and does not vote to go to sleep) and allows CCPU activity via signal layer or diagnostics

R - VMCU leaves bus inactive (i.e., does not request activity and should vote to go to sleep if bus is active) unless CCPU requests activity via signal layer or diagnostics (VMCU will wake bus and allow activity)

H - VMCU leaves I-CAN bus in previous state prior to power mode transition (e.g., bus is kept awake if prior mode was Infotainment) until either the mode is exited or the CCPU sets HMIAudioMode==OFF or HMIAudioMode==LoadShed. Once bus state is inactive, the VMCU does not allow CCPU activity via signal layer or diagnostics unless CCPU requests activity via ActiveLoad request in SET_NM_STATUS

L - VMCU leaves bus inactive (i.e., does not request activity and should vote to go to sleep if bus is active) and does not allow CCPU activity via signal layer or diagnostics unless CCPU requests activity via ActiveLoad request in SET_NM_STATUS

S- If Illumination Inactive, VMCU sets the Powerstate of LIN to the same as I-CAN.
When VMCU puts I-CAN into the sleep state, it waits for the I-CAN to enter sleep state before putting LIN into sleep state.
If Illumination Keep Active, VMCU keeps the LIN active and provides illumination information to keep the ICP illuminated.
If ICP (EFP) Network is configured for CAN, LIN is kept in sleep state.

[END]

2.1.3 CCPU / VMCU Power Mode Interaction

[P06.R034.01]The Power Management Specification defines the interaction between the VMCU and the CCPU to implement the CCPU power modes. The VMCU and CCPU interact via the following two mechanisms:

- Direct I/O Line – A direct hardware interface (typically GPIO and/or IRQ line); refer to “Hardware Power Features”, above
- IPC Communication – Protocol-defined communication via the IPC communication interface between the VMCU and CCPU; refer to “S23b IPC Remote Services PM-NM”

The VMCU and CCPU have distinct roles in the overall power management strategy, as follows:

The VMCU is the power mode master: The VMCU acts as the overall power manager for the SYNC, by monitoring all external events which require a power mode change. The VMCU also responds to requests from the CCPU for changes in power state, but ultimately determines whether such power mode changes are valid based on state information (battery voltage, temperature, commands from other ECUs, etc.)

The CCPU is the power mode slave: The CCPU can only change power modes when requested by the VMCU. The CCPU must respond to (and acknowledge) any valid power mode request from the VMCU at any time, and must inform applications of the change of power mode.
[END]

2.1.4 External Inputs

The power management strategy of the SYNC depends on the following external inputs:

2.1.4.1 CAN Signals

The following CAN signals serve as inputs to the power manager. These signals may be read by both the VMCU and CCPU to determine or direct the power mode.

Message	Signal Name	Notes/Default Value
Locking_Systems_2_HS3	Veh_Lock_Status	0x1 LOCK_ALL
BodyInfo_3_HS3	DrStatDrv_B_Actl	0x0 Closed
	DrStatPsngr_B_Actl	0x0 Closed
	DrStatRl_B_Actl	0x0 Closed
	DrStatRr_B_Actl	0x0 Closed
	DrStatTgate_B_Actl	0x0 Closed
	DrStatInnrTgate_B_Actl	0x0 Closed
	LifeCycMde_D_Actl	0x0 Normal
	Delay_Accy	0x0 Off
	Ignition_Status	0x0 Unknown
	Backlit_LED_Status	0x0 Off
	Dimming_Lvl	0xC Night_12
	Remote_Start_Status	0x0 Null
EnginePwrData_HS3	Shed_Level_Req	0x0 No_Shed
	Batt_Crit_SoC_B	0x0 Inactive
	Batt_Lo_SoC_B	0x0 Inactive
	Shed_Drain_Eng_Off_B	0x0 Inactive
	Shed_T_Eng_Off_B	0x0 Inactive
ACU_Send_Signals_7	ACU_FunctionStatus_St	0x0 Invalid
VehEmergencyData1_HS3	eCallNotification	0x0 Not_Supported 0x1 Normal 0x2 Active 0x3 Not_Used
EngVehicleSpThrottle2_HS3	GearRvrse_D_Actl	0x7 Fault
VehicleOperatingModes_HS3	Eng_D_Stat	0x0 EngOff (default) 0x1 EngOn 0x2 EngAutoStopped 0x3 NotUsed
Body_Info_6_HS3	PrsnlDevChrgEnbl_B_Rq	0x0 Inactive 0x1 Active
	IgnPsswrdDsply_B_Rq	0x0 Inactive 0x1 Active
Body_Info_1_HS3	GearLvrPos_D_Actl	0xE Unknown
TCU_Send_Signals_5_HS3	EmgcyCall_D_Stat	0x0 Null
EFP_Button_Press	Coding_BtnID_A	0x0 Not Pressed
	Coding_BtnID_B	0x0 Not Pressed
	Coding_BtnID_C	0x0 Not Pressed
	Coding_BtnID_D	0x0 Not Pressed
	BtnID_A	0xFF

	BtnID_B	0xFF
	BtnID_C	0xFF
	BtnID_D	0xFF
EFP2_Button_Press	Coding_BtnID_A_29F	0x0 Not Pressed
	BtnID_A_29F	0xFF
RACM_Button_Press	RACM_Coding_BtnID_A	0x0 Not_Pressed
	RACM_Coding_BtnID_B	0x0 Not_Pressed
	RACM_Coding_BtnID_C	0x0 Not_Pressed
	RACM_Coding_BtnID_D	0x0 Not_Pressed
	RACM_BtnID_A	0xFF
	RACM_BtnID_B	0xFF
	RACM_BtnID_C	0xFF
	RACM_BtnID_D	0xFF
Body_Info_10_HS3	VehWlcmFrwlMde_D_Stat	0x0
MasterReset_HS3_ECG_Data	KeyOffPwMde_D_Stat	0x0 Inactive
Body_Info_7_HS3	PwLoApim_T_Actl	0x0 0 minutes – shutdown from Stop to Sleep mode 0x1 1 minutes 0x2 2 minutes … 0x7FF minutes (2047 minutes)
	KeyOffMde_D_Actl	0x0 Normal
GWM_Send_Signals_19_HS3	VehOnSrc_D_Stat	0x0 OFF
GWM_Send_Signals_18_HS3	GearPos_D_Trg	0xF Unknown
Cluster_HEV_Data5_HS3	PlgActvArb_B_Dsply	0x0 OFF

2.1.4.2 Wakeup Sources

[P06.R036.09]In order to transition properly from Suspend, the SYNC module must receive a wakeup (via network activity) prior to receiving certain events. The following sources will serve as a wakeup.

Wakeup Source	Description
Run/ACC	CAN Message BodyInfo_3_HS3, Signals Ignition_Status –OR- CAN Message BodyInfo_3_HS3, Signals Delay_Accy
HMI On/Off (EFP, ICP, GFM or RSEM Button)	CAN Message EFP_Button_Press –OR- CAN Message EFP2_Button_Press –OR- LIN message ICPBtnStateRotary –OR- If REFP == Available: CAN Message RACM_Button_Press
LIN wakeup from ICP	LIN Wakeup on LIN message ICPBtnStateRotary
CD Eject (EFP, ICP, GFM or RSEM Button)	CAN Message EFP_Button_Press –OR- CAN Message EFP2_Button_Press –OR- LIN Message ICPBtnStateRotary –OR- If REFP == Available: CAN Message RACM_Button_Press
CD Insert	CAN Message ACU_Send_Signals_7, Signal ACU_FunctionsStatus_St
Door Unlock	CAN Message Locking_Systems_2_HS3 Signals Veh_Lock_Status
Door Opened	CAN Message BodyInfo_3_HS3, <(DrStatDrv_B_Actl or DrStatPsngr_B_Actl or DrStatRl_B_Actl or DrStatRr_B_Actl or DrStatTgate_B_Actl or DrStatInnrTgate_B_Actl)Door == Ajar
Illumination On	CAN Message BodyInfo_3_HS3, Signal Dimming_Lvl –OR- CAN Message BodyInfo_3_HS3, Signal Backlit_LED_Status
Phone as a Key Charging	If PaaK==Enabled: Body_Info_6_HS3, PrsnlDevChrgEnbl_B_Rq == Active
LBI Password Entry Screen	If PaaK==Enabled: Body_Info_6_HS3, Signal IgnPsswrdDsply_B_Rq == Active

eCallNotification	ECALL == Enabled: VehEmergencyData1_HS3, CAN signal eCallNotification == Active
Key Off power Moding	CAN Message MasterReset_HS3_ECG_Data Signal KeyOffPwMde_D_Stat == On
Predictive trigger approach detection	CAN message Body_Info_10_HS3 signal VehWlcmFrwlMde_D_Stat == Approach
Vehicle remote start	CAN Message BodyInfo_3_HS3, Signal Remote_Start_Status == Remote
Charge cord Unplugged	CAN Message Cluster_HEV_Data5_HS3 signal PlgActvArb_B_Dsply changes from On to Off

[END]

2.1.4.3 LIN Signals

The following LIN signals serve as inputs to the power manager. These signals may be read by both the VMCU and CCPU to determine or direct the power mode.

Message	Signal Name	Notes/Default Value
ICPBtnStateRotary	ICPBtnCoding_A	0x0 Inactive
	ICPBtnID_A	0xFF ICPBtnID_Idle
	ICPBtnCoding_B	0x0 Inactive
	ICPBtnID_B	0xFF ICPBtnID_Idle
	ICPBtnCoding_C	0x0 Inactive
	ICPBtnID_C	0xFF ICPBtnID_Idle
	ICPBtnCoding_D	0x0 Inactive
	ICPBtnID_D	0xFF ICPBtnID_Idle

2.1.4.4 Button Coding State Normalization

[P06.R069.01] For all requirements refering to pressed/not pressed the normalization specified in the Button Strategy SPSS, chapter 1.4.3.14 BUTTON-SR-REQ-014709/C (TcSE ROIN-194455-1) shall be applied.
[END]

The mapping is shown here for information purposes only.

Button Encoding	Normalization
Inactive (0x00)	Not_Pressed
Active (0x01)	Pressed
ShortEvent (0x02)	Pressed
ShortElapsed (0x03)	Pressed
LongEvent (0x04)	Pressed
Stuck (0x05)	Not_Pressed
Undefined (0x06-0x0E)	Not_Pressed
Idle (0x0F)	Not_Pressed

2.1.5 Transition Conditions and Events

For each of the various power mode transitions, a commonly used set of “conditions” (a logical state the system is currently in) and “events” (a trigger that causes a transition) are defined as follows:

Condition	Definition
(C) «Run/ACC Active»	CAN signal [Ignition_Status = Accessory || Run] –OR- [Delay_Accy == ON]
(C) «ACC Active»	<Ignition_Status == Accessory> -OR- [<Ignition_Status == Off> -AND- <Delay_Accy == ON>]
(E) «ACC Event»	«Ignition_Status changes to Accessory» -OR- - [<Ignition_Status == Off> -AND- <Delay_Accy changes to ON>]
(C) «Run Active»	<Ignition_Status == Run>
(E) «Run/ACC Event»	CAN signal Ignition_Status CHANGES to one of Accessory or Run OR Delay_Accy to ON

(C) «KeyOut Active»	CAN signal Ignition_Status == OFF AND Delay_Accy == OFF
(E) «KeyOut Event»	(CAN signal Igntion_Status CHANGES to OFF AND Delay_Accy == OFF) OR (CAN signal Delay_Accy CHANGES to OFF AND Ignition_Status == OFF)
(C) «EngineOn Inactive»	CAN signal Eng_D_Stat == EngOff
(E) «Lock Event»	CAN signal Veh_Lock_Status changes to Locked
(E) «Door Unlock»	CAN signal Veh_Lock_Status changes to Unlock
(C) «PM Activity»	1. The CCPU has requested to remain powered (via the PM_ACTIVITY_TIMER message) within Activity Timer period AND the CCPU has request the CAN bus to remain powered (via the SET_NM_STATUS message with ActiveLoad == 1) within Activity Timer period. "PM_Activity requested" has **events** for entry (NM_STATUS with Active Load bit set) and **conditions** for remaining in VHM (PM_ACTIVITY message within the 10 second timeout). The Vmcu Timer is both a **condition** for entry (not having timed out), as well as an **event** for exiting VHM mode (timeout
(C) «No PM Activity»	1. The CCPU has NOT requested to remain powered (via the PM_ACTIVITY_TIMER message) within Activity Timer period, **and** 2. The CCPU has NOT requested to remain powered (via the SET_NM_STATUS message, with Active_Load==1) within Activity Timer period.
(E) «PM Activity Expires»	VMCU detects that CCPU Activity Timer (defined in Section 2.4.4) has expired before CCPU has requested activity (via PM_ACTIVITY_TIMER)
(C) «Phone Activity»	CCPU has indicated phone call active (via PM_ACTIVITY_PHONE) within Phone Timer period.
(C) «No Phone Activity»	CCPU has NOT indicated phone call active (via PM_ACTIVITY_PHONE) within Phone Timer period. No Phone call activity is currently being requested. For example, this condition is required for a transition out of Infotainment (see T04, 2/3 and 3/3).
(C) «No Priority Phone Activity»	CCPU has NOT indicated PRIORITY phone call active (via PM_ACTIVITY_PHONE with Flags indicating Priority call) within Phone Timer period. (A non-priority phone call may be active).
(E) «Phone Timer Expires»	VMCU detects that Phone Timer (defined in Section 2.4.4) has expired before CCPU has requested phone activity (via PM_ACTIVITY_PHONE)
(C) «VBattState Normal»	9.5V ≤ B+ ≤ 16V Here the condition is necessary, for example, for tranisitions from WaitOn as well as for transitions to Infotainment Power Mode. Also, it's negative is an event for a transition from Infotainment to DisplayOnly Power Mode, for example.
(C) «VBattState Faulted»	B+ < 9.5V –OR-B+ > 16V
(C) «LoadShed Active»	Load Shed is a condition for transitioning from Infotainment to DisplayOnly Power Mode. For the purposes of this specification, «LoadShed Active» refers to the set of CAN signal states defined by SPSS PWRMAN-GREQ-40679.
(E) «LoadShed Event»	LoadShed is an event for transitioning from Infotainment to DisplayOnly Power Mode. For the purposes of this specification, «LoadShed Event» refers to a change of signal states defined by SPSS PWRMAN-GREQ-40679. Or [<Ignition_Status == Off> AND KeyOffMde_D_Actl == Critical Battery]
(C) «Transport Active»	<LifeCycMde_D_Actl == Transport>
(C) «Transport Inactive»	<LifeCycMde_D_Actl != Transport>

(E) «Transport Active To Inactive»	<LifeCycMde_D_Actl == Transport> AND «KeyOut Active»
(E) «HMI On/Off»	Button press (per SPSS BUTTON-GFUN-39745) signaled by EFP via CAN message EFP_Button_Press with Button ID 0x1F (ON/OFF); detected as a change from Not_Pressed to Pressed Or Button press (per SPSS BUTTON-GFUN-39745) signaled by GFM via CAN message EFP2_Button_Press with Button ID 0x1F (ON/OFF); detected as a change from Not_Pressed (0x0) to Pressed (0x1) Or If *REFP* == *Available*: Button press (per SPSS BUTTON-GFUN-39745) signaled by RSEM via CAN message RACM_Button_Press with Button ID 0x1F (ON/OFF); detected as a change from Not_Pressed to Pressed For LIN connected ICP: Button press signaled by ICP via LIN message ICPBtnStateRotary ID 0x2 ICPBtnID_? (ICPBtnID_OnOff = 31); detected as a change from Not_Pressed to Pressed Or If DE0X Touchscreen HMI On Button present configured: CCPU request IPC message 0x0D REQUEST_EXTENDED_PLAY_MODE HMI On/Off
(E) «CD Eject»	Button press (per SPSS BUTTON-GFUN-39745) signaled by EFP via CAN message EFP_Button_Press with Button ID 0x34 (Eject); detected as a change from Not_Pressed to Pressed Or Button press (per SPSS BUTTON-GFUN-39745) signaled by GFM via CAN message EFP2_Button_Press with Button ID 0x34 (Eject); detected as a change from Not_Pressed (0x0) to Pressed (0x1) Or If *REFP* == *Available*: Button press (per SPSS BUTTON-GFUN-39745) signaled by RSEM via CAN message RACM_Button_Press with Button ID 0x34 (Eject); detected as a change from Not_Pressed to Pressed For LIN connected ICP: Button press signaled by ICP via LIN message ICPBtnStateRotary ID 0x2 ICPBtnID_? (ICPBtnID_Eject = 52); detected as a change from Not_Pressed to Pressed
(E) «CD Insert»	CD Insertion notification signaled by AHU via CAN message ACU_Send_Signals_7 with signal ACU_Function_Status_St == 0x2 (Loading)
(C) «eCallNotification Active»	(If DE01.ECALL = Enabled CAN signal eCallNotification changes to 0x2 (Active))
(E) «Doors Opened»	<(DrStatDrv_B_Actl or DrStatPsngr_B_Actl or DrStatRl_B_Actl or DrStatRr_B_Actl or DrStatTgate_B_Actl or DrStatInnrTgate_B_Actl)Door == Ajar> after previous status <Closed> via CAN
(E) «Doors Closed»	CAN signals (DrStatDrv_B_Actl, DrStatPsngr_B_Actl) indicate one or both driver or passenger doors CHANGES to Closed (0x0) after previous state Ajar (0x1) AND both doors currently indicate Closed (0x0)
(C) «Reverse Gear Active»	If Transmission Type = Automatic AND Rear Camera == RVC Present AND ReverseGear = ReverseGearLegacy (Body_Info_1_HS3::GearLvrPos_D_Actl == Reverse (0x01) AND Body_Info_1_HS3::GearLvrPos_D_Actl_UB == 0x01) If Transmission Type = Automatic AND Rear Camera == RVC Present AND ReverseGear = ReverseGearNew (Body_Info_1_HS3::GearLvrPos_D_Actl == Reverse (0x01) AND Body_Info_1_HS3::GearLvrPos_D_Actl_UB == 0x01) AND (GWM_Send_Signals_18_HS3:: GearPos_D_Trg == Reverse (0xE) AND GWM_Send_Signals_18_HS3::GearPos_D_Trg_UB == 0x01)

	If Transmission Type = Manual AND Rear Camera == RVC Present EngVehicleSpThrottle2_HS3::GearRvrse_D_Actl == Active_confirmed (0x03)
(C) «Reverse Gear Inactive»	If Transmission Type = Automatic AND Rear Camera == RVC Present AND ReverseGear = ReverseGearLegacy (Body_Info_1_HS3::GearLvrPos_D_Actl != Reverse (0x01) AND Body_Info_1_HS3::GearLvrPos_D_Actl_UB == 0x01) If Transmission Type = Automatic AND Rear Camera == RVC Present AND ReverseGear = ReverseGearNew (Body_Info_1_HS3::GearLvrPos_D_Actl != Reverse (0x01) AND Body_Info_1_HS3::GearLvrPos_D_Actl_UB == 0x01) OR (GWM_Send_Signals_18_HS3:: GearPos_D_Trg != Reverse (0xE) AND GWM_Send_Signals_18_HS3::GearPos_D_Trg_UB == 0x01) If Transmission Type = Manual AND Rear Camera == RVC Present EngVehicleSpThrottle2_HS3::GearRvrse_D_Actl != Active_confirmed (0x03)
(E) «Illumination BatterySave Timer Expires»	VMCU detects that the Illumination BatterySave Timer (defined in Section **Error! Reference source not found.**) has expired.
(C) «Illumination Keep Active»	BatterySaveTimer > 0 –AND- (Illumination Signals Dimming_Lvl >OFF (0x00) –OR- Backlit_LED_Status >OFF (0x00) were last received before bus went to sleep)
(C) «Illumination Inactive»	BatterySaveTimer == 0 –OR- (Illumination Signals Dimming_Lvl –AND- Backlit_LED_Status are last received OFF (0x00) before bus went to sleep)
(E) «PaaK Charging Timer Expires»	If PaaK==Enabled: VMCU detects that the PaaK Charging Timer (defined in Section **Error! Reference source not found.**) has expired.
(C) «PaaK Charging Active»	If PaaK==Enabled: PrsnlDevChrgEnbl_B_Rq == Active
(C) «PaaK Charging Inactive»	If PaaK==Enabled: PrsnlDevChrgEnbl_B_Rq != Active If PaaK!=Enabled: always inactive
(C) «LBI Password Screen Active»	If PaaK==Enabled: PasswordEntryScreen_Rq == Active
(E) «LBI Interaction Watchdog Timer Expires»	VMCU detects that the LBI Interaction Watchdog Timer (defined in Section **Error! Reference source not found.**) has expired.
(E) «LBI Interaction Indication»	If PaaK==Enabled: PM_ACTIVITY_LBI is received
(E) «Predictive trigger active»	«VehWlcmFrwlMde_D_Stat == 0x1 Approach»
(E) «Predictive trigger inactive»	«VehWlcmFrwlMde_D_Stat != 0x1 Approach»
(C) «KeyOffPwMde Active»	«KeyOffPwMde_D_Stat == 0x1 ON»
(C) «KeyOffPwMde Inactive»	«KeyOffPwMde_D_Stat == 0x0 Inactive»
(C) «KeyOffMde Critical Battery Active»	«KeyOffMde_D_Actl == Critical Battery»
(C) «KeyOffMde Critical Battery Inactive»	«KeyOffMde_D_Actl != Critical Battery»
(C) «VehOnSrc_D_Stat OTA active»	«VehOnSrc_D_Stat == OverTheAir»
(C) «VehOnSrc_D_Stat OTA inactive»	«VehOnSrc_D_Stat != OverTheAir»
(C) «Remote start active»	«Remote_Start_Status == Remote»
(C) «Remote start inactive»	«Remote_Start_Status != Remote»
(E) «Charge cord unplugged»	«PlgActvArb_B_Dsply changes from ON to OFF»

2.2 Goals

The overall goal of the SYNC power manager is to meet the power mode requirements as defined by the SPSS. Additionally, this document will define the interaction between the various systems which implement the power scheme (hardware, BSP, VMCU, CCPU, and external ECUs).

The specific design goals of SYNC power manager are as follows:

- Reliability and graceful shutdown through all normal power transition scenarios.
- Prevent catastrophic failure during unanticipated power-loss or instant suspends.
- Provide applications with notification of the power state transitions (under normal transition conditions)
- Provide applications with sufficient time to shutdown (under normal transition conditions)
- Enforce any global CCPU power policy (in general global policy should be enforced from the VMCU as this is more fail-safe). An example of CCPU power policy is enforcing the Wait-Suspend limit on applications.
- Prevent corruption of NAND flash data by limiting scenarios where NAND writes are jeopardized by power loss

2.3 Non-Goals

The power manager is not designed to do the following:

- Enforce application specific policy (e.g., phone or media player behavior in specific states)
- Guarantee normal operation and normal user experience when unexpected transitions occur (e.g. battery disconnect / reconnect)

2.4 Scenarios

2.4.1 Hardware Power Transitions

This section describes the specific hardware-level operations for various power transitions.

2.4.1.1 Power On

[P06.R017.10] Power On is entered when battery voltage is first connected to the Sync board. The VMCU will come out of reset and enable the on-board and CAN power supplies. The VMCU, GPS, and the DR Sensors are all powered from the VCC_3P3VBAT. This supply is always on if battery voltage is applied to the SYNC. The VMCU will then enable the CCPU main voltage rail to bring up the CCPU subsystem.

When the CCPU first boots up it always enters the WAIT_ON state, where it will remain indefinitely (assuming IPC link is successfully established) until directed by the VMCU

1. VMCU initializes the port Pins -
2. VMCU enables the CLKOUT1_32K_GPS signal. (This signal is always on)
3. VMCU generates the VMCU_2MHZ_OUT (2MHz PWM 50% duty cycle) Not used yet
4. VMCU asserts the VCC_5V_SW_EN to High Enables the VCC_5V_SW supply. This supply is needed for CAN communication.
5. VMCU asserts the VMCU_ADC_EN to High. This enables the battery ADC measurement.
6. VMCU waits for 310ms (10msec is for softstart timing for the VCC_5V_SW supply and RC for battery. The additional 300ms is to insure the display power is off for 1sec)
7. VMCU checks VCC_5V_SW_PG = 1 Insures supply is operational
8. VMCU reads BRD_VAR 0-3
9. VMCU reads BRD_VER 0-3
10. VMCU reads BOM_VER 0-3
11. Battery voltage > 7V
12. VMCU enables and initializes all CAN and LIN controllers
13. VMCU enables and initializes the GNSS and the DR sensor
14. VMCU generates the VMCU_400KHZ_OUT (400Khz PWM 50% duty cycle) Not used yet
15. VMCU asserts the main power voltage rail (MAIN_PWR_EN) and GNSS_ANT_PWR_EN after VMCU and XTAL are stable.
16. VMCU sets the CCPU_SCU_CONFIG2 and 3 signals (CCPU boot option)
17. VMCU sets the CCPU_RECOVERY_N and the CCPU_SW_UPDATE signals. (CCPU update option)
18. VMCU sets the VMCU_LVI signal
19. VMCU sets the VMCU_E_SHTDWN signal

20. VMCU sets the DOOR_OPEN_IND signal
21. VMCU sets the REVERSE_MODE signal
22. VMCU sets the STARTUP_ANIMATION signal
23. VMCU sets up IPC UART signals
24. VMCU waits for 60ms (Soft start time of Buck Boost is 120msec. This is the first part of the wait cycle.)
25. VMCU checks GNSS_ANT connect signal and error signals.
26. VMCU asserts CCPU_PMIC_EN_N low. (This enables the PMIC supply)
27. VMCU waits 10ms (Softstart time for the PMIC supplies)
28. VMCU asserts VCC_5V_USB_EN high.
29. VMCU waits 70msec (Wait time to finish Buck/Boost softstart, PMIC, and USB supplies)
30. VMCU checks VCC_16V_PG signal (Failure of this supply will prevent Display, MCM, and DRVC)
31. VMCU checks the PMIC_PWR_PGOOD signals. (These are not functional till DV units.)
32. VMCU checks the VCC_USB_PG signal. (Failure of this supply will prevent USB)
33. VMCU asserts VCC_8V_EN high.
34. VMCU asserts VCC_9V_EN high. (Only for DRVC)
35. VMCU asserts MCM_EN high.
36. VMCU waits for 530msec
37. VMCU checks VCC_9V_PG, VCC_8V_DIAG, and MCM_PWR_DIAG.
38. VMCU de-asserts DSPL_EN(Low)
39. VMCU waits for 20msec

40. VMCU checks the DSPL_PWR_DIAG.
41. CCPU power up (POR).
42. VMCU follows logic from CCPU for DSPL_EN.
43. VMCU waits for IPC link to be established; if IPC link is not detected with timeout (IPC_EST_TRY_TIME defined in S06), VMCU will perform Cold Reboot of the CCPU upto IPC_EST_RETRIES times. If establishing an IPC link still fails, VMCU removes power from the CCPU. (see P06.R024.06).
44. CCPU Power Manager enters <Wait-On>, establishes IPC link.
45. CCPU acknowledges Wakeup via SET_PM_STATE_COMPLETE with TID=0.
46. VMCU awaits SET_PM_STATE_COMPLETE or the timer CCPU_PM_STATE_COMPLETE_TIMER has expired
47. VMCU commands CCPU to enter the desired power state (PM_SET_PM_STATE)
    Note: VMCU Power Manager should continue to refresh the wakeup source every time it sends SET_PM_STATE() request up until it receives the SET_PM_STATE_COMPLETE() response from CCPU
48. CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters the commanded power state
49. CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)
50. VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource0, WakeSource1, and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b.
    Note: WakeSource0 is set to Voltage. WakeSource 1 and 2 and 3 indicate the most recent wakeup reason.
51. CCPU notifies applications of wakeup event

[END]

2.4.1.2 Low or High Battery

[P06.R018.05]Low or high battery condition is entered when battery voltage drops below 7V or rises above 19V. When a low or high battery condition is detected, the VMCU shall set the VMCU_E_SHTDWN signal. While VMCU set VMCU_E_SHTDWN signal to active the CCPU should pause operations and prevent writes to flash. Typically, the low or high battery condition is a result of the vehicle engine cranking event; but a battery disconnect event or alternator fault could also cause entry into cranking power state.

1. Battery voltage < 7V or Battery voltage > 19V.
2. VMCU sets VMCU_E_SHTDWN signal active.
3. CCPU suspends eMMC flash writes while VMCU_E_SHTDWN is active.

In addition, the following limitations are placed on the CCPU power mode transitions when the VMCU_E_SHTDWN signal is asserted:

1. The VMCU will not take the CCPU out of INFOTAINMENT, WAIT_ON, DISPLAYONLY, or VHM while the VMCU_E_SHTDWN signal is asserted.

2. If the CCPU is in the Wait-Suspend state and the VMCU_E_SHTDWN signal is asserted, the VMCU must transition the CCPU into Wait-On until VMCU_E_SHTDWN is no longer asserted.
3. After 10s of VMCU_E_SHTDWN being asserted, the VMCU shall immediately transition to CCPU Shutdown for the remainder of the current IGN cycle and de-assert the VMCU_E_SHTDWN pin.
4. If the voltage recovers to above 7.2V for 1 second after <10s or 18.8V for 1 second in Low or High Battery, then the VMCU shall de-assert the VMCU_E_SHTDWN pin.

[END]

2.4.1.3 CCPU Shutdown

[P06.R019.05]Shutdown/Unpowered state is entered when the VMCU commands the CCPU into the Wait Shutdown power mode. This mode will not require the VCC_5V_SW or the VCC_16V_OUT to be disabled. The VCC_5V_SW is required for CAN communication. The VCC_16V_OUT should only be shutdown when the SYNC module VMCU is going into VMCU LPU/standby. The VCC_5V_SW should only be disabled if the VMCU is done with all CAN messages and going into VMCU LPU/standby.

1. If the VMCU is performing a rapid shutdown of the CCPU (Rapid Shutdown) and has to bypass the normal Suspend process, then the VMCU shall assert the VMCU_E_SHTDWN pin for 500ms.
2. VMCU de-asserts MCM_EN (Low)
3. VMCU de-asserts DSPL_EN (Low)
4. VMCU de-asserts VCC_9V_EN (Low)
5. VMCU de-asserts VCC_8V_EN (Low)
6. VMCU de-asserts VCC_5V_USB_EN (Low)
7. VMCU de-asserts START_UP_ANIMATION (Low)
8. VMCU de-asserts VMCU_UART_TXD (places VMCU_UART_TXD in sleep state)
9. VMCU de-asserts VMCU_LVI (Low)
10. VMCU de-asserts REVERSE_MODE (Low)
11. VMCU de-asserts DOOR_OPEN_IND (Low)
12. VMCU de-asserts CCPU_PMIC_EN_N (HIGH) This turns off the PMIC power.
    Note: steps 7 to 12 shall be performed within 10 ms
13. VMCU wait 20ms VMCU de-asserts VMCU_E_SHTDWN
14. VMCU wait 300ms to allow power rails to drop low and ensure full reset

[END]

2.4.1.4 CCPU Stop/Sleep

[P06.R019.05]CCPU Stop/Sleep state is entered when the VMCU commands the CCPU into the Stop/Sleep power mode. This mode is used to maintain the CCPU section in a low power mode for quick boot. This mode will require the VCC_5V_SW and the PMIC power to be enabled. The VMCU needs to maintain the GPIO in the Stop/Sleep mode. The CCPU must maintain the DISPLAY_EN _N signal high in Stop/Sleep mode to insure the display power is off. The CCPU is also required to insure the 1 second guarantee off time is maintained for proper display start up.

1. VMCU de-asserts MCM_EN (Low)
2. VMCU follows logic from CCPU DISPLAY_EN_N for DSPL_EN
3. VMCU de-asserts VCC_9V_EN (Low)
4. VMCU de-asserts VCC_8V_EN (Low)
5. VMCU de-asserts VCC_5V_USB_EN (Low)
6. VMCU de-asserts MAIN_PWR_EN (Low) This disables buck/boost and main FET.
7. VMCU de-asserts START_UP_ANIMATION (Low)
8. VMCU de-asserts VMCU_UART_TXD (places VMCU_UART_TXD in sleep state)
9. VMCU asserts VMCU_LVI (High)
10. VMCU de-asserts REVERSE_MODE (Low)
11. VMCU de-asserts DOOR_OPEN_IND (Low)
12. VMCU de-asserts VMCU_E_SHTDWN (Low)
13. VMCU then transitions to a Stop mode. This mode must maintain the GPIO state to the CCPU and the 5V power supply and PMIC enabled. It is assumed the VMCU would place the GNSS into a low power sleep mode, turn off the GNSS antenna power (GNSS_ANT_PWR_EN = Low), place the Accel/Gyro into sleep mode, the CAN and LIN controllers into standby and sleep. The wakeup from this mode will **not** require the

VMCU to reboot. This mode must maintain the RTC clock output. Note: The power mode for the VMCU and the other VMCU IO devices are intended to keep the software design and verification simple.

The VMCU will transition to the Standby state and follow the Shutdown sequence in section 2.4.1.3 if any of the following conditions are met during a CAN wakeup:

- PwLoApim_T_Actl == 0 minutes (shutdown)
- The Stop State Timer has expired. The Stop State Timer will be maintained and checked by the RTC interrupt.
- << LoadShed Active >> -OR- <<KeyOffMde Critical Battery Active>>

[END]

2.4.1.5 CCPU Start/Resume

[P06.R065.04] The CCPU Start/Resume procedure is executed when the VMCU wakes up the CCPU from the Stop/Sleep state. The VMCU is in STOP and requires a CAN or LIN wakeup event. It also might use the real time clock as a failsafe wakeup transition to CCPU shutdown state.

1. CCPU is in Stop/Sleep state and VMCU is in Stop state.
2. VMCU checks Battery voltage > 7V.
3. VMCU enables and initializes all CAN and LIN controllers
4. VMCU checks for IGN status or other CAN wake up events. If there are no wakeup events the VMCU goes back to STOP.
5. VMCU checks for battery status charge to insure the wake up is not to place the CCPU into shutdown due to battery charge state. The VMCU may also check the Real Time Clock wake up to transition into CCPU shutdown mode if the fail safe timer has expired.
6. Check to see if this is a valid START for the CCPU. IF yes proceed.
7. VMCU asserts the main power voltage rail (MAIN_PWR_EN) and the GNSS_ANT_PWR_EN.
8. VMCU sets the VMCU_E_SHTDWN signal
9. VMCU sets the DOOR_OPEN_IND signal
10. VMCU sets the REVERSE_MODE signal
11. VMCU sets the STARTUP_ANIMATION signal
12. VMCU sets up IPC UART signals
13. VMCU waits for 70ms (Soft start time of Buck Boost is 140msec. This is the first part of the wait cycle.)
14. VMCU checks GNSS_ANT connect signal and error signals.
15. VMCU asserts VCC_5V_USB_EN high.
16. VMCU waits 70msec (Wait time to finish Buck/Boost softstart and USB supplies)
17. VMCU checks VCC_16V_PG signal (Failure of this supply will prevent Display, MCM, and DRVC)
18. VMCU checks the VCC_USB_PG signal. (Failure of this supply will prevent USB)
19. VMCU asserts VCC_8V_EN high.
20. VMCU asserts VCC_9V_EN high. (Only for DRVC)
21. VMCU asserts MCM_EN high.
22. VMCU waits 20msec (wait for supplies to stabilize.)
23. VMCU checks VCC_9V_PG, VCC_8V_DIAG, and MCM_PWR_DIAG
24. VMCU drives VMCU_LVI low to signal the CCPU to Wake up
25. CCPU wakes up
26. VMCU follows logic from CCPU DISPLAY_EN_N for DSPL_EN.
27. VMCU waits for IPC link to be established; if IPC link is not detected with timeout (IPC_EST_TRY_TIME defined in S06), VMCU will perform Reboot of the CCPU (up to IPC_EST_RETRIES total times), after which it will remove power from the CCPU.
28. CCPU Power Manager enters <Wait-On>, establishes IPC link
29. CCPU acknowledges Wakeup via SET_PM_STATE_COMPLETE or the timer CCPU_PM_STATE_COMPLETE_TIMER has expired
30. VMCU commands CCPU to enter the desired power sate (PM_SET_PM_STATE)
    Note: VMCU Power Manager should continue to refresh the wakeup source every time it sends SET_PM_STATE() request up until it receives the SET_PM_STATE_COMPLETE() response from CCPU
31. CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters the commanded power state
32. CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)

33. VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource0, WakeSource1, and WakeSource2 set to indicate wake source as defined in S23b.
    Note: WakeSource0 is set to None. WakeSource 1 and 2 indicates the most recent wakeup reason.
34. CCPU notifies applications of wakeup event

[END]

2.4.1.6 Wakeup

[P06.R065.04] The Wakeup procedure is executed when the VMCU wakes up the CCPU from the Unpower state. The VMCU is in LPU/standby and requires a CAN or LIN wakeup event.

35. CCPU is Shutdown/Unpowered and VMCU is in LPU/standby state.
36. The VMCU resumes the port pin and the port definitions.
37. VMCU generates the VMCU_2MHZ_OUT (2MHz PWM 50% duty cycle) Not used yet
38. VMCU asserts the VCC_5V_SW_EN to High Enables the VCC_5V_SW supply. This supply is needed for CAN communication.
39. VMCU asserts the VMCU_ADC_EN to High. This enables the battery ADC measurement.
40. VMCU waits for 10ms (Softstart timing for the VCC_5V_SW supply and RC for battery)
41. VMCU checks VCC_5V_SW_PG = 1 Insures supply is operational
42. Battery voltage > 7V.
43. VMCU enables and initializes all CAN and LIN controllers
44. VMCU checks for IGN status or other CAN wake up events. If there are no wakeup events the VMCU goes back to Standby.
45. VMCU enables and initializes the GNSS and the DR sensor
46. VMCU generates the VMCU_400KHZ_OUT (400Khz PWM 50% duty cycle) Not used yet
47. VMCU asserts the main power voltage rail (MAIN_PWR_EN) and the GNSS_ANT_PWR_EN.
48. Check to see if this is a valid wakeup for the CCPU. IF yes proceed.
49. VMCU sets the CCPU_SCU_CONFIG2 and 3 signals (CCPU boot option)
50. VMCU sets the CCPU_RECOVERY_N and the CCPU_SW_UPDATE signals. (CCPU update option)
51. VMCU sets the VMCU_LVI signal
52. VMCU sets the VMCU_E_SHTDWN signal
53. VMCU sets the DOOR_OPEN_IND signal
54. VMCU sets the REVERSE_MODE signal
55. VMCU sets the STARTUP_ANIMATION signal
56. VMCU sets up IPC UART signals
57. VMCU waits for 60ms (Soft start time of Buck Boost is 120msec. This is the first part of the wait cycle.)
58. VMCU checks GNSS_ANT connect signal and error signals.
59. VMCU asserts CCPU_PMIC_EN_N low. (This enables the PMIC supply)
60. VMCU waits 10ms (Softstart time for the PMIC supplies)
61. VMCU asserts VCC_5V_USB_EN high.
62. VMCU waits 70msec (Wait time to finish Buck/Boost softstart, PMIC, and USB supplies)
63. VMCU checks VCC_16V_PG signal (Failure of this supply will prevent Display, MCM, and DRVC)
64. VMCU checks the PMIC_PWR_PGOOD signals. (These are not functional yet.)
65. VMCU checks the VCC_USB_PG signal. (Failure of this supply will prevent USB)
66. VMCU asserts VCC_8V_EN high.
67. VMCU asserts VCC_9V_EN high. (Only for DRVC)
68. VMCU asserts MCM_EN high.
69. VMCU waits for 530msec
70. VMCU checks VCC_9V_PG, VCC_8V_DIAG, and MCM_PWR_DIAG
71. VMCU de-asserts DSPL_EN(Low).
72. VMCU waits 20msec
73. VMCU checks DSPL_PWR_DIAG.
74. CCPU wakes up (The CCPU starts its boot process after the CCPU_PMIC_EN is driven high)
75. VMCU follows logic from CCPU for DSPL_EN.
76. VMCU waits for IPC link to be established; if IPC link is not detected with timeout (IPC_EST_TRY_TIME defined in S06), VMCU will perform Reboot of the CCPU (up to IPC_EST_RETRIES total times), after which it will remove power from the CCPU.
77. CCPU Power Manager enters <Wait-On>, establishes IPC link
78. CCPU acknowledges Wakeup via SET_PM_STATE_COMPLETE or the timer CCPU_PM_STATE_COMPLETE_TIMER has expired

79. VMCU commands CCPU to enter the desired power sate (PM_SET_PM_STATE)
    Note: VMCU Power Manager should continue to refresh the wakeup source every time it sends SET_PM_STATE() request up until it receives the SET_PM_STATE_COMPLETE() response from CCPU
80. CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters the commanded power state
81. CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)
82. VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource0, WakeSource1, and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b.
    Note: WakeSource0 is set to None. WakeSource 1 and 2 and 3 indicates the most recent wakeup reason.
83. CCPU notifies applications of wakeup event

[END]

2.4.1.7 VMCU Reset/Reboot

The VMCU Reset/Reboot is performed for the ECU $11 command and during a VMCU code update. All the SYNC 4 power supplies are disabled for this command. This guarantees all the supplies will be off for at least 20msec. The VMCU reset command will then start at the Power ON.

To enact a VMCU Reset/Reboot, the VMCU shall perform the following sequence:

1. VMCU sets “HMIAudioMode == OFF” or “HMIAudioMode == LoadShed” (if «LoadShed Active»), if not already
2. VMCU sets “InfoSysMasterPw_D_Stat == Inactive”, if not already
3. VMCU tears down the IPC Link
4. VMCU asserts VMCU_E_SHTDWN
5. VMCU delays 200ms
6. VMCU de-asserts the VCC_5V_SW_EN (Low) (No CAN messages after this power is disabled)
7. VMCU de-asserts the MAIN_PWR_EN (Low)
8. VMCU de-asserts the GNSS_ANT_PWR_EN (Low)
9. VMCU de-asserts MCM_EN (Low)
10. VMCU de-asserts DSPL_EN (Low)
11. VMCU de-asserts VCC_9V_EN (Low)
12. VMCU de-asserts VCC_8V_EN (Low)
13. VMCU de-asserts VCC_5V_USB_EN (Low)
14. VMCU de-asserts START_UP_ANIMATION
15. VMCU de-asserts VMCU_UART_TXD (places VMCU_UART_TXD in sleep state)
16. VMCU de-asserts VMCU_LVI (Low)
17. VMCU de-asserts REVERSE_MODE (Low)
18. VMCU de-asserts DOOR_OPEN_IND (Low)
19. VMCU de-asserts CCPU_PMIC_EN_N (HIGH) This turns off the PMIC power.
    Note: steps 7 to 18 shall be performed within 10 ms
20. VMCU wait 20ms VMCU de-asserts VMCU_E_SHTDWN
21. VMCU executes a Reset

The reboot will follow the Power ON sequence.

2.4.1.8 CCPU Reset/Reboot

[P06.R020.10]CCPU Shutdown is performed when a fault event is detected.

To enact a CCPU Reset/Reboot, the VMCU shall perform the following sequence:

1. VMCU sets “HMIAudioMode == OFF” or “HMIAudioMode == LoadShed” (if «LoadShed Active»), if not already
2. VMCU sets “InfoSysMasterPw_D_Stat == Inactive”, if not already
3. VMCU tears down the IPC Link
4. VMCU asserts VMCU_E_SHTDWN
5. VMCU delays 500ms

6. VMCU de-asserts MCM_EN (Low)
7. VMCU de-asserts DSPL_EN (Low)
8. VMCU de-asserts VCC_9V_EN (Low)
9. VMCU de-asserts VCC_8V_EN (Low)
10. VMCU de-asserts VCC_5V_USB_EN (Low)
11. VMCU de-asserts START_UP_ANIMATION
12. VMCU de-asserts VMCU_UART_TXD (places VMCU_UART_TXD in sleep state)
13. VMCU de-asserts VMCU_LVI (Low)
14. VMCU de-asserts REVERSE_MODE (Low)
15. VMCU de-asserts DOOR_OPEN_IND (Low)
16. VMCU de-asserts CCPU_PMIC_EN_N (HIGH) This turns off the PMIC power.
    Note: steps 7 to 15 shall be performed within 10 ms
17. VMCU wait 20ms VMCU de-asserts VMCU_E_SHTDWN
18. Delay 1 second for voltage rails to drop.
19. VMCU sets the CCPU_SCU_CONFIG2 and 3 signals (CCPU boot option)
20. VMCU sets the CCPU_RECOVERY_N and the CCPU_SW_UPDATE signals. (CCPU update option)
21. VMCU sets the VMCU_LVI signal
22. VMCU sets the VMCU_E_SHTDWN signal
23. VMCU sets the DOOR_OPEN_IND signal
24. VMCU sets the REVERSE_MODE signal
25. VMCU sets the STARTUP_ANIMATION signal
26. VMCU sets up IPC UART signals
27. VMCU asserts CCPU_PMIC_EN_N (Low) Enable the PMIC
28. VMCU sets USB_GND_EN_N to low
29. VMCU waits 10msec
30. VMCU asserts VCC_5V_USB_EN (High)
31. VMCU waits 10msec
32. VMCU de-asserts DSPL_EN (Low)
33. VMCU asserts MCM_EN (High)
34. VMCU asserts VCC_8V_EN (High)
35. VMCU asserts VCC_9V_EN (High)
36. VMCU checks for VCC_5V_USB_PG, DSPL_PWR_DIAG, MCM_PWR_DIAG, VCC_8V_DIAG, VCC_9V_PG and PMIC_PWR_GOOD signals for 100msec. (Note: PMIC_PWR_GOOD not available for testing until DV units. VCC_9V_PG only valid for DRVC units.)
37. CCPU will go through all the initialization
38. CCPU Power Manager enters <Wait-On>, establishes IPC link
39. CCPU acknowledges Wakeup via SET_PM_STATE_COMPLETE with TID-0.
40. VMCU awaits SET_PM_STATE_COMPLETE or the timer CCPU_PM_STATE_COMPLETE_TIMER has expired
41. VMCU commands CCPU to enter the desired power state (PM_SET_PM_STATE)
    Note: VMCU Power Manager should continue to refresh the wakeup source every time it sends SET_PM_STATE() request up until it receives the SET_PM_STATE_COMPLETE() response from CCPU
42. CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters the commanded power state
43. CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)
44. VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource0, WakeSource1, and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b.
    Note: WakeSource0 is set to CCPU Reboot. WakeSource 1 and 2 and 3 indicate the most recent wakeup reason.
45. CCPU notifies applications of wakeup event

The SYNC 4 only requires the proper pin direction definition. The power sequence of pins between the VMCU and the CCPU is handled in hardware.

If the VMCU detects that any failure or retry logic around multiple resets of the CCPU has been met, then the VMCU should follow the CCPU Shutdown procedure as the final step for that ignition cycle.

The VMCU will perform a CCPU Reset/Reboot for the following reasons:

1. Non-voltage fault condition (IPC link, etc.) detected

2. CCPU request to COLD Reset (Bit 5 == 1 of PM_ACTION_FLAG in SET_PM_STATE_COMPLETE from CCPU)
3. CCPU request to COLD Reset via SET_PM_STATE(REBOOT)
4. CCPU fails to establish IPC link at power-on
5. CCPU Watchdog timeout
6. SET_PM_STATE failures

The CCPU shall use the VMCU_E_SHTDWN assertion as an indication to save any state, flush file buffers, etc. before the reboot sequence is started, and must ensure that application level data corruption is not possible in the CCPU (flash level data corruption must be protected as well (e.g. via the CCPU NAND Flash Driver)).

If B+ is faulted (<9.5V or >16V), the VMCU shall ignore ANY CCPU reboot request (Bit5==1 or Bit6==1 of PM_ACTION_FLAG in SET_PM_STATE_COMPLETE during a transition to Suspend)..

The VMCU shall ignore the CCPU reboot request (via Bit5==1 or Bit 6 == 1 of PM_ACTION_FLAG set in SET_PM_STATE_COMPLETE reply from CCPU) if the VMCU commanded the Suspend (via SET_PM_STATE) and the VMCU must transition into Infotainment or DisplayOnly. (This prevents the CCPU from enacting a scheduled reboot if the SYNC must immediately re-enter Infotainment from an interrupted Suspend cycle).

When the CCPU reboots after the reset, it always enters the WAIT_ON state, where it will remain indefinitely (assuming IPC link is successfully established) until directed by the VMCU.
[END]

2.4.1.9 CAN/LIN Power Control

[P06.R021.06] CAN needs the VCC_5V_SW power turned on to communicate.

Power On:

1. VMCU enables CAN and LIN controller interface.
   HSxCAN_STB = L ; LIN4_SLP_N = H

Low Power:

1. VMCU sets the STB and SLP_N signal to enter standby or sleep mode.
   HSxCAN_STB = H ; LIN4_SLP_N =L
   Note: Any CAN interface or LIN interface which is not used should be placed in sleep or Standby mode to save power.

LPU/STANDBY WAKE:

1. VMCU should disable CAN and LIN controller interface.
2. The CAN_RXD and the LIN_RXD pins must be configured for wakeup
3. The VCC_5V_SW can be turned off in this mode. The CAN transceiver only needs the VCC_3P3VBAT.
4. The CAN port should be placed in standby.
5. The LIN port should be placed in sleep.
6. The VMCU can be placed in LPU/standby state.
7. A low transition on the CAN_RXD or the LIN_RXD will wake up the VMCU.
   Note: If VCC_5V_SW is disabled the CAN port will be in standby mode.

[END]

2.4.1.10 Reverse Gear Control

[P06.R066.01] The VMCU shall toggle a dedicated hardware I/O to the CCPU to indicate Reverse Gear detection such that the CCPU can have a quick trigger for RVC

«Reverse Gear Active»:

1. VMCU enables the REVERSE_MODE signal

«Reverse Gear Inactive»:

1. VMCU disables the REVERSE_MODE signal

[END]

2.4.1.11 Door Open Control

[P06.R067.01] The VMCU shall toggle a dedicated hardware I/O to the CCPU to indicate whenever the Driver or Passenger doors are opened/closed such that the CCPU can have a quick trigger to display the Welcome Animation in the HMI.

Door Open detected:
1. VMCU enables the DOOR_OPEN_IND signal

Door Close detected:
1. VMCU disables the DOOR_OPEN_IND signal

[END]

2.4.1.12 Audio Power Control

[P06.R022.03]The VMCU enables the VCC_5V_SW voltage to the VA rail of the CODEC; the VREF and VL rails are not switchable. The VMCU also enables the 8V LDO to provide VCC_8V voltage to the audio input/output opamp circuits and the microphone phantom power.

Power On:
1. CCPU (on POR) sets the CODEC_I2S_EXT_MCLK.
2. Delay (10msec min) to wait for the clock to become stable.
3. CCPU sets CODEC_RST_N signal low.
4. Delay (2msec min) for reset pulse width specification.
5. CCPU sets AUD_CODEC_RST_N signal high.
6. Delay (2msec min) for reset to I2C start specification.
7. CCPU configures the digital audio interface to the CODEC.
8. CCPU sends I2C commands to setup the CODEC.

Low Power:
1. CCPU sends I2C commands to write Power Control register to place codec in low power mode.
2. This is done for the DAC's and the ADC's.
3. If all analog audio including microphones is disabled, the VCC_8V_EN can be powered down. This would require an IPC call to disable the VCC_8V.

[END]

2.4.1.13 Video Power Control

[P06.R023.04]All voltage rails for the Video CODEC are not switchable. The CCPU can use a low power input to place the video CODEC into low power mode. The recommended low power mode is using the I2C registers.

Power On:
1. CCPU sets ARVC_RST_N signal low.
2. CCPU sets ARVC_PD_N signal high.
3. Delay (5msec min) for reset pulse width specification.
4. CCPU sets ARVC_RST_N signal high.
5. CCPU configures the digital video interface to the CODEC.
6. CCPU sends I2C commands to setup the CODEC.

Low Power:
1. CCPU uses the I2C interface to place the Video Codec in low power mode when the rear camera is not enabled.

[END]

2.4.1.14 Wireless (Bluetooth / WiFi) Power Control

[P06.R060.02] All voltage rails for the Radio Module are not switchable. The CCPU uses a power down input to place the Radio Module into low power mode.

Power On:

1. CCPU sets WIFI_BT_PD_N signal high.
2. CCPU sets PCIE1_RESET_N signal low.
3. Delay (100msec min) for Bluetooth component of Radio Module to reset.
4. CCPU sets PCIE1_RESET_N signal high.

Low Power:
1. CCPU sets WIFI_BT_PD_N signal Low.

[END]

2.4.1.15 Display Power Module Power Control

[P06.R025.03]The CCPU use 2 signals to control the display module. The DISPLAY_PDB_N signal will power up or down the FPDLINKKIII serializer. This serializer is used to connect to the display module. The CCPU also provides a DISPLAY_EN_N signal to the VMCU. The VMCU uses this signal to control the power to the display module.

Power On:
1. CCPU sets DISPLAY_PDB_N signal high.
2. CCPU waits 10msec for the I2C communication to be established
3. CCPUconfigures the LVDS display channel..
4. CCPU sets DISPLAY_EN_N signal low.
5. Delay (200msec min) for voltage rails to become stable and I2C communication to be enabled.
6. CCPU enables the LCD controller to display image on the LCD.
7. CCPU increases backlight brightness by sending I2C command to module.

Power Off:
1. CCPU reduces backlight brightness by sending I2C command to module.
2. CCPU turns the LCD module off using the I2C commands.
3. CCPU disables the LCD controller (data/sync/clk signals).
4. CCPU sets DISPLAY_EN_N signal low.

[END]

2.4.1.16 USB Power Control CCPU

[P06.R059.03] The CCPU controls the power to the single USB port by using the USB_SWITCH_CTL1_OR_COMBINED pin. If this pin is low the USB port connector will not have power. If this pin is high the USB port has power.

Power On:
CCPU sets the USB_SWITCH_CTL1_OR_COMBINED pin to high.Low Power:
CCPU sets the USB_SWITCH_CTL1_OR_COMBINED pin to low.[END]

[P06.R028.02]
Requirement is obsolete.
[END]

2.4.1.17 KeyIn Animation Control

[P06.R068.02] The VMCU must monitor the CAN bus and assert the START_UP_ANIMATION GPIO pin if these CAN signals are received with the values specified.

KeyIn:
1. BodyInfo_3_HS3::Ignition_Status, values Accessory(0x2) or Run (0x04) or Start (0x08)

The GPIO shall remain asserted until none of the signals report the value as per the list above.

For all other situations the GPIO has to remain de-asserted.
[END]

2.4.1.18 DRVC Power Control

[P06.R070.02] The VMCU shall monitor Inbound Diagnostics OID Digital RVC Power Control (250:007) and switch the VCC_9V_EN according to this conditions:

1. Power supply shall default to ON at boot up.
2. CCPU can send the OID to either power up or power down
    a. Power down requires no logic. It's a command that is followed at all times.
    b. Power up requires an evaluation of VCC_9V_PG 20ms after VCC_9V_EN becomes asserted.
        i. If this VCC_9V_PG evaluates to power stable, a continuous DTC shall be handled as fault not present in accordance with the IDS.
        ii. Else a continuous DTC shall be handled as fault present in accordance with the IDS.

The GPIO shall be asserted according to the list above.
[END]

2.4.1.19 USB Ground Protection

The VMCU shall protect the Ground plane from high currents coming in through the USB Ground connection. E.g. when a USB cable is connected and the shielding makes contact with a 12V power port, extremely high inrush currents can occur and damage components or the PCB itself.

To accomplish this three nets are connected to the VMCU:

Name	Direction	Type	Signal Description	Condition
USB_GND_DET	Input	Digital	L: USB Ground has no short to battery H: USB Ground has a short to battery	Only valid when USB_GND_EN_N is High (disconnected).
USB_GND_EN_N	Output	Digital	L: USB Ground connected to device Ground H: USB Ground separated from device Ground	Requires VCC_5V_USB to be High to connect to device Ground.
USB_GND_SNS_OUT	Input	Analog	Voltage proportional to USB Ground inrush current: Voltage = Current * 0.01Ohm	Only valid when USB_GND_EN_N is Low (connected).

[P06.R071.02]
Before USB_GND_EN_N is asserted, the VMCU shall monitor the USB_GND_DET digital net to confirm that the USB Ground does not have a short to battery. Only if no short to battery is stable for at least 50 milliseconds, USB_GND_EN_N shall be asserted.
[END]

[P06.R072.02]
When USB_GND_EN_N is asserted, the VMCU shall monitor the USB_GND_SNS_OUT analog net to detect a high current (>20A) on the USB Ground connection. Once this current is detected, the USB_GND_EN_N has to be de-asserted within 10ms.
[END]

[P06.R073.02]
After a high current has been detected, the VMCU shall not assert USB_GND_EN_N until USB_GND_DET indicates "no short to battery" and is stable for at least 500 milliseconds.
[END]

The following is an implementation proposal for the Calypso:

1. USB_GND_EN_N shall default to GPIO mode and level H from the bootloader at boot up.
2. During power up or wake up the VMCU sets USB_GND_EN_N to GPIO mode and level H. Then it enters USB_Gnd_Check State:

USB_Gnd_Check State:
1. VMCU repeatedly reads USB_GND_DET:
    a. If USB_GND_DET == L:
        i. If USB_GND_DET == L for >= 50 ms:
            1. VMCU enabled CMP module.
            2. VMCU switches USB_GND_EN_N to CMP Out mode.
            3. VMCU enter USB_Gnd_OK State.
        ii. If USB_GND_DET == L for < 50 ms:
            1. VMCU remains in USB_Gnd_Check State.
    b. Else:
        i. VMCU remains in USB_Gnd_Check State.

USB_Gnd_OK State:
1. VMCU CMP compares USB_GND_SNS_OUT to reference voltage:
    a. If USB_GND_SNS_OUT >= 200 mV:
        i. VMCU CMP immediately switches CMP Output to High (no SW involved)
        ii. VMCU IRQ switches USB_GND_EN_N to GPIO mode and level H.
        iii. VMCU enters USB_Gnd_Short State.
    b. Else:
        i. VMCU remains in USB_Gnd_OK State.

USB_Gnd_Short State:
2. VMCU repeatedly reads USB_GND_DET:
    a. If USB_GND_DET == L:
        i. If USB_GND_DET == L for >= 500 ms:
            1. VMCU switches USB_GND_EN_N to CMP Out mode.
            2. VMCU enter USB_Gnd_OK State.
        ii. If USB_GND_DET == L for < 500 ms:
            1. VMCU remains in USB_Gnd_Short State.
    b. Else:
        i. VMCU remains in USB_Gnd_Short State.

When the VMCU powers down or enters sleep mode, USB Ground is disconnected and the CMP is disabled.

USB_Gnd_Off State:
1. VMCU switches USB_GND_EN_N to GPIO mode and level H.
2. VMCU disables CMP module.

2.4.2 Power Mode Transitions

The following power mode transitions, including the communication between the CCPU and VMCU (direct I/O and IPC), are detailed below. In addition, SPSS defined power modes, as well as power-related application behavior, is described for clarity.
For these Power Mode Transitions there shall be no different handling, if HMI On/Off is received from EFP Front only or EFP Rear only or any combination of both.

2.4.2.1 LPU/Standby or Unpowered to Infotainment [T01]

[P06.R038.09]
SPSS Scenarios:
Enter MMActive Mode (via Run/ACC) (PWRMAN-GUC-40493-X)
Enter Extended Play Mode (via HMI button press) (PWRMAN-GUC-40451-X)

Application Scenarios:
N/A

Transition Sequence:
Scenario A: VMCU is in LPU/Standby
- CCPU is Unpowered, CAN and LIN networks inactive
- VMCU detects CAN wakeup or LIN wakeup and wakes up
- VMCU detects one of the following conditions:
    1. «ACC Active» –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    2. «Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive») –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    3. SYNC is configured for Extended Play –AND– <HMI On/Off> received via CAN –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    4. <eCallNotification == Active>
    5. «LBI Password Screen Active»
- VMCU follows Wakeup sequence (Section 2.4.1.6)

- CCPU sets “HMIAudioMode == ON” (within 650 to 700ms after bus awake)
- CCPU Power Manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE)
- VMCU starts Extended Play timer, if entered via <HMI On/Off>
- VMCU starts LBI Interaction Watchdog Timer, if entered via «LBI Password Screen Active»
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <Infotainment> state
- CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- CCPU notifies applications of wakeup event via Event Manager

Scenario B: VMCU Unpowered

- VMCU and CCPU are Unpowered
- Power is applied
- VMCU follows Power On sequence (Section 2.4.1.1)
- VMCU detects one of the following conditions:
    1. «ACC Active» –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    2. «Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive») –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    3. SYNC is configured for Extended Play –AND– <HMI On/Off> received via CAN –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    4. <eCallNotification == Active>
    5. «LBI Password Screen Active»
- CCPU sets “HMIAudioMode == ON” (within 650 to 700ms after bus awake if VMCU was powered, or within 1500 to 1550ms after bus awake if VMCU was unpowered)
- CCPU Power Manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE)
- VMCU starts Extended Play timer, if entered via <HMI On/Off>
- VMCU starts LBI Interaction Watchdog Timer, if entered via «LBI Password Screen Active»
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <Infotainment> state
- CCPU Power Manager requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- CCPU notifies applications of wakeup event via Event Manager

[END]

2.4.2.2 Unpowered to DisplayOnly [T02]

[P06.R039.14]
SPSS Scenarios:

Application Scenarios:

SYNC “Welcome” strategy
CD Insert/Eject HMI indication
Transport Mode

Transition Sequence:

Scenario A: VMCU is in LPU/Standby

- CCPU is in Unpowered, CAN and LIN networks inactive
- VMCU detects CAN wakeup or LIN wakeup and wakes up
- VMCU detects one of the following condition:
    1. «Doors Opened» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»

2. <CD Eject> via CAN or LIN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
3. <CD Insert> via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
4. <Run/ACC Event> –AND– «LoadShed Active»
5. «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
6. «Run Active» –AND– «EngineOn Inactive» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
7. «Door Unlock» via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
8. «Illumination Keep Active»
9. «PaaK Charging Active»
10. «Predictive trigger active» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
11. «Remote start active»
12. «Charge cord unplugged» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»

- VMCU follows Wakeup sequence (Section 2.4.1.6)
- CCPU Power manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE)
- VMCU starts DisplayOnly Timer if «Run/ACC NOT Active» for more than 5 seconds while I-CAN bus is awake– AND- «Illumination Inactive»
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- CCPU notifies applications of wakeup event via Event Manager
- VMCU starts DisplayOnly Timer if timer is not running -AND- «Run/ACC NOT Active» for more than 5 seconds while I-CAN bus is awake –AND- VMCU detects one of the following conditions:
  - «Doors Opened»
  - <CD Eject> via CAN or LIN
  - <CD Insert> via CAN
  - <Run/ACC Event> –AND– «LoadShed Active»
  - «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  - «Door Unlock» via CAN
  - «Predictive trigger active»
  - «Remote start active to inactive»
  - «Charge cord unplugged»
- VMCU starts PaaK Charging Timer if «PaaK Charging Active»

Scenario B: Unpowered

- VMCU and CCPU are Unpowered
- Power is applied
- VMCU follows Power On sequence (Section 2.4.1.1)
- VMCU detects one of the following conditions:
  1. «Doors Opened» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
  2. <CD Insert> via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
  3. <Run/ACC Active> –AND– «LoadShed Active»
  4. «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  5. «Run Active» –AND– «EngineOn Inactive» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  6. «Door Unlock» via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
  7. «Illumination Keep Active»
  8. «PaaK Charging Active»

9. «Predictive trigger active» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
10. «Remote start active»
11. «Charge cord unplugged» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»

- CCPU Power manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE)
- VMCU starts DisplayOnly Timer if «Run/ACC NOT Active while I-CAN bus is awake –AND- «Illumination Inactive»
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- VMCU starts DisplayOnly Timer if timer is not running -AND- «Run/ACC NOT Active» for more than 5 seconds while I-CAN bus is awake –AND- VMCU detects one of the following conditions:
  - «Doors Opened»
  - <CD Eject> via CAN or LIN
  - <CD Insert> via CAN
  - <Run/ACC Event> –AND– «LoadShed Active»
  - «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  - «Door Unlock» via CAN
  - «Predictive trigger active»
  - «Remote start active to inactive»
  - «Charge cord unplugged»
- VMCU starts PaaK Charging Timer if «PaaK Charging Active»

[END]

2.4.2.3 Suspend to VHM [T03]

[P06.R040.06]

SPSS Scenarios:

PWRMAN-FUN-REQ-295414/A-Key OFF Power Moding - ECG and SYNC

Application Scenarios:

ECG requires the SYNC to be powered on
Over The Air update

Transition Sequence:

Scenario A: VMCU is in LPU/Standby

- CCPU is in Unpowered, CAN and LIN networks inactive
- VMCU detects CAN wakeup and wakes up
- VMCU detects the following condition:
  1. «KeyOffPwMde Active» AND «Transport Inactive» AND «LoadShed Inactive» AND «KeyOffMde Critical Battery Inactive» AND «VBattState Normal»"
- VMCU follows Wakeup sequence (Section 2.4.1.6)
- CCPU Power manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <VHM> state (PM_SET_PM_STATE)
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <VHM> state
- CCPU requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- CCPU notifies applications of wakeup event via Event Manager
- CCPU must periodically send PM_ACTIVITY_TIMER to remain in VHM
- If SYNC is up due to KeyOffPwMde_D_Stat = ON, and a load shed event becomes active then Load shed is ignored while KeyOffPwMde_D_Stat = ON and SYNC remains powered up

Scenario B: Unpowered

- VMCU and CCPU are Unpowered
- Power is applied

- VMCU follows Power On sequence (Section 2.4.1.1)
- VMCU detects the following conditions:
    1. «KeyOffPwMde Active» AND «Transport Inactive» AND «LoadShed Inactive» AND «KeyOffMde Critical Battery Inactive» AND «VBattState Normal»"
- CCPU Power manager enters <Wait-On>, establishes IPC link
- VMCU commands CCPU to enter <VHM> state (PM_SET_PM_STATE)
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <VHM> state
- CCPU requests wake source from VMCU (GET_PM_WAKE_SOURCE)
- VMCU acknowledges wake source (GET_PM_WAKE_SOURCE_COMPLETE) with fields WakeSource1 and WakeSource2 and WakeSource3 set to indicate wake source as defined in S23b
- CCPU must periodically send PM_ACTIVITY_TIMER to remain in VHM
- If SYNC is up due to KeyOffPwMde_D_Stat = ON, and a load shed event becomes active then Load shed is ignored while KeyOffPwMde_D_Stat = ON and SYNC remains powered up

[END]

2.4.2.4 Infotainment to Wait-Shutdown [T04]

[P06.R041.04]
Requirement Obsolete, state transition will be through DisplayOnly and then to Wait-Shutdown
[END]

2.4.2.5 Infotainment to DisplayOnly [T05]

[P06.R042.17]
SPSS Scenarios:

Battery voltage out-of-range while in Extended Play, MMActive, or Phone Mode (PWRMAN-GREQ-40633-X)

Application Scenarios:

Transition Sequence:

- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:
    1. «Extended Play timer expires» –AND– «No Phone Activity»–AND– «No PM Activity» -AND- «LBI Interaction Watchdog Timer is not running»
    2. «KeyOut Active» –AND– <HMI On/Off> received via CAN –AND– «No Phone Activity» -AND- «LBI Interaction Watchdog Timer is not running»
    3. «KeyOut Active» –AND– «Phone Timer expires»–AND– «Extended Play timer Expired» -AND- «LBI Interaction Watchdog Timer is not running»
    4. «KeyOut Event» –AND– «No Phone Activity»
    5. «Transport Active» –AND– «ACC Event» –AND– «LoadShed Inactive»
    6. «Transport Active» –AND– «Run Active » –AND– «EngineOn Inactive» –AND– «LoadShed Inactive»
    7. «Transport Active» –AND– «KeyOut Active» –AND– «LoadShed Inactive»
    8. «Extended Play timer expires» –AND– «No PM Activity» –AND– «No Phone Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active») -AND- «LBI Interaction Watchdog Timer is not running»
    9. «KeyOut Event» –AND– «No Phone Activity» –AND– «No PM Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active»)
    10. «Extended Play timer expires» –AND– «PM Activity» –AND– «VMCU Maximum Activity Timer Expired for this ignition cycle» –AND– «No Phone Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active») -AND- «LBI Interaction Watchdog Timer is not running»
    11. «KeyOut Event» –AND– «PM Activity» –AND– «VMCU Maximum Activity Timer Expired for this ignition cycle» –AND– «No Phone Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active»)
    12. «HMI On/Off» received via CAN –AND– «KeyOut Active» –AND– «No Phone Activity» –AND– «No PM Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active»)
    13. «Phone Timer Expires» –AND– «KeyOut Active» –AND– «Extended Play Timer Expired» –AND– «No PM Activity» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active») -AND- «LBI Interaction Watchdog Timer is not running»

14. «HMI On/Off» received via CAN –AND– «KeyOut Active» –AND– «No Phone Activity» –AND– «PM Activity» –AND– «VMCU Maximum Activity Timer Expired for this ignition cycle» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active»)
15. «Phone Timer Expires» –AND– «KeyOut Active» –AND– «Extended Play Timer Expired» –AND– «PM Activity» –AND– «VMCU Maximum Activity Timer Expired for this ignition cycle» -AND- («Illumination Keep Active» -OR- «PaaK Charging Active») -AND- «LBI Interaction Watchdog Timer is not running»
16. «LBI Interaction Watchdog Timer Expires» -AND- -NOT- «Run Active»
    -AND- -NOT-
    [
     (
      («Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive»))
      – OR-
      («ACC Active» –AND– «Transport Inactive»)
)
     –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    ]
    -AND- -NOT- <eCallNotification == Active>
17. Touchscreen HMI On Button present is configured –AND– «KeyOut Event» –AND– «No Phone Activity»

- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE), with “Power Status” field updated with appropriate fields (Voltage Error, Key State, Load Shed)
- If configured for X40 = Pre X40 then CCPU follows Station Management deactivation (SPSS STMGNT-GSTM-121258) and sets “HMIAudioMode == OFF” or “HMIAudioMode == LoadShed” as appropriate. If configured for X40 = X40 then CCPU sends “HMIAudioMode==OFF” to AHU once CCPU has completely stopped Sxedl and X40 chip is powered down.
- For condition 17. only, VMCU starts 3-Minute Clock timer.
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU notifies applications of state change via Event Manager
- VMCU starts DisplayOnly Timer if timer is not running -AND- «KeyOut Active» –AND- VMCU detects one of the following conditions:
  - «Doors Opened»
  - <CD Eject> via CAN or LIN
  - <CD Insert> via CAN
  - «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  - «Predictive trigger active»
  - «Remote start active to inactive»
  - «Charge cord unplugged»

[END]

2.4.2.6 Infotainment to VHM [T06]

[P06.R043.07]
Requirement Obsolete, state transition will be through DisplayOnly and then to VHM
[END]

2.4.2.7 DisplayOnly to Infotainment [T07]

[P06.R044.10]
SPSS Scenarios:

Enter Extended Play Mode (from 10-Minute Clock Mode) (PWRMAN-GUC-40451-X)
Enter MMActive Mode (from 10-Minute Clock Mode) (PWRMAN-GUC-40493-X)

Application Scenarios:

Enter Extended Play Mode from “Welcome” strategy
Enter MMActive Mode from “Welcome” strategy
eCall (911)

Transition Sequence:

- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:

1. «ACC Active» –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
2. «Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive») –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
3. SYNC is configured for Extended Play –AND– <HMI On/Off> received via CAN or LIN –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
4. <eCallNotification == Active>
5. «LBI Password Screen Active»
   -AND- -NOT-
   [
   («Acc Active» -OR-
   («Run Active» -AND- «EngineOn Inactive») -OR-
   «KeyOut Active»
) -AND-
   «Transport Active» -AND-
   «VBattState Normal» -AND-
   «LoadShed Inactive»
   ]

- CCPU sets “HMIAudioMode == ON” (within 650 to 700ms after bus awake)
- VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE)
- VMCU if HMI On/Off button pressed, then start EP Timer
- VMCU starts LBI Interaction Watchdog Timer, if entered via «LBI Password Screen Active»
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <Infotainment> state
- CCPU notifies applications of state change via Event Manager

[END]

2.4.2.8 DisplayOnly to Wait-Shutdown [T08]

[P06.R045.11]

SPSS Scenarios:

Exit Transport Mode

Application Scenarios:

TBD

Transition Sequence:

- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:
  1. «No PM Activity» –AND– DisplayOnly Timer has expired –AND–«Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- («KeyOut Active» AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
  2. «PM Activity» –AND– DisplayOnly Timer has expired –AND–«VMCU Maximum Activity Timer has expired» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- («KeyOut Active» AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
  3. «No PM Activity» –AND– «Lock Event» after <Doors Closed> -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- (((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)
  4. «PM Activity» –AND– «Lock Event» after <Doors Closed> –AND– «VMCU Maximum Activity Timer has expired» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- (((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)
  5. «No PM Activity» –AND–«KeyOut Event» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- («Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
  6. CCPU requests transition to <Suspend> via SET_PM_STATE -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- ((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)

7. «Illumination BatterySave Timer Expires» -AND- ((«PaaK Charging Active» -AND- «PaaK Charging Timer is Expired») –OR- «PaaK Charging Inactive»)
8. «PaaK Charging Timer Expires» -AND- ((«Illumination Active» -AND- «Illumination BatterySave Timer is Expired») –OR- «Illumination Inactive») -NOT- (DisplayOnly Timer is still running) -AND- -NOT- Power Fault Scenario A –AND- -NOT- Power Fault Scenario B -AND- -NOT-
   (
   («Acc Active» -OR-
   («Run Active» -AND- «EngineOn Inactive») -OR-
   «KeyOut Active»
) -AND-
   «Transport Active» -AND-
   «VBattState Normal» -AND-
   «LoadShed Inactive»
)
9. «Illumination Keep Active» transitions to «Illumination Inactive»» -AND- «PaaK Charging Inactive» -AND- -NOT- (DisplayOnly Timer is still running) - AND- -NOT- (3-Minute Clock Timer due to Touchscreen HMI On Button present configuration is still running) -AND-NOT- Power Fault Scenario A –AND- -NOT- Power Fault Scenario B -AND- -NOT- ((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
10. «PaaK Charging Active» transitions to «PaaK Charging Inactive» -AND- «Illumination Keep Inactive» -AND- -NOT- (DisplayOnly Timer is still running) –AND- -NOT- (3-Minute Clock Timer due to Touchscreen HMI On Button present configuration is still running) -AND- NOT- Power Fault Scenario A –AND- -NOT- Power Fault Scenario B -AND- -NOT- ((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
11. «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- ((«KeyOut Active» -AND- «Transport Active» transitions to «Transport Inactive») OR («Transport Active To Inactive» event occurs))
12. «No PM Activity» –AND– 3-Minute Clock Timer due to Touchscreen HMI On Button present configuration has expired » –AND– -NOT- (DisplayOnly Timer is still running) –AND–«Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- («KeyOut Active» AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)
13. «PM Activity» –AND– 3-Minute Clock Timer due to Touchscreen HMI On Button present configuration has expired –AND–«VMCU Maximum Activity Timer has expired» » –AND– -NOT- (DisplayOnly Timer is still running) -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- -NOT- («KeyOut Active» AND «Transport Active» AND «VBattState Normal» –AND– «LoadShed Inactive»)

- VMCU commands CCPU to enter <Suspend> state (PM_SET_PM_STATE)
- VMCU starts 30s CAN Bus Shutdown timer and 120s CCPU Shutdown Timer
- CCPU enters Wait-Shutdown and follows Station Management deactivation (SPSS STMGNT-GSTM-122158) and sets “HMIAudioMode == OFF”
- VMCU cancels 30s CAN Bus Shutdown timer and should leave the CAN buses inactive (i.e. does not request activity and should vote to go to sleep if bus is active)
  - If the 30s CAN Bus Shutdown timer expires before CCPU sets HMIAudioMode == OFF, then the VMCU sets HMIAudioMode == OFF or HMIAudioMode == LoadShed (if «LoadShed Active») and should leave CAN buses inactive (i.e. does not request activity and should vote to go to sleep if bus is active)
- CCPU starts preparing Shutdown (running Shutdown proces and saving the persistency data)
- After all the data is saved, the CCPU will either send SET_PM_STATE(OFF) or SET_PM_STATE(SUSPEND).
- If the CCPU sends SET_PM_STATE(OFF), the following sequence is followed:
  - If there is no response back from the VMCU, the CCPU shall keep trying to send SET_PM_STATE(OFF) every 5s.
  - If the VMCU never sees the CCPU send the SET_PM_STATE(OFF) after the 120s Wait-Shutdown timer has expired, the VMCU shall follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
  - Upon receiving the SET_PM_STATE(OFF) from the CCPU, the VMCU sends SET_PM_STATE_COMPLETE(OFF) to the CCPU
  - CCPU starts sending Shutdown commands to internal modules and processes
  - After Shutdown complete, CCPU acknowledges Suspend request (SET_PM_STATE_COMPLETE) to the VMCU
  - CCPU is ready to be unpowered

- VMCU will request to tear down the link if the link is still up
  - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
- The VMCU will follow the CCPU Shutdown sequence (Section 2.4.1.3).

- If the CCPU sends SET_PM_STATE(SUSPEND), the following sequence is followed:
  - CCPU is ready to enter stop
  - If the CAN signal PwLoApim_T_Actl == 0 -OR- << LoadShed Active >> -OR- <<KeyOffMde Critical Battery Active>> -OR- LifeCycMde_D_Actl != Normal -OR- CCPU Watchdog expired
    - VMCU will send SET_PM_STATE(OFF) to the CCPU.
    - Upon receiving the SET_PM_STATE(OFF) from the VMCU, the CCPU sends SET_PM_STATE_COMPLETE(OFF) to the VMCU when it is ready to be unpowered
    - If the VMCU never sees the CCPU send the SET_PM_STATE_COMPLETE(OFF) after the 120s Wait-Shutdown timer has expired, the VMCU shall follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - CCPU is ready to be unpowered
    - VMCU will request to tear down the link if the link is still up
      - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - The VMCU will follow the CCPU Shutdown sequence (Section 2.4.1.3).
  - If the CAN signal PwLoApim_T_Actl != 0
    - Upon receiving the SET_PM_STATE(SUSPEND) from the CCPU, the VMCU sends SET_PM_STATE_COMPLETE(SUSPEND) to the CCPU
    - The VMCU initialize the Stop State Timer to the value of the PwLoApim_T_Actl signal
    - VMCU will request to tear down the link if the link is still up
      - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - The VMCU will then follow the Stop/Sleep sequence (Section 2.4.1.4)

[END]

2.4.2.9 DisplayOnly to VHM [T09]

[P06.R046.09]
SPSS Scenarios:

Application Scenarios:
Application requests SYNC to remain powered

Transition Sequence:
- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:
  1. «PM Activity» –AND– DisplayOnly Timer has expired –AND–«VBattState Normal» –AND– «VMCU Maximum Activity Timer not expired» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- «Transport Inactive» -AND- -NOT- (((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)
  2. «PM Activity» –AND– «Lock Event» after <Doors Closed> –AND– «VBattState Normal» –AND– «VMCU Maximum Activity Timer not expired» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- «Transport Inactive» -AND- -NOT- (((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)
  3. «PM Activity» –AND– «KeyOut Event» –AND– «VBattState Normal» –AND– «VMCU Maximum Activity Timer not expired» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- «Transport Inactive»-AND- -NOT- (((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)

4. «KeyOffPwMde Active» –AND– «KeyOut Event» –AND– «VBattState Normal» -AND- «Illumination Inactive» -AND- «PaaK Charging Inactive» -AND- «Transport Inactive»- AND- «KeyOffMde Critical Battery Inactive» -AND- -NOT- ((((«Acc Active» OR («Run Active» AND «EngineOn Inactive») OR «KeyOut Active») AND «Transport Active») AND «VBattState Normal» –AND– «LoadShed Inactive»)

- VMCU commands CCPU to enter <VHM> state (PM_SET_PM_STATE)
- CCPU acknowledges request (SET_PM_STATE_COMPLETE) and enters <VHM> state
- CCPU notifies applications of state change via Event Manager
- CCPU must periodically send PM_ACTIVITY_TIMER to remain in VHM

[END]

2.4.2.10 VHM to Infotainment [T10]

[P06.R047.07]
SPSS Scenarios:

Enter MMActive Mode (via Run/ACC) (PWRMAN-GUC-40493-X)
Enter Extended Play Mode (via HMI button press) (PWRMAN-GUC-40451-X)

Application Scenarios:

eCall (911)

Transition Sequence:

- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:
  1. «Acc Active» –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
  2. «Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive») –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
  3. SYNC is configured for Extended Play –AND– <HMI On/Off> received via CAN –AND– «Transport Inactive» –AND– <«VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
  4. <eCallNotification == Active>
  5. «LBI Password Screen Active»
     -AND- -NOT-
     [
     («Acc Active» -OR-
     («Run Active» -AND- «EngineOn Inactive») -OR-
     «KeyOut Active»
) -AND-
     «Transport Active» -AND-
     «VBattState Normal» -AND-
     «LoadShed Inactive»
     ]
- CCPU sets “HMIAudioMode == ON” (within 650 to 700ms after bus awake)
- VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE)
- VMCU starts LBI Interaction Watchdog Timer, if entered via «LBI Password Screen Active»
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <Infotainment> state
- CCPU notifies applications of wakeup event via Event Manager

[END]

2.4.2.11 VHM to DisplayOnly [T11]

[P06.R048.08]
SPSS Scenarios:

Application Scenarios:

SYNC “Welcome” strategy
CD Insert/Eject HMI indication
Transport Mode

Transition Sequence:

- CCPU is running, CAN and LIN networks active
- VMCU detects one of the following condition:
    1. «Doors Opened» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    2. <CD Eject> via CAN or LIN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    3. <CD Insert> via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    4. «Acc Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    5. «Transport Active» –AND– «Run Active » –AND– «EngineOn Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    6. «Illumination Keep Active»
    7. «PaaK Charging Active»
    8. «Predictive trigger active» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    9. «Remote start active»
    10. «Charge cord unplugged» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE)
- VMCU starts DisplayOnly Timer if «KeyOut Active »
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU notifies applications of wakeup event via Event Manager
- VMCU starts DisplayOnly Timer if timer is not running -AND- «KeyOut Active» –AND- VMCU detects one of the following conditions:
    - «Doors Opened»
    - <CD Eject> via CAN or LIN
    - <CD Insert> via CAN
    - «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    - «Predictive trigger active»
    - «Remote start active to inactive»
    - «Charge cord unplugged»

[END]

2.4.2.12 VHM to Wait-Shutdown [T12]

[P06.R049.05]
SPSS Scenarios:

MMInactive to Suspend

Application Scenarios:

TBD

Transition Sequence:

- CCPU is running, CAN networks active
- VMCU detects one of the following conditions:
    1. «KeyOffPwMde inactive» AND «No PM Activity» (CCPU does not request activity via PM_ACTIVITY_TIMER within timeout, i.e., 10 sec)
    2. «KeyOffPwMde inactive» AND «VMCU Maximum Activity Timer» expires.
    3. «No PM Activity» (CCPU does not request activity via PM_ACTIVITY_TIMER within timeout, i.e., 10 sec) AND «KeyOffPwMde Active» AND CAN bus goes to sleep

- VMCU commands CCPU to enter <Suspend> state (PM_SET_PM_STATE)
- VMCU starts 30s CAN Bus Shutdown timer and 120s CCPU Shutdown Timer
- CCPU enters Wait-Shutdown and follows Station Management deactivation (SPSS STMGNT-GSTM-122158) and sets “HMIAudioMode == OFF”
- VMCU cancels 30s CAN Bus Shutdown timer and should leave the CAN buses inactive (i.e. does not request activity and should vote to go to sleep if bus is active)

- If the 30s CAN Bus Shutdown timer expires before CCPU sets HMIAudioMode == OFF, then the VMCU sets HMIAudioMode == OFF or HMIAudioMode == LoadShed (if «LoadShed Active») and should leave CAN buses inactive (i.e. does not request activity and should vote to go to sleep if bus is active)

- CCPU starts preparing Shutdown (running Shutdown process and saving the persistency data)
- After all the data is saved, the CCPU will either send SET_PM_STATE(OFF) or SET_PM_STATE(SUSPEND).
- If the CCPU sends SET_PM_STATE(OFF), the following sequence is followed:
  - If there is no response back from the VMCU, the CCPU shall keep trying to send SET_PM_STATE(OFF) every 5s.
  - If the VMCU never sees the CCPU send the SET_PM_STATE(OFF) after the 120s Wait-Shutdown timer has expired, the VMCU shall follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
  - Upon receiving the SET_PM_STATE(OFF) from the CCPU, the VMCU sends SET_PM_STATE_COMPLETE(OFF) to the CCPU
  - CCPU starts sending Shutdown commands to internal modules and processes
  - After Shutdown complete, CCPU acknowledges Suspend request (SET_PM_STATE_COMPLETE) to the VMCU
  - CCPU is ready to be unpowered
  - VMCU will request to tear down the link if the link is still up
    - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
  - The VMCU will follow the CCPU Shutdown sequence (Section 2.4.1.3).
- If the CCPU sends SET_PM_STATE(SUSPEND), the following sequence is followed:
  - CCPU is ready to enter stop
  - If the CAN signal PwLoApim_T_Actl == 0 -OR- << LoadShed Active >> -OR- <<KeyOffMde Critical Battery Active>> -OR- LifeCycMde_D_Actl != Normal -OR- CCPU Watchdog expired
    - VMCU will send SET_PM_STATE(OFF) to the CCPU.
    - Upon receiving the SET_PM_STATE(OFF) from the VMCU, the CCPU sends SET_PM_STATE_COMPLETE(OFF) to the VMCU when it is ready to be unpowered
    - If the VMCU never sees the CCPU send the SET_PM_STATE_COMPLETE(OFF) after the 120s Wait-Shutdown timer has expired, the VMCU shall follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - CCPU is ready to be unpowered
    - VMCU will request to tear down the link if the link is still up
      - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - The VMCU will follow the CCPU Shutdown sequence (Section 2.4.1.3).
  - If the CAN signal PwLoApim_T_Actl != 0
    - Upon receiving the SET_PM_STATE(SUSPEND) from the CCPU, the VMCU sends SET_PM_STATE_COMPLETE(SUSPEND) to the CCPU
    - The VMCU initialize the Stop State Timer to the value of the PwLoApim_T_Actl signal
    - VMCU will request to tear down the link if the link is still up
      - If the 120s Wait-Shutdown Timer expires at any time during the Wait_Shutdown process, the VMCU will tear down the link if the link is still up and follow the CCPU Rapid Shutdown sequence (Section 2.4.1.3).
    - The VMCU will then follow the Stop/Sleep sequence (Section 2.4.1.4)

[END]

2.4.2.13 Wait-Shutdown to Infotainment [T13]

[P06.R050.07]

SPSS Scenarios:

Enter MMActive Mode (Run/ACC, Suspend interrupted) (PWRMAN-GUC-40493-X)
Enter Extended Play Mode (via HMI button press, Suspend interrupted) (PWRMAN-GUC-40451-X)

Application Scenarios:

eCall (911)

Transition Sequence:

- CCPU is active, CAN networks inactive

- VMCU detects one of the following conditions:
    1. «ACC Active» –AND– «Transport Inactive» –AND– «VBattState Normal» –AND– «Load Shed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    2. «Run Active» -AND- («EngineOn Active» -OR- «Transport Inactive») –AND– «VBattState Normal» –AND– «LoadShed Inactive» –AND– «VehOnSrc_D_Stat OTA inactive»
    3. SYNC is configured for Extended Play –AND– <HMI On/Off> received via CAN or LIN–AND– «Transport Inactive» –AND– «VBattState Normal» –AND– <Load Shed> is not active –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    4. <eCallNotification == Active>
    5. «LBI Password Screen Active»
       -AND- -NOT-
       [
        («Acc Active» -OR-
          («Run Active» -AND- «EngineOn Inactive») -OR-
          «KeyOut Active»
) -AND-
        «Transport Active» -AND-
        «VBattState Normal» -AND-
        «LoadShed Inactive»
       ]
- If the VMCU has sent SET_PM_STATE_COMPLETE(OFF) to the CCPU and is currently in the process of a CCPU Shutdown, the VMCU shall finish the Shutdown process and complete a Wakeup process before continuining
- CCPU sets “HMIAudioMode == ON” (within 650 to 700ms after bus awake)
- VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE)
- VMCU starts Extended Play timer, if entered via <HMI On/Off>
- VMCU starts LBI Interaction Watchdog Timer, if entered via «LBI Password Screen Active»
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <Infotainment> state
- CCPU notifies applications of state change via Event Manager

[END]

2.4.2.14 WaitSuspend to DisplayOnly [T14]

[P06.R051.09]
SPSS Scenarios:

Application Scenarios:

SYNC “Welcome” strategy
CD Insert/Eject HMI indication
Transport Mode

Transition Sequence:

- CCPU is active, CAN networks inactive
- VMCU detects one of the following condition:
    1. «Doors Opened» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    2. <CD Eject> via CAN or LIN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    3. <CD Insert> via CAN –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    4. «Run/ACC Event» –AND–«Load Shed Active»
    5. «Acc Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    6. «Transport Active» –AND– «Run Active» –AND– «EngineOn Inactive» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
    7. «Illumination Keep Active»
    8. «PaaK Charging Active»
    9. «Predictive trigger active» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»
    10. «Remote start active»

11. «Charge cord unplugged» –AND– «VehOnSrc_D_Stat OTA inactive» –AND– «KeyOffMde Critical Battery Inactive»

- If the VMCU has sent SET_PM_STATE_COMPLETE(OFF) to the CCPU and is currently in the process of a CCPU Shutdown, the VMCU shall finish the Shutdown process and complete a Wakeup process before continuining
- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE)
- VMCU starts DisplayOnly Timer if «KeyOut Active» -AND- –NOT- «Illumination Keep Active»
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU notifies applications of state change via Event Manager
- VMCU starts DisplayOnly Timer if timer is not running -AND- «KeyOut Active» –AND- VMCU detects one of the following conditions:
  - «Doors Opened»
  - <CD Eject> via CAN or LIN
  - <CD Insert> via CAN
  - <Run/ACC Event> –AND– «LoadShed Active»
  - «ACC Active» –AND– «Transport Active» –AND– «VBattState Normal» –AND– «LoadShed Inactive»
  - «Predictive trigger active»
  - «Remote start active to inactive»
  - «Charge cord unplugged»

[END]

2.4.2.15 WaitSuspend to VHM [T15]

[P06.R052.03]
SPSS Scenarios:
N/A

Application Scenarios:
Application wakeup
External diagnostic request

Transition Sequence:

- CCPU is active, CAN and LIN networks inactive
- VMCU detects one of the following conditions:
  1. «PM Activity» –AND–«VBattState Normal» –AND– «VMCU Maximum Activity Timer not expired»
  2. CCPU request (via SET_PM_STATE) to enter VHM
  3. «KeyOffPwMde Active» AND «Transport Inactive» AND «LoadShed Inactive» AND «KeyOffMde Critical Battery Inactive» –AND–«VBattState Normal»
- If the VMCU has sent SET_PM_STATE_COMPLETE(OFF) to the CCPU and is currently in the process of a CCPU Shutdown, the VMCU shall finish the Shutdown process and complete a Wakeup process before continuining
- VMCU commands CCPU to enter <VHM> state (PM_SET_PM_STATE)
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <VHM> state
- CCPU notifies applications of state change via Event Manager
- CCPU must periodically send PM_ACTIVITY_TIMER to remain in VHM

[END]

2.4.2.16 Application Requests Immediate Reboot

[P06.R053.01]
SPSS Scenarios:
N/A

Application Scenarios:
Application reboot (e.g., via Dr. Watson service)

Transition Sequence:

- CCPU is running, CAN and LIN networks active
- CCPU Application requests reboot via power manager API
- CCPU Power Manager requests VMCU to enter Suspend Mode (SET_PM_STATE)

- VMCU may reject request (ignored)
- VMCU requests CCPU enter Suspend mode
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) with reboot (Bit 6 of PM_ACTION_FLAG == 1, or Bit 5 AND Bit 6 of PM_ACTION_FLAG == 1 for COLD reboot)

The CCPU must not clear the reboot request flag until after a reboot/reset or the condition is no longer valid, in the case that the VMCU refuses or does not actually reset the CCPU (maybe it transitions it back to another PM state or SUSPEND state). Under these scenarios, the CCPU would then maintain the state of the reboot request flag and perform another request at the end of the next Suspend process.
[END]

[P06.R061.01]
SPSS Scenarios:
N/A

Application Scenarios:
Application reboot (e.g., during expiration of CCPU Reboot timer)

Transition Sequence:
- CCPU is running, CAN and LIN networks may or may not be active (SUSPEND transition)
- CCPU Application requests reboot via power manager API due to CCPU reboot timer expiration
- CCPU Power Manager requests VMCU to Reboot the CCPU (SET_PM_STATE)
- VMCU performs the CCPU reboot
- VMCU commands CCPU back to the SUSPEND state and then to the Stop Mode if the conditions are correct for Stop mode, otherwise it will shutdown the CCPU.

The CCPU will maintain a configurable Reboot Timer to indicate when a reboot from the VMCU is required.
[END]

2.4.2.17 Application Requests Delayed Reboot

[P06.R054.01]
SPSS Scenarios:
N/A

Application Scenarios:
Application delayed reboot

Transition Sequence:
- CCPU is running, CAN and LIN networks active
- CCPU Application requests delayed reboot via power manager API
- CCPU Power Manager stores flag maintaining delayed reboot request
- CCPU Power Manager awaits next VMCU request to transition into Suspend
- CCPU Power Manager acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) with reboot (Bit 6 of PM_ACTION_FLAG == 1)

[END]

2.4.3 Fault Scenarios

2.4.3.1 Power Faults

[P06.R055.16]
SPSS Scenarios:
System Master Power Moding (PWRMAN-GREQ-40633-3-MFD)

Transition Sequence:

Scenario A (power fault in Infotainment):
- Current CCPU Power Mode is Infotainment

- The VMCU detects one of the following scenarios:
  - CAN <Ignition_Status> signal missing for more than 5 seconds and last known Ignition_Status was RUN while I-CAN bus is awake
  - Voltage B+ > 16V for more than Thysterisis (15.5 V for Thysterisis before go back to functional)
  - Voltage B+ < 9.5V for more than Thysterisis (10.0 V for Thysterisis before go back to functional)
  - «LoadShed Event» and «No Priority Phone Activity»
- VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE) and sets “Power Status” field to signal Voltage Error (Bit “V” == 1) or Load Shed (Bit “L” == 1) as appropriate, VMCU disables the MCM if VbattState = Fault . If the detected fault is CAN related (missing message) then the Voltage Error and Load Shed bits shall both be set to 0.CCPU follows Station Management deactivation (SPSS STMGNT-GSTM-121258) and sets “HMIAudioMode == OFF” or “HMIAudioMode == LoadShed” as appropriate
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) and enters <DisplayOnly> state
- CCPU notifies applications of power error event via Event Manager
- If «KeyOut Active» for more than 5 seconds while I-CAN bus is awake, VMCU starts DisplayOnly Timer
- While in DisplayOnly due to the fault condition from Infotainment, the VMCU should still observe the following power mode transitions from DisplayOnly:
  - <eCallNotification == Active> {T07}
  - «KeyOut Event» {T08}
  - <DisplayOnly Timer expires> {T08}
- Once fault has cleared:
  - If «Run/ACC Active», VMCU enables the MCM if VbattState = Normal,VMCU commands CCPU to enter <Infotainment> state (PM_SET_PM_STATE) and clears “Power Status” field (Bit “V” == 0, Bit “L” == 0), CCPU sets ”HMIAudioMode==ON”, then CCPU follows Station Management activation (SPSS STMGNT-GSD-121238)
  - If «Run/ACC NOT Active», VMCU enables the MCM if VbattState = Normal,VMCU commands CCPU to enter <DisplayOnly> state (PM_SET_PM_STATE) with DisplayOnly Timer active and clears “Power Status” field (Bit “V” == 0, Bit “L” == 0); this shall not change the condition or value of the timer (i.e., timer is running and shall not be reset)

Scenario B (power fault in other modes):

- Current CCPU Power Mode is DisplayOnly or VHM
- The VMCU detects one of the following scenarios:
  - CAN <Ignition_Status> signal missing for more than 5 seconds and last known Ignition_Status was RUN while I-CAN bus is awake
  - Voltage B+ > 16V for more than Thysterisis (15.5 V for Thysterisis before go back to functional)
  - Voltage B+ < 9.5V for more than Thysterisis (10.0 V for Thysterisis before go back to functional)
  - «LoadShed Event»
- VMCU commands CCPU (via PM_SET_PM_STATE) *using current power mode* and sets “Power Status” field to signal Voltage Error (Bit “V” == 1) or Load Shed (Bit “L” == 1) as appropriate, VMCU disables the MCM if VbattState = Fault. If the detected fault is CAN related (missing message) then the Voltage Error and Load Shed bits shall both be set to 0.
- CCPU acknowledges VMCU (PM_SET_PM_STATE_COMPLETE) (no actual state change)
- CCPU sets “HMIAudioMode == OFF” or “HMIAudioMode == LoadShed” as appropriate
- If the current power mode is DisplayOnly, the CCPU notifies applications of power error event via Event Manager
- Unless otherwise indicated, VMCU shall treat other mode transitions as valid while battery fault is active
- Once fault has cleared, VMCU commands CCPU (PM_SET_PM_STATE) *using current power mode*,clears “Power Status” field (Bit “V” == 0, Bit “L” == 0) and VMCU enables the MCM if the VbattState = Normal.
- If fault occurred in <DisplayOnly> power mode and DisplayOnly timer was active, the fault shall not change the condition or value of the timer (i.e., timer remains running)

Scenario C (Load Shed while IPC link is not established):
- VMCU detects «LoadShed Event» while IPC link is not active
- VMCU sets “HMIAudioMode == LoadShed”
- If fault clears (i.e., VMCU detects «LoadShed Inactive») while IPC link is not active, VMCU sets “HMIAudioMode == OFF”

Scenario D (Delay_Accy is missing):
- VMCU detects CAN <Delay_Accy> signal missing for more than 5 seconds while I-CAN bus is awake when <Igntion_Status == Run>
- VMCU sets “Delay_Accy = OFF”
- If fault clears (i.e., VMCU detects Delay_Accy is back), VMCU uses the current value in Delay_Accy

Scenario E (Veh_Lock_Status is missing when <Ignition_Status == Run>):
- VMCU detects CAN <Veh_Lock_Status> signal missing for more than 5 seconds while I-CAN bus is awake when <Ignition_Status == Run>
- VMCU sets “Veh_Lock_Status = Lock”
- If fault clears (i.e., VMCU detects Veh_Lock_Status is back), VMCU uses the current value in Veh_Lock_Status

Scenario F (Ignition_Status, Delay_Accy, & Veh_Lock_Status is missing when <Ignition_Status != Run>):
- VMCU detects CAN <Ignition_Status>, <Delay_Accy>, or <Veh_Lock_Status> signal missing for more than 5 seconds while I-CAN bus is awake when <Ignition_Status != Run>
- VMCU maintains the previous signal states of Ignition_Status, Delay_Accy, and Veh_Lock_Status
- If fault clears (i.e., VMCU detects Veh_Lock_Status is back), VMCU uses the current value in Veh_Lock_Status

Scenario G (Load Shed while KeyOffPwMde_D_Stat = ON):
- If SYNC is up due to KeyOffPwMde_D_Stat = ON, and a load shed event becomes active then Load shed is ignored while KeyOffPwMde_D_Stat = ON and SYNC remains powered up

Scenario H (KeyOffMde_D_Actl is missing):
- VMCU detects CAN < KeyOffMde_D_Actl > signal missing for more than 5 seconds while I-CAN bus is awake when <Igntion_Status == Run>
- VMCU sets “KeyOffMde_D_Actl = Normal”
- If fault clears (i.e., VMCU detects KeyOffMde_D_Actl is back), VMCU uses the current value in KeyOffMde_D_Actl
- When Ignition_Status = OFF if there is no KeyOffMde_D_Actl signal on the bus then assume the last known value
- VMCU shall retain the last known value of KeyOffMde_D_Actl signal during VMCU sleep for use on wake-up

Scenario I (Remote_Start_Status is missing):
- VMCU detects CAN < Remote_Start_Status > signal missing for more than 5 seconds while I-CAN bus is awake and Remote_Start_Status == Remote
- VMCU sets “Remote_Start_Status = Null”
- If fault clears (i.e., VMCU detects Remote_Start_Status is back), VMCU uses the current value in Remote_Start_Status

Scenario J (PwLoApim_T_Actl is missing):
- VMCU detects CAN < PwLoApim_T_Actl > signal missing for more than 5 seconds while I-CAN bus is awake when <Igntion_Status == Run>
- VMCU sets Stop Mode Timer to 0 (disables Stop Mode feature)
- If fault clears (i.e., VMCU detects KeyOffMde_D_Actl is back), VMCU uses the current value in PwLoApim_T_Actl

[END]

2.4.3.2 CCPU Response Failure

[P06.R056.06]
SPSS Scenarios:
N/A

If the CCPU fails to respond (via SET_PM_STATE_COMPLETE) to a request by the VMCU to transition to a state other than Suspend, the following behavior shall occur:

- The VMCU shall wait CCPU_PM_STATE_COMPLETE_TIMER to receive the reply (SET_PM_STATE_COMPLETE);
- The VMCU shall retry (request and await reply) up to 5 times;
- If the CCPU does not respond after the last retry, the VMCU shall immediately reboot the CCPU

Name	Timer or Counter	Value	Reset Conditions
CCPU_PM_STATE_COMPLETE_TIMER	Timer	2s	CCPU transitions from non-WAIT_SUSPEND state to WAIT_SUSPEND
CCPU_SUSPEND_FAILURE_TIMER	Timer	120s	VMCU sends SET_PM_STATE(SUSPEND)

[END]

2.4.4 Timers

[P06.R057.07]The VMCU power mode transitions depend on various timers, which are designed to ensure that the VMCU transitions to a lower-power state without waiting indefinitely for the CCPU. The following timers are defined:

Timer	Value	Notes
DisplayOnly Timer	Program specific	If DE02.Vehicle == CX727, then the timer will be set to 5 minutes. Otherwise the timer will be set to 3 minutes.
Extended Play	1 Min when LifeCycMde = Factory otherwise <Provisioned> in all other Life Cyc Modes.	Enforces maximum EP timer (SPSS requirement) Disabled by Run/ACC (in Functional mode)
Activity Timer	10 s	If set, CCPU transitions to VHM instead of Suspend Holds CCPU in Suspend while not elapsed Reset by PM_ACTIVITY_TIMER or transition from non-VHM to VHM state
Phone Timer	10 s	If set, CCPU stays in Infotainment when ignition off or Extended Play elapsed If Priority is set, prevents ending Infotainment during Load Shed event Reset by PM_ACTIVITY_PHONE and expired by PM_ACTIVITY_PHONE_EXPIRED
Illumination BatterySave Timer	<Provisioned>	Enforces maximum illumination time as the vehicle bus goes to sleep. (See Illumination SPSS)
PaaK Charging Timer	1 hour	Enforces maximum USB charging time to protect from Battery drainage. The timer is restarted every time «PaaK Charging Inactive» transitions to «PaaK Charging Active» unless the timer has previously expired. (See Power Management SPSS)
LBI Interaction Watchdog Timer	30 seconds	Enforces the maximum time after an interaction before CCPU shutdown.

		Reset by PM_ACTIVITY_LBI («LBI Interaction Indication»). (See LBI SPSS)
Stop State Timer	<Provisioned>	Provides the maxium amount of time the VMCU/CCPU can be in STOP mode before full shutdown. This strategy is determined by the configuration. If set to SYNC Strategy, a 2 hour timer shall be used and reset upon every LPU/Standby/Unpowered to Infotainment transition. The timer will decrement while in stop mode and throughout any wakeups to any power state that is not Infotainment. If set to BCM Strategy, the timer shall use the time provided by the CAN signal PwLoApim_T_Actl. The timer shall be reset by the value of the CAN signal PwLoApim_T_Actl whenever it is received or set to 0 in fault conditions defined in this spec. If set to Off, the timer shall always be set to 0 and the Stop Mode feature disabled.
CCPU Reboot Timer	<Provisioned>	Provides the accumulated amount of time of the CCPU in a run state before it must request a reboot from the VMCU.

[END]

2.5 Design Dependencies

None.

2.6 Security Requirements

TODO Special security requirements are contained or referenced within this section. Security features may be further refined in other sections, but this section should either fully specify or at least enumerate/link to security requirements. The rationale for this is so that a consolidated list of security requirements can be easily extracted.

3 System Requirements

3.1 Power State Requirements

[P06.R001.04]Current Consumption: The SYNC module shall not exceed TBD mA current consumption in LPU/Standby mode. [END]

[P06.R003.01]VMCU is Power Master: The VMCU shall command all power mode transitions.[END]

[P06.R004.01]CCPU is Power Slave: The CCPU shall obey all power mode transitions as commanded by the VMCU.[END]

Whenever the IPC link is re-established the VMCU must request the CCPU to transition to the appropriate power state (the power state of the CCPU after any link failure is not deterministic for the VMCU software); the CCPU is not required to explicitly inform the VMCU of the current power state (via an unsolicited SET_PM_STATE_COMPLETE message), except in the case of the initial link.

3.2 Certification Requirements

Not Applicable.

3.3 Performance Requirements

[P06.R005.01]SPSS Performance Requirements: The power manager will meet the following SPSS timing requirements:

PWRMAN-GREQ-40701
PWRMAN-GREQ-40702
PWRMAN-GREQ-40703
PWRMAN-GREQ-40704
PWRMAN-GREQ-40705
PWRMAN-GREQ-40706
PWRMAN-GREQ-40707
[END]

3.4 HMI Requirements

HMI Requirements will be described in A22G – “Welcoming/Program Screenflows”.

3.5 Settings

3.5.1 Configurable Settings

3.5.2 Region/Market Requirements

Not Applicable.

3.5.3 Provisioning Requirements

[P06.R007.01]The SYNC module will provide a provisionable settings to enable/disable Extended Play Mode.[END]

[P06.R008.01]The SYNC module will provide a provisionable settings to set the length of Extended Play Mode, in minutes.[END]

[P06.R009.01] The SYNC module will provide a provisionable settings to configure the VHM state timeout value. [END]

3.5.4 Driving Restrictions

Not Applicable.

3.6 API Requirements

The SYNC power manager must interact with applications to inform applications of changes in power state and the source of a wakeup, as well as provide the ability for applications to request changes in power state.

3.6.1 Main API Scenarios

3.6.2 Developer Interaction Goals

In general, applications will act upon changes in powers state (as notified by Event manager). Few applications will require the ability to request changes in the power state, either directly or indirectly (via the APIs described above).

3.6.3 Developer Documentation

[P06.R016.01]Documentation describing all power manager APIs will be provided.[END]

3.7 Testing Requirements

The testing of the power management will be based on the hardware available at the time. Power Manager testing will be accomplished by verifying the IPC messages being sent between the VMCU and CCPU and by examining the software-based state transitions using standard Operating System debug methods (debug messages, breakpoints, etc.)

3.8 Q&A and Big Decisions

Appendix A: Some Kind of (Optional) Appendix