QUALCOMM®
CDMA Technologies

PM8921™ Software Driver Overview

80-N2283-20 C

Qualcomm Confidential and Proprietary

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

Revision History

Version	Date	Description
A	Mar 2011	Initial release
B	May 2011	Numerous changes were made to this document. It should be read in its entirety.
C	Mar 2012	Updated Interrupt operation (pertaining to modem Hexagon® PMIC driver), overview, VREG interfaces, 19.2 MHz – XO and RC oscillator; added Switch Mode Battery Charging section

Contents

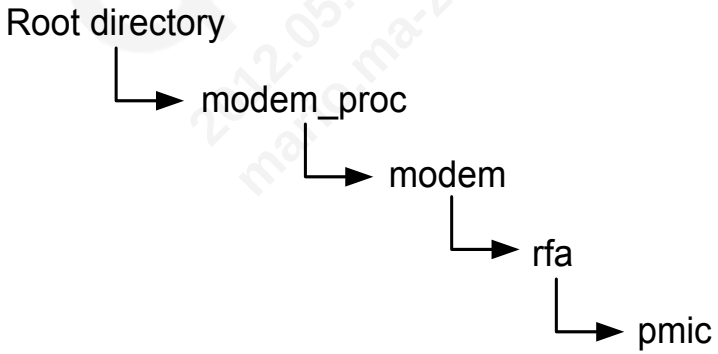
Introduction

Objectives

- At the end of this presentation, you will understand:
  - System clocks
  - Voltage regulator
  - Interrupt Manager (IRQ)
  - Multipurpose pins
  - GPIO
  - Analog multiplexer
  - External interfaces
  - Hard reset feature
  - Switch Mode Battery Charger
  - UART multiplexing

Scope

- This presentation describes PMIC8921 features, functions, API and application examples
- This document is applicable to:
  - Chipset – Qualcomm MSM8960 ASIC
- PMIC software for modem (Hexagon) is at:

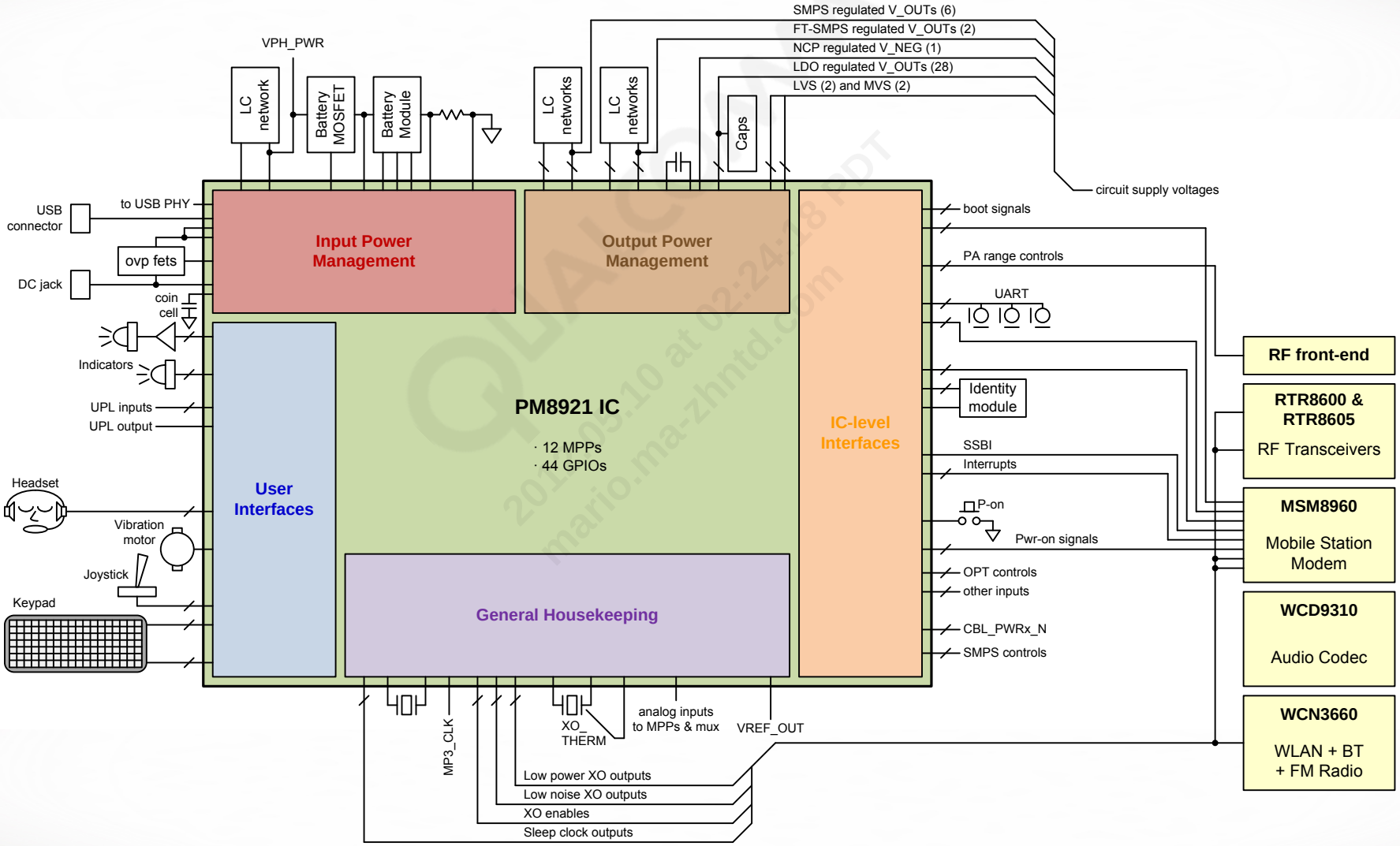


PMIC Features Comparison

Features	PM8058	PM8921
# LDOs Power Switches	26 2	28 (9)
Battery Management Services (BMS: Fuel Gauge)	No	Yes
Switch Mode Charger	No	Yes
RUIM/SIM Level Translator	Yes	Yes
USB OTG Switch	No	Yes
OVP	28 V	30V
IRQ Manager	Secure	Secure
32 kHz Buffer	3	3
XO buffer	4	5
Sleep Clock Generator	32 KHz Xtal XO/Div	32 KHz Xtal Cal RC
Autonomous charger	Yes	Yes
Trickle charger	Yes	Yes
Pre charge indicator	Yes (LED 5 mA)	Yes
CCCV charger	Yes	Yes
CV charger	Yes	Yes
Pulse charger	Yes	No

Features	PM8058	PM8921
HKADC	15 bits	15 bits
19.2 MHz XO	Yes	Yes
Vibrator Driver	1	1
General purpose LED driver	3 (40 mA) 3 (300 mA)	3 (40 mA) 1 (300 mA)
LPG (Light Pulse Generator)	8 channels	8 channels
RF PA controller	6	No
GPIO	40	44
MPP	12	12
8x18 Keypad scanner	Yes	Yes
MPM support	Yes	Yes
# LDOs	18	28
Buck regulator	5	6, 1.5A each 2 (FT) 2.0 A each
Boost	No	No
Camera Flash	Torch mode	Torch mode
OTHC	3	3
Size (mm)	7 x 7	7.8 x 7.8

PM8921 Functional Block Diagram



System Clocks

Overview

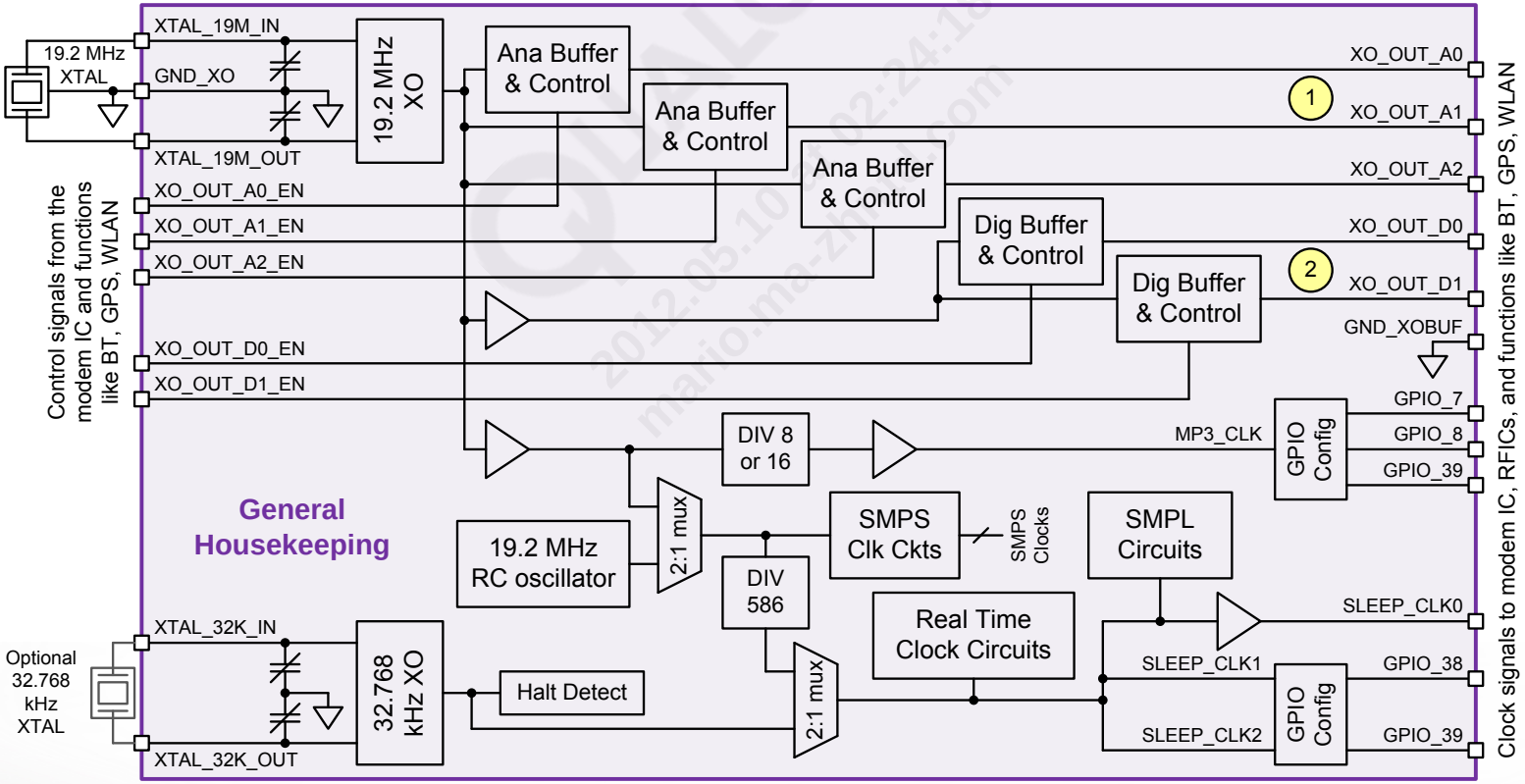
- PM8921 includes several clock circuits with outputs used for general housekeeping functions and elsewhere within the handset system.
  - 19.2 MHz crystal oscillator (XO) circuit
  - Relaxation Crystal (RC) oscillator (internal 19.2 MHz)
  - 32.768 kHz crystal oscillator (XTAL)
  - Divided and buffered clock for MP3 support
  - One dedicated sleep clock output; two additional sleep clock output buffers from GPIO
  - Switch Mode Power Supply (SMPS) clocks
- An example XO distribution scheme based on the MSM8x60 chipset is shown in the following figure.

System Clocks Block Diagram

All clock outputs are now rail to rail outputs – the Ax outputs are low noise where as the Dx outputs are low power outputs

1) Analog buffers: PWR = VREG_L13;

2) Digital buffers: PWR = VREG_L4;



19.2 MHz – XO and RC Oscillator

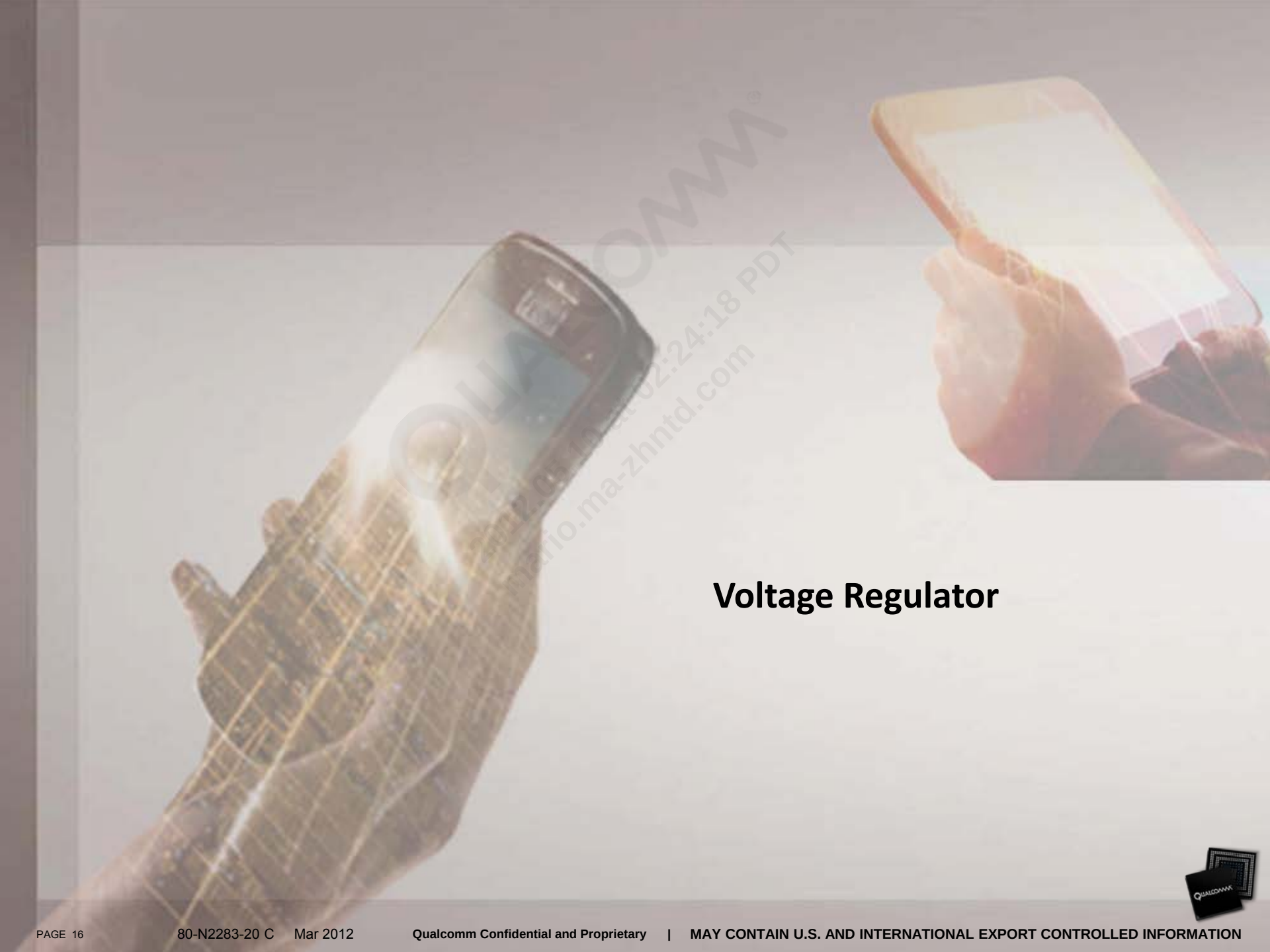
- 19.2 MHz crystal oscillator (XO) circuit
  - The XO signal is the handset's primary timing and frequency reference.
- RC oscillator (19.2 MHz)
  - The RC oscillator is the default clock source during PMIC power-up. It is used until the MSM IC clears interrupts that allow switchovers to the 32.768 kHz crystal oscillator and the 19.2 MHz XO source. Transitions between the clock sources are synchronized to ensure valid pulses at the SLEEP_CLK output(s) and the SMPS input (wide pulse widths, no glitches). After the switchovers are made, the internal RC oscillator is powered down to reduce power consumption.

XO Buffers

- PM8921has five XO buffers
  - Two digital buffers
  - Three analog buffers
- Buffers can be configured in either Manual or Automatic mode
- In Manual mode, buffer output and supply voltage can be turned ON and OFF
- In Automatic mode, buffer output and supply voltage are controlled by the external enable pin

Buffered Sleep Clock Outputs and SMPS Clocks

- Buffered sleep clock outputs
  - Three SLEEP_CLK outputs are derived from one of three sources
    - 32.768 kHz external crystal source (high accuracy)
    - 19.2 MHz XO divided by 586 (32.7645 kHz nominal); this can be used only if RTC functions and SMPL features are not supported
    - 19.2 MHz RC oscillator divided by 586
- SMPS clocks
  - Switched mode power supplies are driven by one of two clock sources
    - 19.2 MHz XO source (external)
    - 19.2 MHz RC oscillator (internal)
  - To Select Clock Source and Dividers to configure switching frequency



Voltage Regulator

Overview

- The output voltage regulation circuits generate all regulated voltages needed for most low-cost wireless handset applications (and many others). Independent regulated power sources are required for various electronic functions to *avoid signal corruption between diverse circuits*, support power sequencing, and meet different voltage-level requirements. There are *44 voltage regulators* provided, all programmable, all derived from a common band gap reference.
  - 6 HF-SMPS (High Frequency SMPS) or buck converters
  - 2 FT-SMPS (Fast Transient SMPS)
  - 28 LDO regulators
  - 1 Negative Voltage Charge Pump (NCP) rated for -1.8 V/200 mA
  - 7 low voltage switches (LVS)
  - 2 medium voltage switches (MVS)

VREG Interfaces

- Controlling voltage regulators
  - Voltage regulators can be
    - Turned on or off (enabled/disabled)
    - Set to a specific voltage level
    - Operating modes, i.e., HPM/LPM that optimize their operation for a high/low current loading condition
  - API examples from PMIC driver in SBL3:

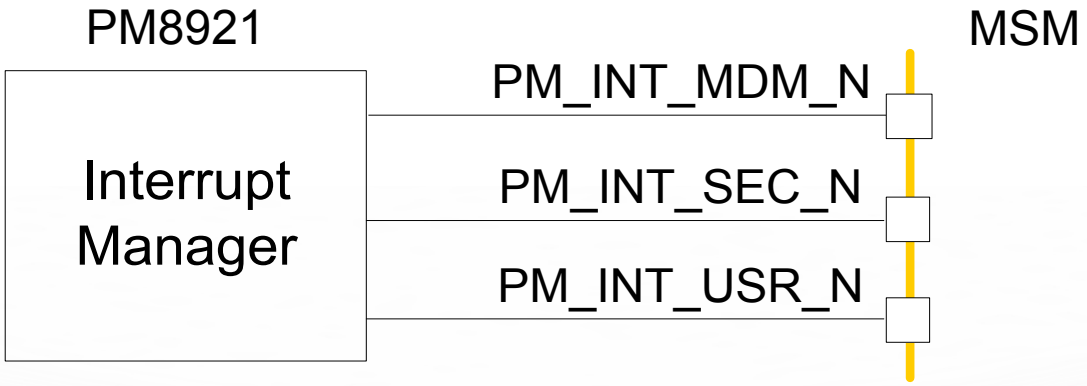
```
pm_vreg_set_level() /*Configures voltage level*/
pm_lp_mode_control() /*Enables/disables LPM for LDOs*/
pm_vreg_smps_config()  /*HPM/LPM etc. for SMPSs*/
```

  - Modem PMIC driver uses NPA and PAM based implementation
    - NPA: Node Power Architecture, a resource management framework in BSP layer
    - PAM: PMIC Arbitration Matrix (maps client operating modes into PMIC rail configs)
    - All rail configuration requests are forwarded to RPM, which consolidates from other processors on SoC as well.
  - HLOS implementation is HLOS dependent.

Interrupt Manager (IRQ)

Overview

- PMIC generates *several interrupts to MSM SoC* for events like a valid charger is present, die temperature is too hot, battery voltage is low, etc.
- All PMIC interrupts to MSM are tied to just three physical interrupt lines from PMIC (called domains)
- Each interrupt will
  - Always assert PM_INT_SEC_N
  - Assert PM_INT_MDM_N if PMIC was configured to assert that line for that interrupt
  - Assert PM_INT_USR_N if PMIC was configured to assert that line for that interrupt



Overview (cont.)

- Typical expected interrupt routing for the **three interrupt domains** is:
  - PM_INT_MDM_ N – Standard interrupt to the modem processor
  - PM_INT_SEC_N – Secure interrupt to the secure root of trust
  - PM_INT_USR_N – User interrupt to the applications processor

Overview (cont.)

- Each interrupt/event has the following *associated SBI* bits:
  - Interrupt mask (read/write)
    - When set, this bit allows the modem device to ignore the interrupting event.
    - The latched status is hidden and PM_INT_MDM_N is not asserted.
  - Domain (read/write)
  - Interrupt real-time status (read-only)
    - This bit follows the real-time interrupt event status (active or inactive) for standard configuration interrupts (highlighted in blue in the figure).
  - Interrupt latched status (read-only)
    - This bit is set when the interrupt event is active and the interrupt mask bit is cleared. It stays set until the interrupt clear bit is set.
    - This bit can only be read while the interrupt mask bit is cleared.
  - Interrupt clear (read/write)
    - Setting this bit clears the interrupt event's latched status.
    - It is cleared automatically after the latched status is read, which has no effect other than allowing it to be set later to again clear the event's latched status.

Interrupt Operation (Pertaining Modem Hexagon PMIC Driver)

- PMIC interrupts modem through PM_INT_MDM_N – Routed to an MSM GPIO that would be configured as an interrupt by the modem software)
- All interrupts are masked by default
- APIs used for ISR configuration
  - To register an ISR

```
pm_set_irq_handle ( pm_irq_hdl_type irq_id , /*Which IRQ*/
                    pm_isr_ptr_type handler); /*ISR*/
```

  - Interrupt can be configured using

```
pm_config_irq ( pm_irq_hdl_type irq_id, /* Which IRQ */
                pm_irq_trigger_type irq_trigger, /* The trigger type for the IRQ. */
                pm_sec_irq_domain_type irq_domain, /* interrupt domain SEC/USR/MDM*/
                boolean irq_permission); /* The permission of the IRQ */
```

Interrupt Operation (cont.)

- APIs used for ISR configuration (cont.)
  - Interrupt can be cleared using

```
pm_clear_irq ( pm_irq_hdl_type irq_id ); /*Which IRQ*/
```

  - The RT status of an interrupt can be read using

```
pm_get_rt_status ( pm_irq_hdl_type rt_status_id , /*Which IRQ*/
                   boolean *rt_status );
```

  - The latched status of an interrupt can be read using

```
pm_get_irq_status( pm_irq_hdl_type irq_id ,  /*Which IRQ*/
                   boolean *status);
```

Power-On Latched Interrupts

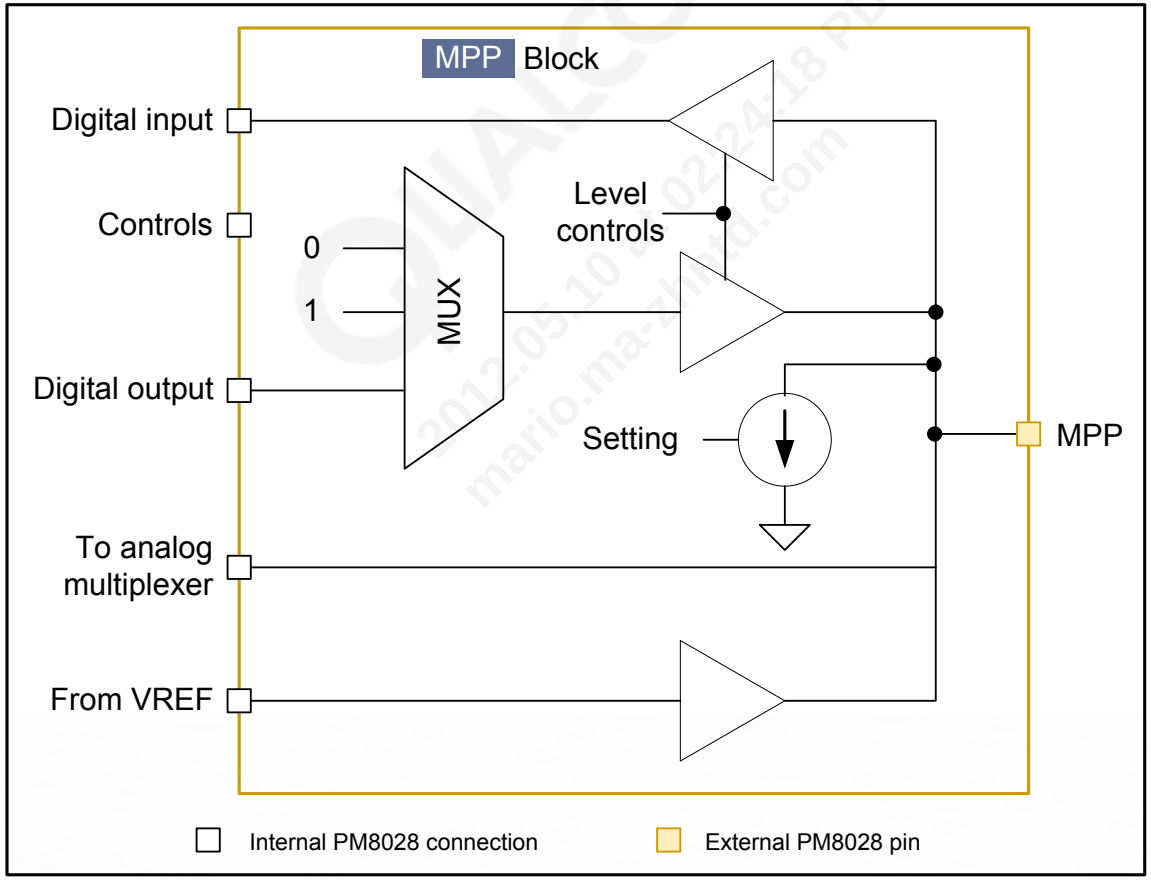
- The phone can be triggered to power on when one or more of the following events occur:
  - Cable power-on
  - RTC alarm
  - SMPL
  - Keypad power-on key
  - Charger
  - Hard Reset
- Related API
  - This API can be used determine the reason for the auto power-on event

```
pm_err_flag_type pm_get_power_on_status(uint32* pwr_on_status);
```

Multipurpose Pins

Overview

- There are 12 Multipurpose Pins (MPP) available.
- MPPs in the PMIC have a similar operation as the GPIO pins on MSM.



Common uses

- Custom features
  - CBLPWRON (two dedicated, MPP11 and MPP12)
  - UIM-level shifter
- Generic features
  - General ADC inputs
  - Battery measurement
  - XO temperature
  - PA temperature
  - Light measurement
  - Joystick position measurement
  - GSM PA voltage reference

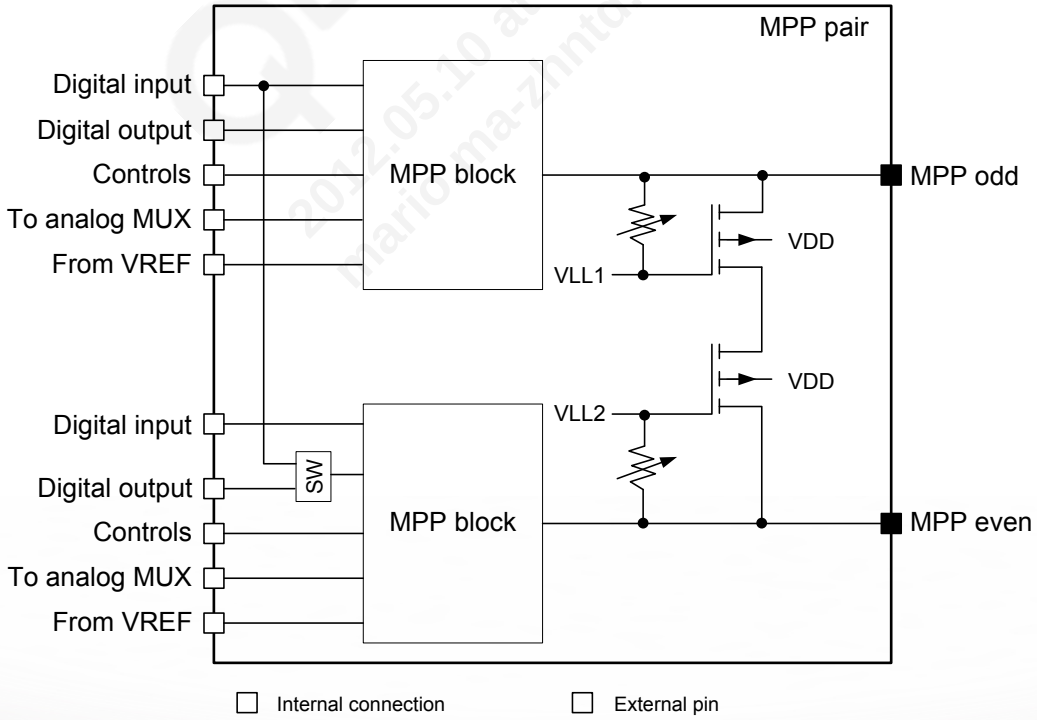
Configuration Options

- Each MPP can be SBI-programmed to any digital input, digital output, bidirectional I/O, analog input, analog output, and current sync configuration.

```
pm_mpp_config_digital_input() /*Interrupt */
pm_mpp_config_digital_output() /*Used as a switch-level translator */
pm_mpp_config_digital_inout() /*RUIM-level translator */
pm_mpp_config_analog_input() /*Routed to AMUX as an input to HKADC */
pm_mpp_config_analog_output() /*Output reference voltage */
pm_mpp_config_i_sink() /*Programmable current sink (LEDs) */
```

MPP Pairs

- Each MPP can be used independently or paired with its complement and used as a level or a bidirectional translator.
- In a level translator application, either side (MPP odd or MPP even) can be configured as the input or output. Each side is pulled up to a selectable power supply, and the two are connected on the internal side of the blocks by using a switch at the even MPP block.

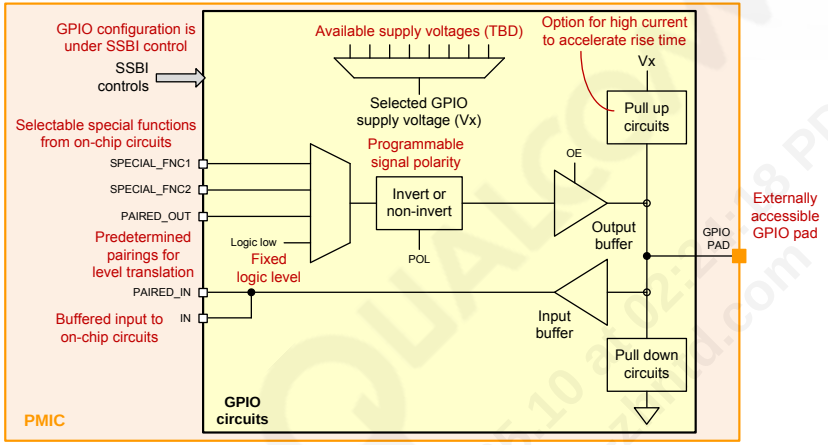


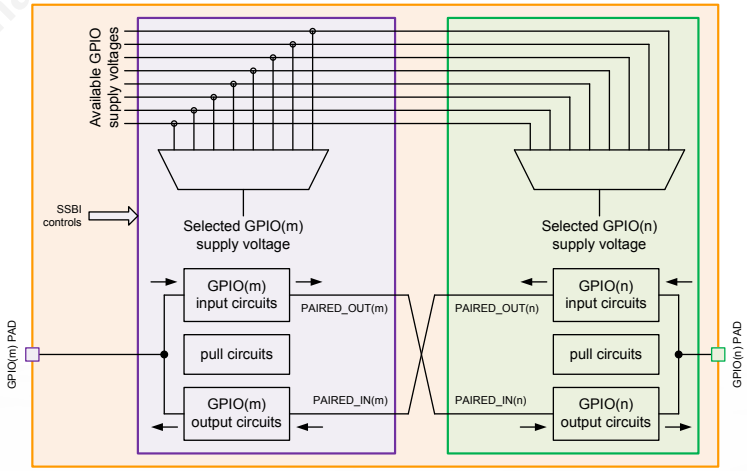
GPIO

Overview

- There are 44 GPIOs
- GPIO pins in PMIC have a similar functionality as the MPPs; differences between the two are:
  - No analog circuit; PMIC GPIO cannot handle analog signal
  - PMIC GPIO block is at least 10x faster than similar MPP block
- General applications
  - Keypad
  - UART level translation

GPIO – Diagram



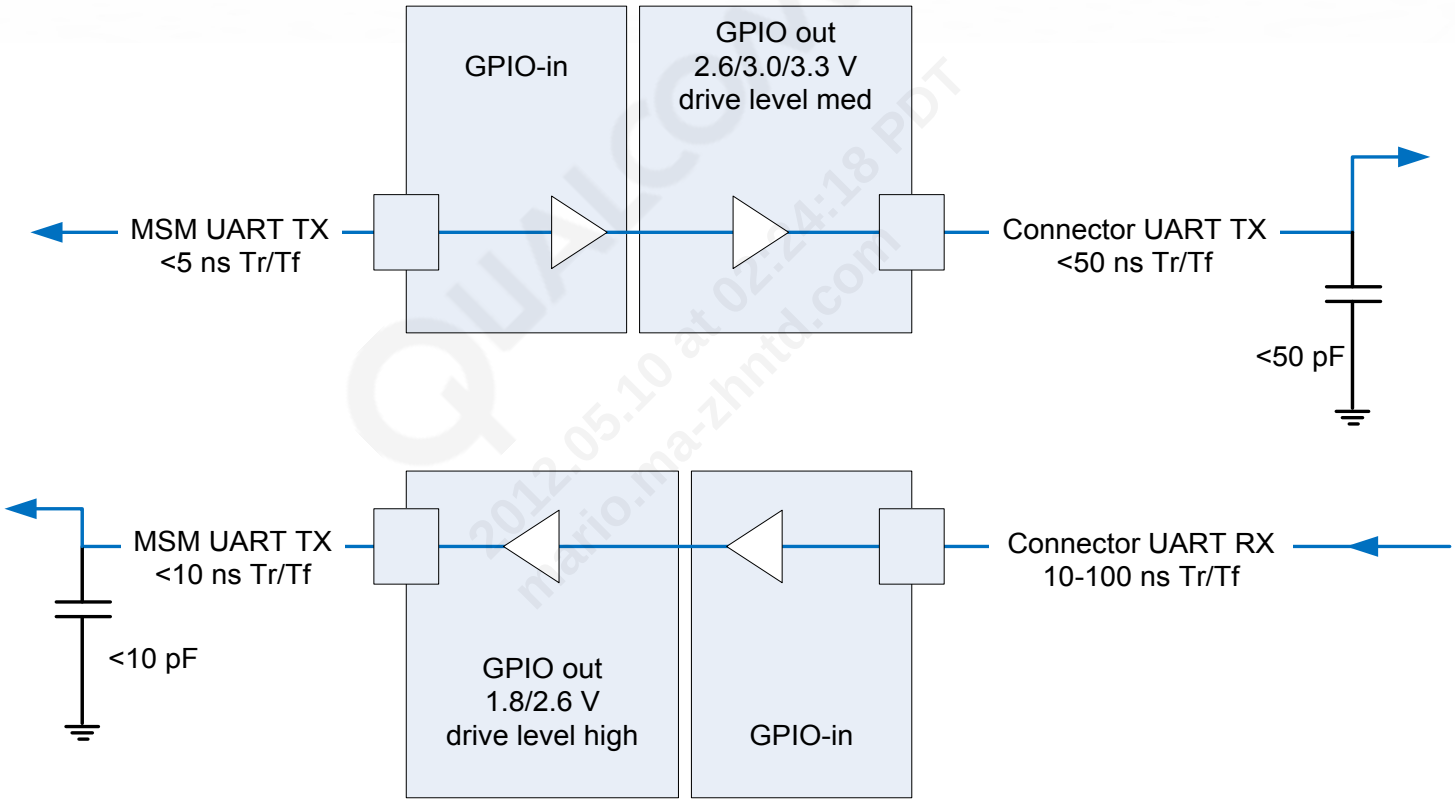


GPIO-Related APIs

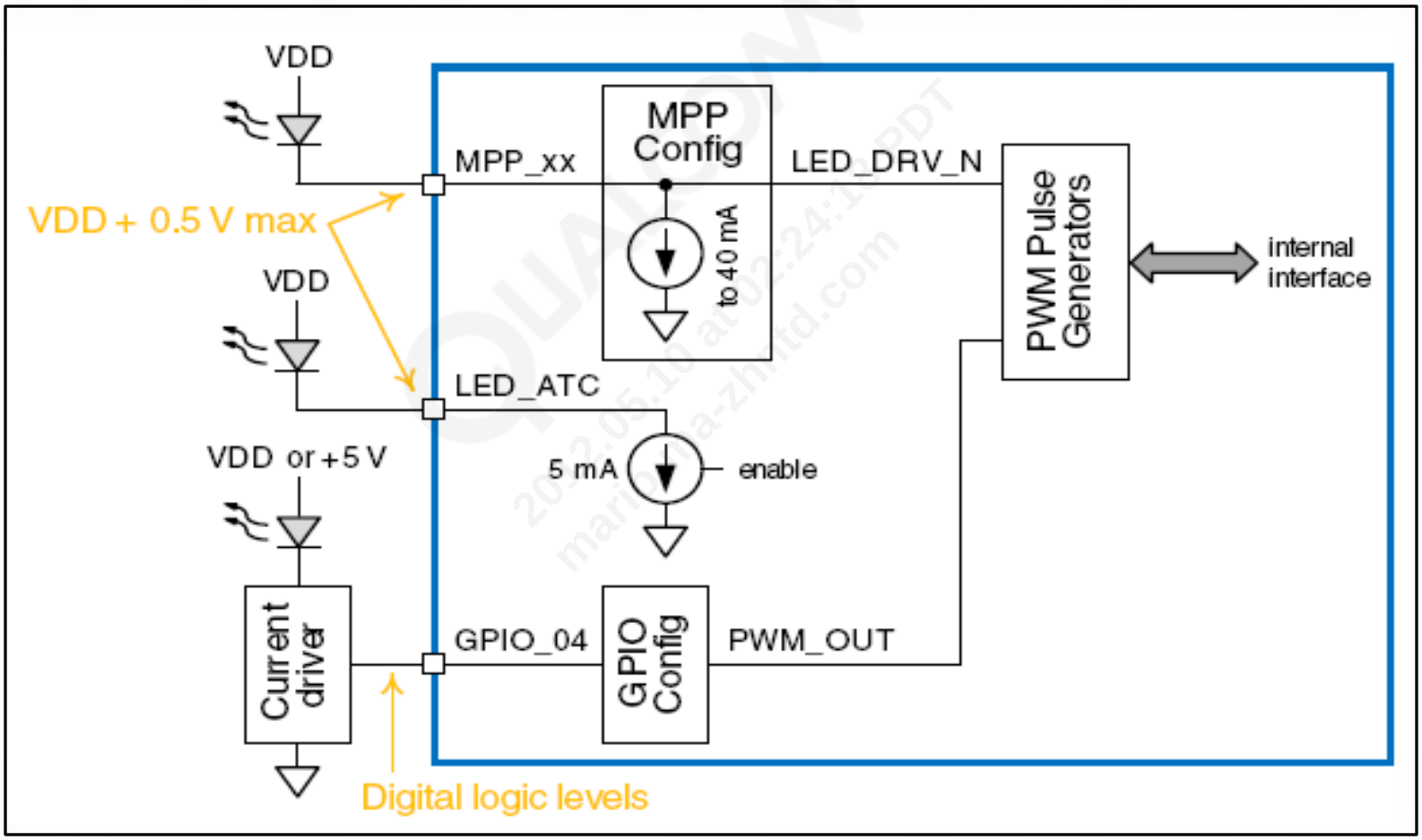
- GPIO APIs

```
pm_gpio_config_digital_input() /*GPIO as digital input*/
pm_gpio_config_digital_output() /*GPIO as digital output*/
pm_gpio_set_ext_pin_config() /*Puts EXT_PIN at high Z state and disables the
block */
```

General Application – UART-Level Translation

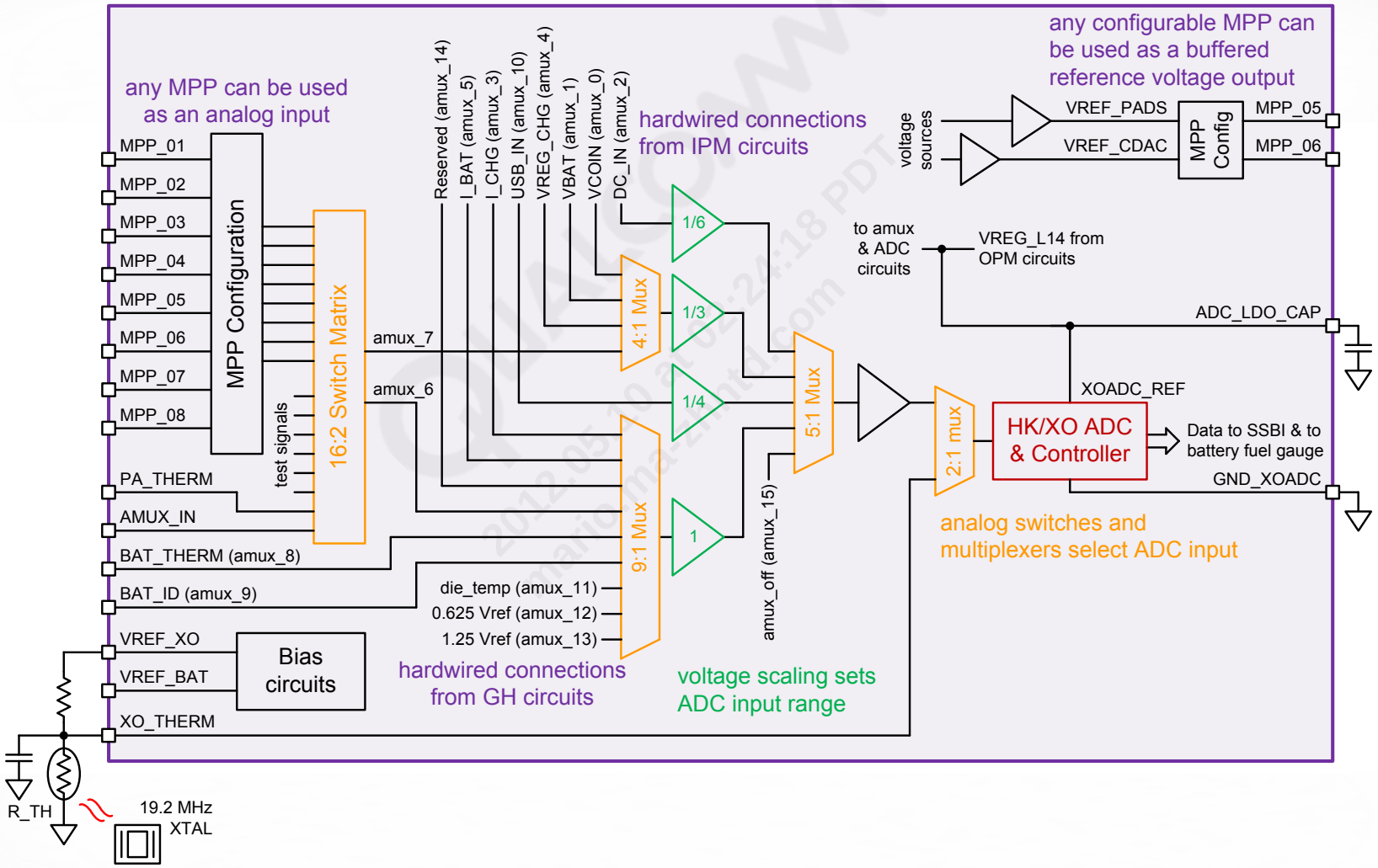


GPIO Application Example – PWM Current Drivers



Analog Multiplexer

Analog Multiplexer



Overview

- General housekeeping functions include analog multiplexing with voltage scaling, the HK/XO ADC, system clock circuits, a real-time clock, and over-temperature protection.
- The PMIC includes a 16-to-1 Analog Multiplexer (AMUX) that selects a single analog signal for routing to XOADC module for analog-to-digital conversion. The 16 multiplexer inputs are derived from five internal connections, six hardwired external connections, and MPPs that can be configured as multiplexer inputs.
- The IC MPPs are configured as analog inputs that are routed through switch circuits to create five multiplexer inputs, which are available to monitor system parameters, such as temperature and battery ID.
- The output of the AMUX is offset-adjusted, and then the new analog voltage is scaled. Both the offset and gain are SBI-programmable.
- The XOADC is a 15 bit type of analog-to-digital converter.

Application Example

- To read the battery ID and detect if it is connected to MPP1, the following setup is required:

```
/*Configure MPP1 as an analog input and assign it to one of
 the analog MUX channels (5 to 9)*/
  pm_mpp_config_analog_input(PM_MPP__1, PM_MPP__AIN__CH_AMUX5);
  pm_xoadc_init();
  pm_xoadc_reset_filter();

/*Selects the input, PMIC AMUX channels, for XO ADC
 conversion. */
  pm_xoadc_set_input(PM_XOADC_INPUT_PMIC);

/*Sets the XOADC decimation ratio and the XOADC clock rate to
 give the sampling rate. */
  pm_xoadc_set_decimation_ratio(PM_XOADC_DECIMATION_RATIO_32K)
  pm_xoadc_set_conversion_rate(PM_XOADC_CONVERSION_RATE_
  TCXO_DIV_4);
```

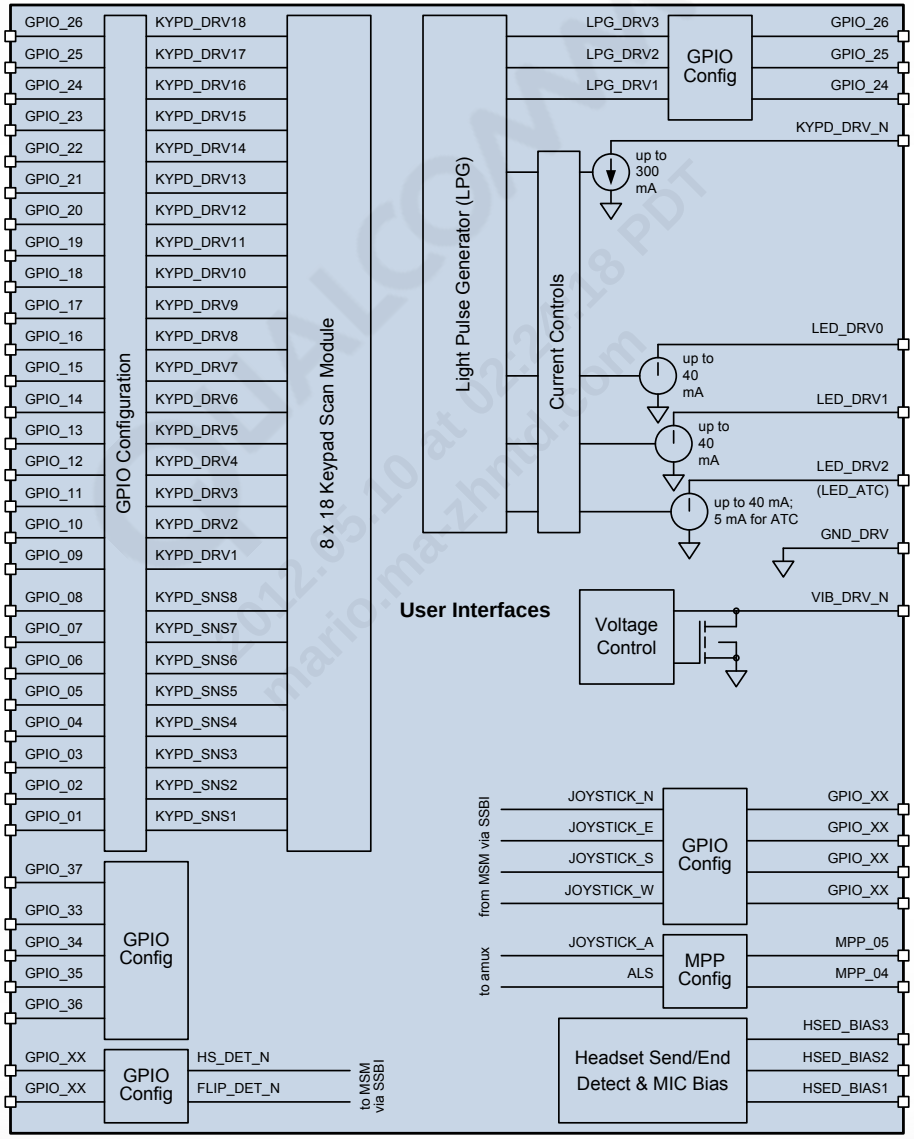
Application Example (cont.)

- To read the battery ID and detect if it is connected to MPP1, the following setup is required: (cont.)

```
/*Sets the XOADC decimation filter configuration */
  pm_xoadc_set_filter_config(PM_XOADC_FILTER_ENABLE_SINC1_
  ENABLE_SINC2);
/*Starts the XOADC conversion. */
  pm_xoadc_enable_modulator(PM_XOADC_ENABLE_CMD);
/*Reads the PMIC XOADC data (24 bits). Only 16 bits are used.
 1 MSM sign bit + 15 data bits. 2's complement form*/
  pm_xoadc_read_data(&test_read);
/*Stops the XOADC conversion. */
  pm_xoadc_enable_modulator(PM_XOADC_DISABLE_CMD);
```

External Interfaces

External Interface



Overview

- A variety of user interface circuits extend the PMIC capabilities and reduce the external support circuits required in typical handset applications. The user interfaces and supporting circuits include:
  - Light pulse generator (LPG)
  - Current driver and related outputs
  - LPG controls for external current drivers (GPIOs)
  - Fixed 5 mA Auto Trickle Charging (ATC) indicator
  - Programmable current sinks for driving LEDs
  - Programmable high-voltage Flash and backlight drivers
  - Vibration motor driver for silent alarms
  - Three headset send/end detection and mic bias pins
  - Joystick support (GPIOs and MPP)
  - Keypad interface (GPIOs)

LCD/Keypad and Flash LED drivers

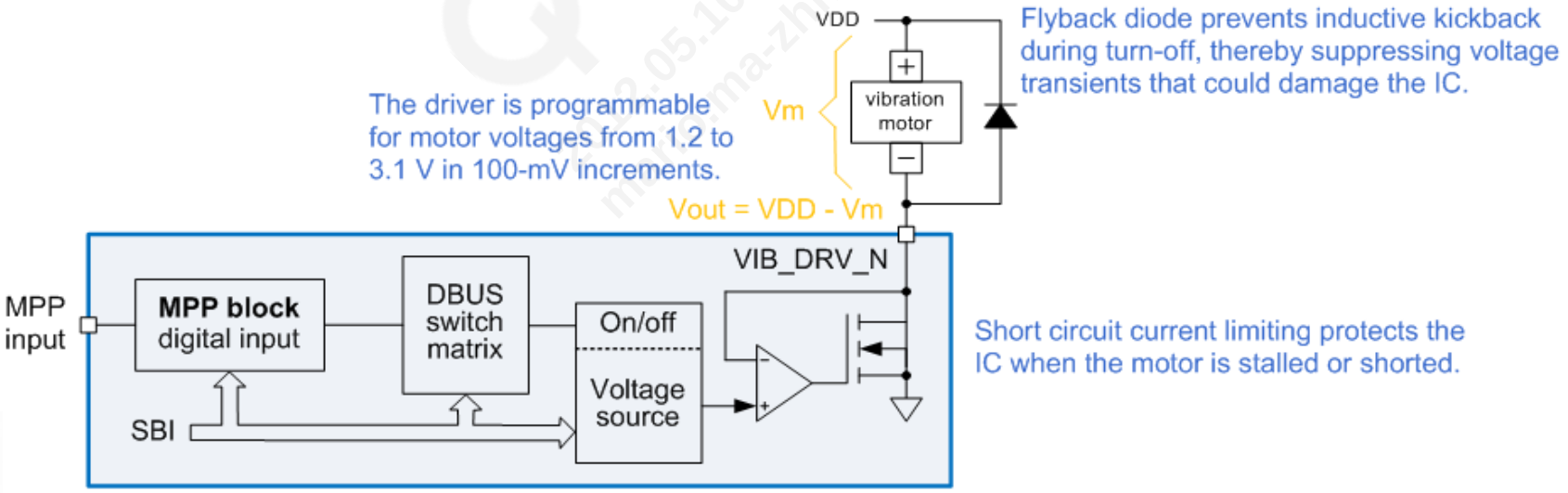
- LCD/keypad

```
/*Turns on or off and sets to different levels of intensity */
```

  - `pm_set_led_intensity()`

Vibrator Motor Driver

- Vibrator motor driver is a low-side voltage regulator that drives a 1.3 V to 3 V motor; can be turned on or off, and set to different modes, e.g., Manual or Automatic
  - pm_vib_mot_set_volt()
  - pm_vib_mot_set_mode()
  - pm_vib_mot_set_polarity()



Vibrator Application Example

- Alerting the handset user of incoming calls is supported with automatic vibration. The vibrator will turn on (3 V) if MPP5 is high and turn off if MPP5 is low.

```
/*Disables the vibrator Motor */
  err = pm_vib_mot_set_volt(0);
/*Configures MPP5 to be a digital input, sets the voltage
 reference to MSMP VREG, and drives data bus 2 (DBUS2). */
  err |= pm_mpp_config_digital_input(PM_MPP_5,
    PM_MPP__DLOGIC__LVL_MSMP, PM_MPP__DLOGIC_IN__DBUS2);
/*Configures vibrator motor to automatic and uses DBUS2 as
 the control line. */
  err |= pm_vib_mot_set_mode(PM_VIB_MOT_MODE__DBUS2);
/*Configures DBUS polarity to active high. */
  err |= pm_vib_mot_set_polarity(PM_VIB_MOT_POL__ACTIVE_HIGH);
/*Configures the vibrator motor to 3 V. */
  err |= pm_vib_mot_set_volt(3000);
```

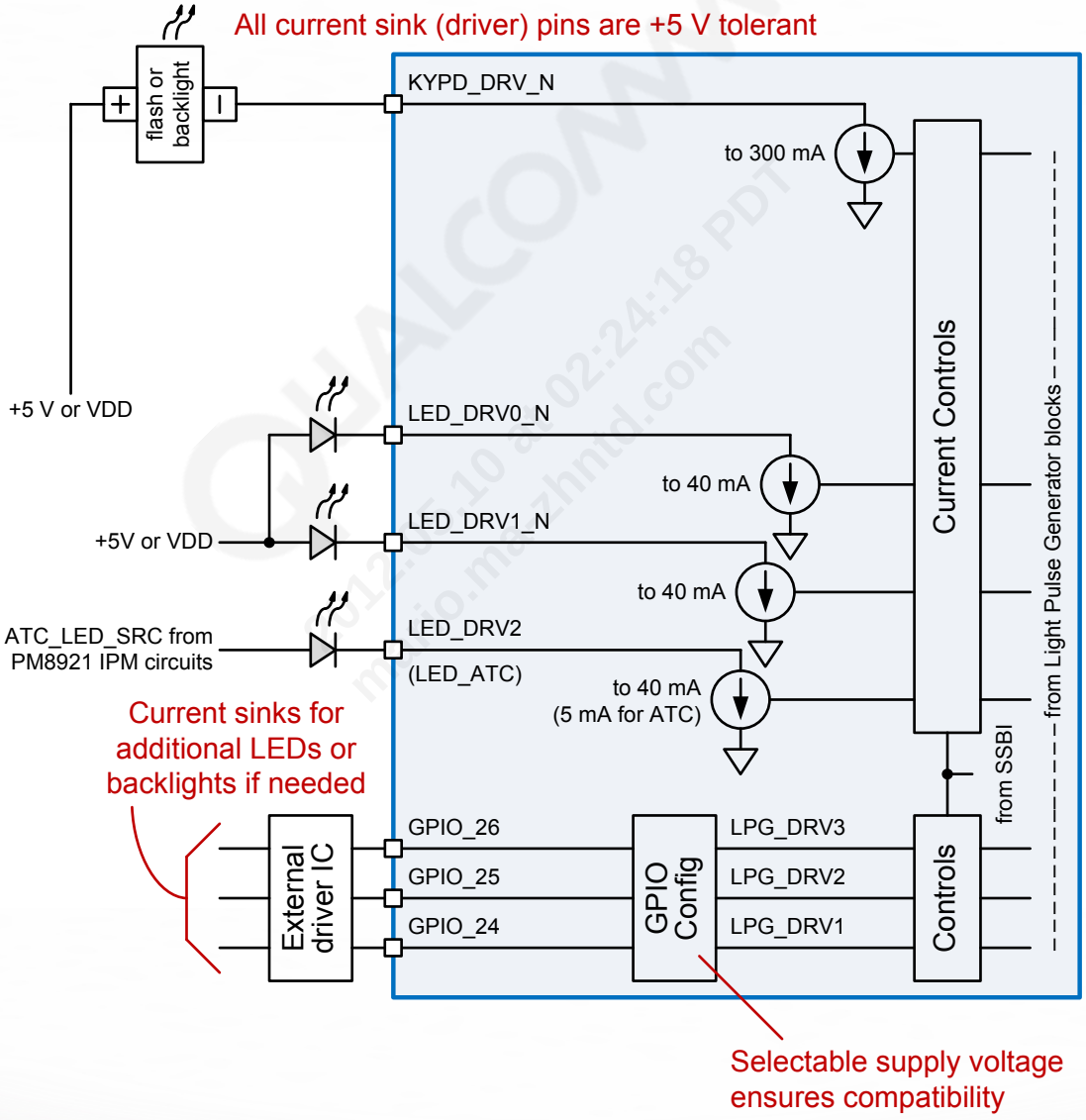
Vibrator Application Example (cont.)

- At this point, the vibrator turns on (3 V) if MPP5 is high and turns off if MPP5 is low.

Current Drivers

- The PMIC includes a set of dedicated current drivers. There are MPPs that can be configured as current sinks and GPIOs that can be connected to a Pulse Width Modulation (PWM) generator for flashing LEDs, and GPIOs that can be configured as digital output for driving external current sinks.
  - A fixed 5 mA driver can be enabled/disabled to indicate the status of the auto trickle charger (LED_ATC).
  - MPPs can be configured as current sinks (LED_DRV_N).
  - The embedded PWM generator (LPG) can be routed to GPIOs for controlling off-chip drivers.

Current Drivers (cont.)



HSED and MIC BIAS

- The PMIC supports Headset Send/End Detection (HSED) – Also known as One-Touch Headset Control (OTHC). Three HSED_BIAS circuits and I/Os, each providing microphone biasing and HSED.
- Each HSED_BIAS pin serves a dual purpose
  - Generates the microphone bias
  - Detects headset switch changes
- Switch transitions are reported via interrupts to the modem device.
- Normally Open (NO) and Normally Closed (NC) MIC End/Send buttons are supported.
- Insertion/removal detection is supported for NO headsets, but not NC types.

HSED and MIC BIAS (cont.)

- Related APIs

```
/*Sets current thresholds. 10 to 160 µA for NC and 200 to 1700 µA
  for NO */
```

- `pm_hsed_set_current_threshold()`

```
/*Configures HSED to always ON, OFF, follows TCXO_EN or (PWM |
  TCXO_EN) control */
```

- `pm_hsed_enable()`

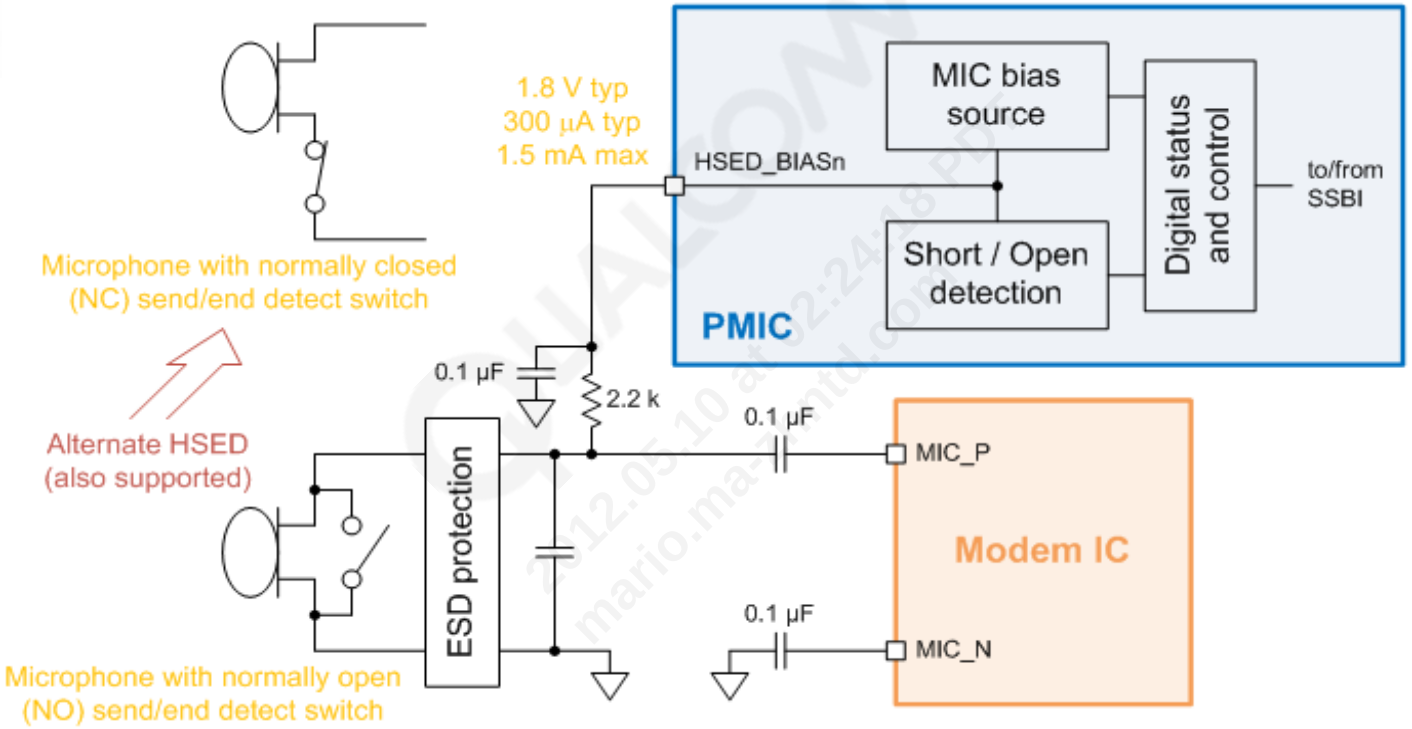
```
/*Configures the hysteresis clock (1.024 kHz) predivider and the
  hysteresis clock*/
```

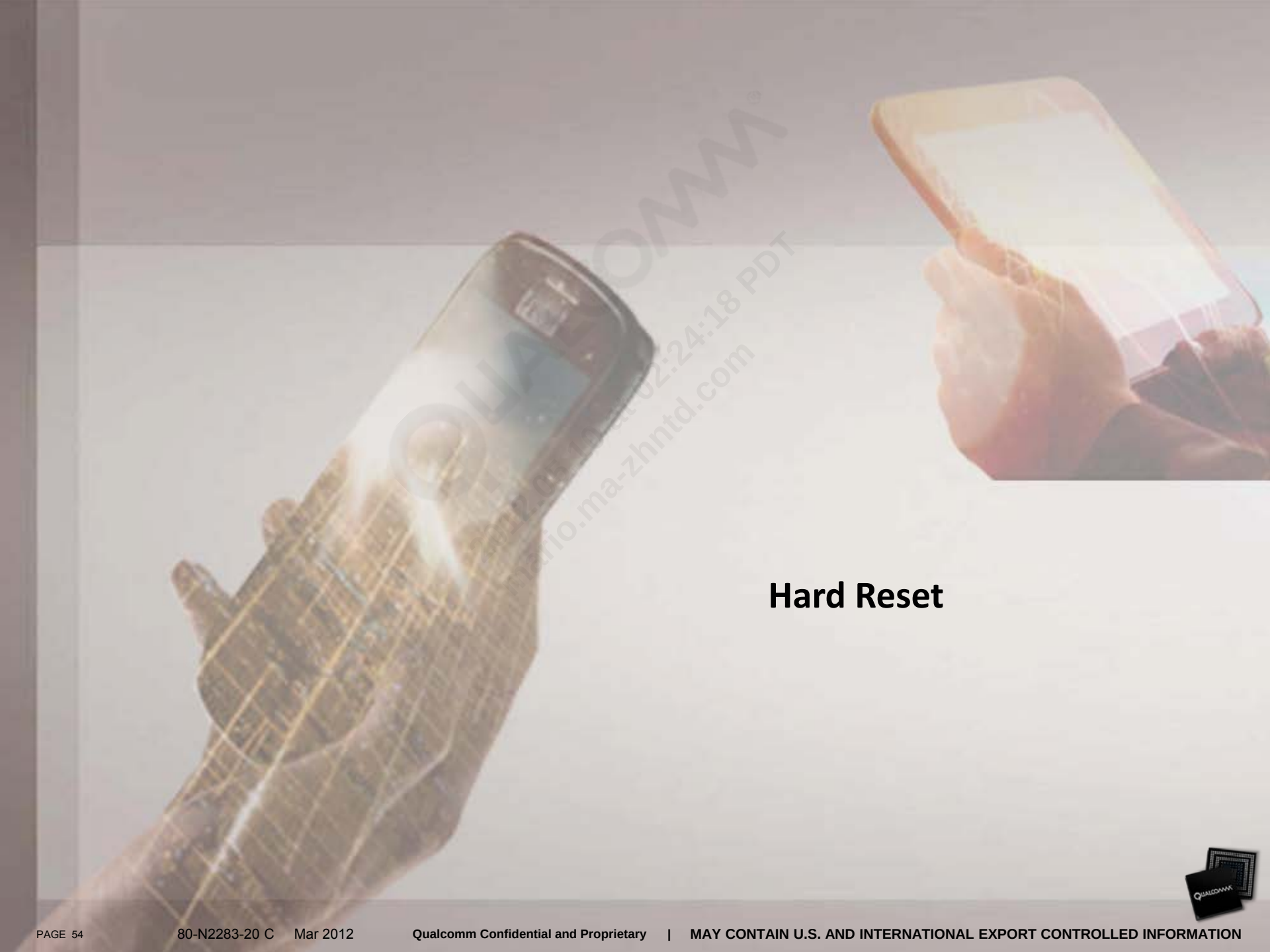
- `pm_hsed_set_hysteresis()`

```
/*Configures the period clock (1.024 kHz) predivider and the period
  clock */
```

- `pm_hsed_set_period()`

HSED and MIC BIAS (cont.)

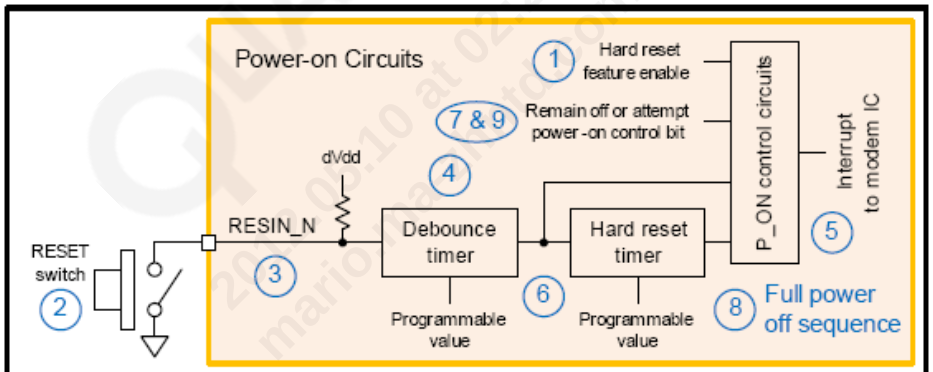




Hard Reset

Hard Reset

- The PMIC includes a feature that allows the handset user to force a hard reset. If the handset locks-up for any reason, including an improper modem IC shutdown, an external switch can be pressed that will cause the PMIC to reset the handset, orchestrate a new power-on sequence, and renew normal operation.



1) Hard reset feature must be enabled via SBI control bit
2) Handset locks-up for some reason, so handset user presses the reset button
3) RESIN_N is pulled low (pulled up internally)
4) PMIC debounce timer begins (programmable from 32 msec to 10.25 seconds)
   - If goes high before timer expires – probably and inadvertent event so do nothing
   - If stays low until timer expires, go to (5)
5) Reset is initiated by sending an interrupt to the modem IC … and …
6) Starting the hard reset timer (programmable from 10 msec to 2 seconds)
7) If the PMIC powers down cleanly (without error) before the hard reset timer expires, the next action is determined by an SBI-controlled bit (remain off or attempt to power-on)
8) If the PMIC does not power off cleanly, a full power-off sequence is executed
9) Once the PMIC has powered off, the next action is determined by the 'remain off or attempt power-on' SBI-controlled bit

Hard Reset (cont.)

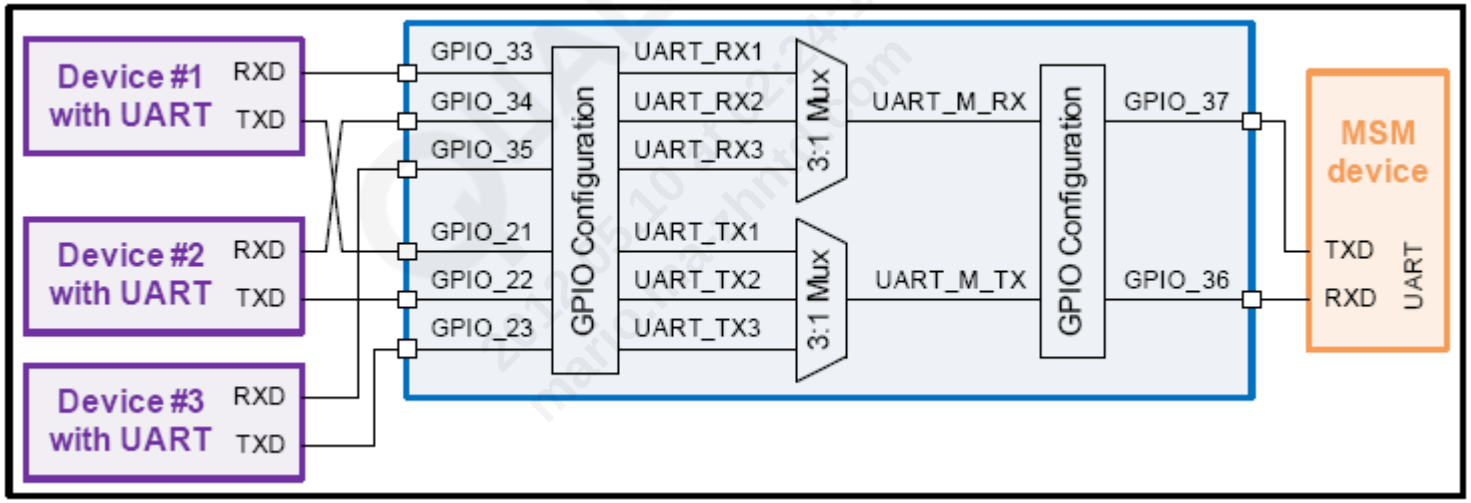
- Related APIs

```
/*Enable hard reset*/
  - pm_pwron_hard_reset_enable()
/*Configure the delay timer. The default value is 500 ms*/
  - pm_pwron_hard_reset_delay_timer_set()
/*Configure the debounce timer. The default value is 3072 ms*/
  - pm_pwron_hard_reset_debounce_timer_set()
```

UART Multiplexing

UART Multiplexing

- The PMIC includes two 3-to-1 multiplexers for routing three phone-level UART interfaces to a single MSM or QSC interface; one multiplexer for the Rx path and one for the Tx path. The associated I/Os are implemented using GPIOs.



- Related API:

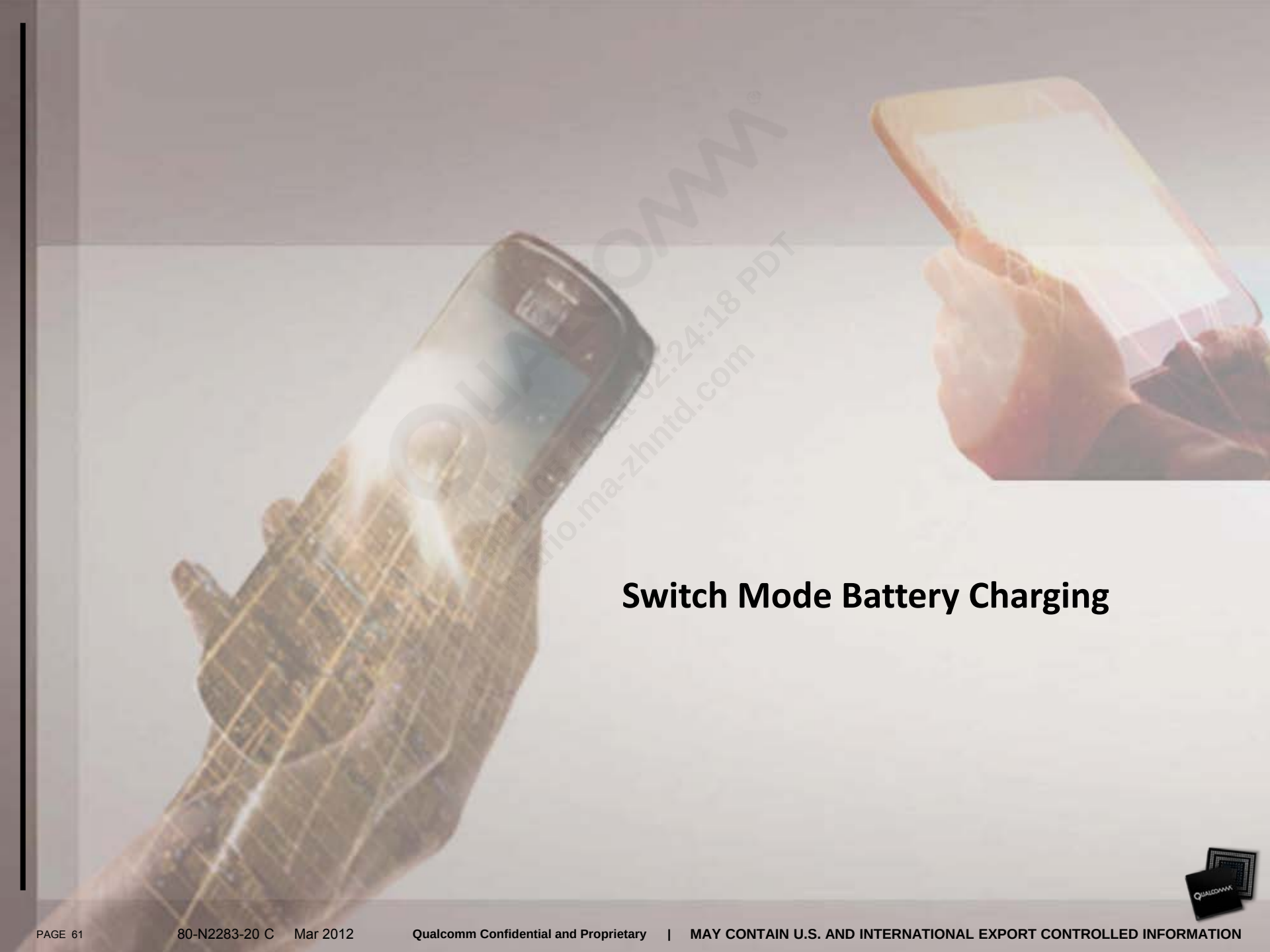
```
/*Selects input channel on 3:1 multiplexer*/
```

  - `pm_gpio_set_mux_ctrl()`

Battery Management

Battery Management Services

- Battery fuel gauge for accurate management of battery resources.
- The module provides function to monitor the battery capacity in conjunction with XOADC, which provides battery voltage information when needed.
- The following PMIC BMS related functions are:
  - Enable/disable BMS module – pm_bms_enable()
  - Set mode Overide/Automatic (default) – pm_bms_set_mode()
  - Enable VBAT measurement in Overide mode – pm_bms_enable_vbat()
  - Configure BMS output data register to get resistance measurement – pm_bms_config_Output()
  - Read 16-bit output data – pm_bms_read_output_data()



Switch Mode Battery Charging

Overview

- With autonomous charging, the PMIC is able to control charging without software intervention
- After a successful startup, the state machine recognizes if the battery needs to be charged and takes the appropriate steps; it conducts battery charging until the end-of-charge requirements are met, and then it stops charging and allows the external supply to power the handset until the freshly charged battery is needed
- Other features of the autonomous charging state machine are:
  - Trickle charging algorithm
  - Fast charging algorithm
- Key SMBC advantage over a linear regulator as the core charger circuit
  - Higher efficiency (and therefore less heat generated)

Note: Pulse charging is not available during state machine operation; it is only available when the handset is using its software-controlled charging mode.

References

Ref.	Document	
Qualcomm		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q3	*PM8921™ Power Management IC Device Specification*	80-N4420-1
Q4	*PM8921™ Power Management IC Revision Guide*	80-N4420-4
Q6	*MSM8960™ Chipset Training: PM8921™ Power Management Topics*	80-N1622-25

Questions?

https://support.cdmatech.com