Click here to ask an associate for production status of specific part numbers.

MAX20355

Power Line Communication with ModelGauge Fuel Gauge and Buck-Boost Converter

General Description

The MAX20355 is a Power Line Communication (PLC) master with ModelGauge™ m5 EZ fuel gauge and buck-boost converter. The MAX20355 and MAX20357 provide a complete system solution for charging and data transfer between a charging case and a battery-powered device over a single contact.

The PLC interface is capable of 100kpbs throughput while simultaneously providing 400mA of total charging/system current. A 4Mbaud half-duplex data-only UART mode provides an easy and fast method for firmware updates, debugging interface, and factory modes.

MAX20355 utilizes a 3.3W buck-boost converter with dynamic voltage scaling (DVS) to automatically manage the charging voltage based on the slave device MAX20357 control signal.

Additional features include comprehensive slave device insertion and removal notifications, moisture detection, overcurrent protection, and 8kV contact rated ESD protection on the PLC outputs.

Applications

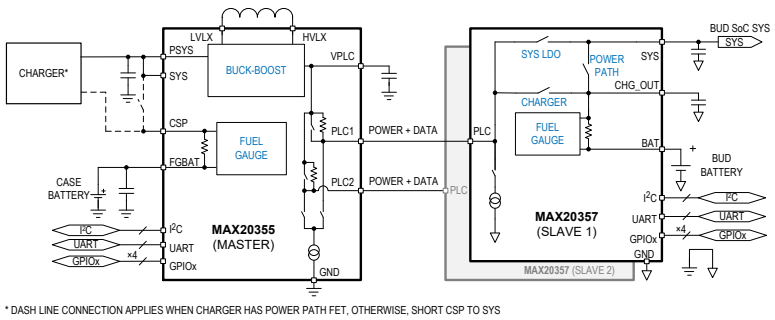
- TWS Headphones
- Augmented Reality Glasses
- Wearable Devices

Benefits and Features

- Power Line Communication (PLC) Interface
  - 100kpbs System Throughput
  - 166.7kbps Bit Rate
  - 200mA per Output Charging Current
  - Automatic Earbud Insertion/Removal Detection
  - 4Mbaud, Half-Duplex, Data-Only UART Mode
  - PLC Controllable GPIOs, Reset and Shipping Mode
- High-Efficiency Autonomous Charging System
  - 90% End-to-End Charging Efficiency from Master to Slave Battery
  - 200mA per Output Charging/System Current
  - Integrated 3.3W Buck-Boost
  - Automatic DVS for each Earbud to Optimize Charging Efficiency
- Robust PLC Output Protection Features
  - Highly Flexible Moisture Detection Block
  - Programmable Overcurrent Protection
  - 8kV Contact ESD Protection
- Small Solution Size
  - Small Single 1608 Inductor
  - 3.08mm x 2.83mm, 42-Bump Wafer-Level Package (WLP)

***Ordering Information* appears at end of data sheet.**

Simplified Application Circuit



ModelGauge is a trademark of Maxim Integrated Products, Inc.

19-101273; Rev 0; 6/22

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Absolute Maximum Ratings

FGBAT, PSYS, SYS to GND ..... -0.3V to +6V
CSP to FGBAT ..... -10mV to +10mV
ASYS to SYS ..... -0.1V to +0.1V
VDIG to GND ..... -0.3V to +2V
LVLX to GND ..... -0.3V to +6V
LVLX to PSYS ..... +0.3V
HVLX to GND ..... -0.3V to +6V
HVLX to VPLC ..... +0.3V
PLC1, PLC2 to GND ..... -0.3V to +6V
VPLC to GND ..... -0.3V to +6V
SDA, SCL, INTb to GND ..... -0.3V to +6V
EN, GPIO1, GPIO2, GPIO3, GPIO4 to GND ..... -0.3V to +6V
THM to GND ..... -0.3V to +6V
THM to FGBAT ..... +0.3V
ALRT to GND ..... -0.3V to +17V
UART_R, UART_T ..... -0.3V to +6V
CTC to GND ..... -0.3V
CTC to FGBAT ..... +0.3V
GND, PGND, DGND ..... -0.3V to +0.3V
Operating Temperature Range ..... -40ºC to +85ºC
Junction Temperature ..... +150ºC
Storage Temperature Range ..... -40ºC to +150ºC
Soldering Temperature (reflow) ..... +260ºC
FGBAT to CSP sense resistor current limit (10% utilization for 100k hours) 1.5A
FGBAT to CSP sense resistor current limit (Peak pulsed current, 250ms maximum pulse width, 10% maximum duty cycle, 1% utilization) 2.7A

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Package Information

WLP

Package Code	W422B3+1
Outline Number	*21-100524*
Land Pattern Number	Refer to *Application Note 1891*
Thermal Resistance, Four Layer Board:	
Junction-to-Ambient (θ_{JA})	44.11°C/W
Junction-to-Case Thermal Resistance (θ_{JC})	NA

Electrical Characteristics

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY (SYS, PSYS, ASYS, FGBAT, VDIG)						
V_{SYS} Voltage Range	V_{SYS}		2.8		5.5	V
V_{SYS} Start-Up Voltage	V_{SYS_STUP}		3.1			V
FGBAT Voltage Range	V_{FGBAT}		2.7		4.9	V
FGBAT Start-Up Voltage	V_{FGBAT_STUP}		3.1			V
SYS POR Threshold	$V_{SYS\_POR\_R}$	Supply rising		2.47	2.69	V
	$V_{SYS\_POR\_F}$	Supply falling	2.25	2.41		
SYS UVLO Threshold	$V_{SYS\_UVLO\_R}$	Supply rising		2.75	2.85	V
	$V_{SYS\_UVLO\_F}$	Supply falling	2.6	2.70		

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
FGBAT UVLO Threshold	$V_{FGAT\_UVLO\_R}$	Supply rising			2.5	2.7	V
	$V_{FGAT\_UVLO\_F}$	Supply falling		2.2	2.4		
VDIG POR Threshold	$V_{DIG\_POR\_R}$	Supply rising			1.42	1.51	V
	$V_{DIG\_POR\_F}$	Supply falling		1.16	1.34		
Supply Current	I_{SYS}	Low_pwr_ena = 1	EN = 1, master/slave detection		4		µA
			EN = 1, idle mode (low-current PU/PU ping-pong)		18		
		V_{OVLO} = 5.5V; SYS = 3.6V	EN = 1, PLC lines powered, PING active		300		
			EN = 1, PLC lines powered, active data transmission with min I PLC		72		mA
Shutdown Supply Current	I_{SYS_SHDN}	EN = 0, device disabled			0.1	1	µA
BUCK-BOOST CONVERTER (LVLX, HVLX, VPLC)							
Output Voltage Range	V_{OUT}			2.5		V_{OVLO}	V
Programmable Output Voltage Resolution	V_{OUT_STEP}				13.7		mV
Max Inductor Current	I_{L_MAX}				2		A
Max Output Power	P_{MAX}	V_{SYS} ≥ 2.9V; i2c_zccm_enb = 0		3.3			W
Overvoltage Threshold	V_{OVLO}	Programmable in 100mV steps		4.8		5.5	V
POWER LINE COMMUNICATION (PLC1, PLC2)							
Data Throughput	T_{Pdat}	Maximum output current			>100		kbps
PLC RCHG	R_{CHG}				0.2	0.4	Ω
		Mono-slave PLC1 = PLC2 internally shorted			0.1	0.2	
PLC RTX	R_{TX}	I_PLC = 200mA		1.8	2	2.2	Ω
		I_PLC = 400mA	Mono-slave PLC1 = PLC2 internally shorted	0.9	1	1.1	
Receiver Threshold	R_{TX_TH}	PLC1/2 = 4V	PLC_CONFIG0[1:0] = 00		-40		mV
			PLC_CONFIG0[1:0] = 01		-56		
			PLC_CONFIG0[1:0] = 10		-70		
			PLC_CONFIG0[1:0] = 11		-84		
OVERCURRENT PROTECTION							

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Programmable Overcurrent Range	I_{LIMIT}			100		400	mA
		Mono-slave PLC1 = PLC2 internally shorted		200		800	
Programmable Overcurrent Resolution	I_{LIMIT_STEP}				50		mA
		Mono-slave PLC1 = PLC2 internally shorted			100		
Overcurrent Accuracy	I_{LIMIT_ACC}	PLC RCHG				20	%
		PLC RTX				30	
Overcurrent Response Time	I_{LIMIT_DEL}				16		µs
MOISTURE DETECTION							
Impedance Measurement Range	R_{MOIST_RANGE}			6		1400	kΩ
Impedance Measurement Accuracy	R_{MOIST_ACC}				10		%
Current Source Accuracy	I_{ID_ACC}	PLC1/2 = 1.5V	I_{ID} = 1, 4, 16, 64µA	-5.5		+5.5	%
Current Source	I_{ID}				1		µA
					4		
					16		
					64		
ADC Resolution	ADC_RES				8		bit
ADC Voltage Step	ADC_STEP				5.9		mV
ADC Full-Scale Error	ADC_ERR			-2		+2	%
ADC Noise Filtering	ADC_NOISE				100		µs
ADC Full Scale	ADC_SCALE				1.5		V
PLC COMMUNICATION CURRENT SINK							
Current Sink Accuracy	I_{SINK_ACC}			-14		+14	%
Current Sink	I_{SINK}	PLC1/2 = 4V	PLC_CONFIG0[7:6] = 00		85		mA
			PLC_CONFIG0[7:6] = 01		102		
			PLC_CONFIG0[7:6] = 10		119.4		
			PLC_CONFIG0[7:6] = 11		136		
UART SWITCH							
Analog Signal Range	V_{UART_RNG}			0		MIN(3.3, V_{SYS})	V
On-Resistance	R_{ONUART}				12	24	Ω
On-Resistance Flatness	$R_{FLATUART}$				0.1	0.3	Ω
Off-Leakage Current	$I_{LUART(OFF)}$					1	µA

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
FUEL GAUGE POWER SUPPLY (FGBAT)						
Shutdown Supply Current	I_{DD0}			0.5		µA
Hibernate Supply Current	I_{DD1}	Average current		5.1		µA
Active Supply Current	I_{DD2}	Average current not including thermistor measurement current		15		µA
FUEL GAUGE ANALOG-TO-DIGITAL CONVERSION						
FGBAT Measurement Error	V_{GERR}	T_A = +25°C	-7.5		+7.5	mV
		-40°C ≤ T_A ≤ +85°C	-20		+20	
FGBAT Measurement Resolution	V_{LSB}			78.125		µV
FGBAT Measurement Range	V_{FS}		2.7		4.9	V
Sense Resistance	R_{SNS}	T_A = +25°C		15		mΩ
Current Measurement Offset Error	I_{OERR}	Long term average without load current		±1		mA
Current Measurement Resolution	I_{LSB}			104.167		µA
Current Measurement Gain Error	I_{GERR}	(Note 1)		±2.5		% of Reading
Current Measurement Error	I_{ERR}	0.25A and 0.5A (Note 1)	-3		+3	% of Reading
Internal Temperature Measurement Error	TI_{GERR}	-40°C ≤ T_A ≤ +85°C		±1		ºC
Internal Temperature Measurement Resolution	TI_{LSB}			0.00391		ºC
FUEL GAUGE INPUT/OUTPUT						
External Thermistance Resistance	R_{EXT10}	Config.R100 = 0		10		kΩ
	R_{EXT100}	Config.R100 = 1		100		
Input Logic High, ALRT	V_{IH}		1.5			V
Input Logic Low, ALRT	V_{IL}				0.5	V
Battery-Detach Detection Threshold	V_{DET}	Measured as a fraction of V_{FGBAT} on TH rising	91	96.2	99	%
Battery-Detach Detection Threshold Hysteresis	$V_{DET-HYS}$	Measured as a fraction of V_{FGBAT} on TH falling		1		%
Battery-Detach Comparator Delay	t_{TOFF}	TH step from 70% to 100% of V_{FGBAT} (Alrtp = 0, EnAIN = 1, FTHRM = 1)			100	µs
FUEL GAUGE LEAKAGE						
Leakage Current, THM	I_{LEAK_THM}		-500		+500	nA
FUEL GAUGE TIMING						
Time-Base Accuracy	t_{ERR}	T_A = +25°C	-1		+1	%
		-40°C ≤ T_A ≤ +85°C	-5		+5	
TH Pre-charge Time	t_{PRE}		8.48			ms
DIGITAL SIGNALS (SDA, SCL, nINT, EN, GPIO1, GPIO2, GPIO3, GPIO4, UART_T, UART_R, ALRT)						

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
SDA, SCL, nINT, EN, GPIO1, GPIO2, GPIO3, GPIO4, UART_T, UART_R Input leakage	I_{LK_IO}	Input pullup/down resistor disabled		-1		+1	µA
ALRT Input Leakage	I_{LK_ALRT}	V_{ALRT} < 15V		-1		+1	µA
SDA, SCL, EN Input Logic low	V_{IL_IO}					0.4	V
SDA, SCL, EN Input Logic high	V_{IL_IH}			1.4			V
GPIO1, GPIO2, GPIO3, GPIO4 Input CMOS Logic low	$V_{IL\_IO\_C}$	GPIOCmosEn = 1			0.3 x SYS		V
GPIO1, GPIO2, GPIO3, GPIO4 Input CMOS Logic high	$V_{IL\_IH\_C}$	GPIOCmosEn = 1			0.7 x SYS		V
GPIO1,GPIO2, GPIO3,GPIO4 Input TTL Logic low	$V_{IL\_IO\_T}$	GPIOCmosEn = 0				0.4	V
GPIO1, GPIO2, GPIO3, GPIO4 Input TTL Logic high	$V_{IL\_IH\_T}$	GPIOCmosEn = 0		1.4			V
SDA, nINT, ALRT GPIO1, GPIO2, GPIO3, GPIO4 Output Logic Low	V_{OL}	I = 4mA				0.4	V
I²C TIMING (SDA, SCL)							
SCL Clock Frequency	f_{SCL}	(Note 2)		0		400	kHz
Bus Free Time Between a STOP and START Condition	t_{BUF}			1.3			µs
Hold Time (Repeated) START Condition	$t_{HD:STA}$	(Note 3)		0.6			µs
Low Period of SCL Clock	t_{LOW}			1.3			µs
High Period of SCL Clock	t_{HIGH}			0.6			µs
Setup Time for a Repeated START Condition	$t_{SU:STA}$			0.6			µs
Data Hold Time	$t_{HD:DAT}$	(Note 4, Note 5)		0		0.9	µs
Data Setup Time	$t_{SU:DAT}$	(Note 4)		100			ns
Setup Time for STOP Condition	$t_{SU:STO}$			0.6			µs
Spike Pulse Width Suppressed by Input Filter	t_{SP}	(Note 6)				50	ns
ESD PROTECTION							
ESD Protection Rating		PLC1, PLC2	Contact discharge	8			kV
		All other pins	HBM	2			
THERMAL PROTECTION							

(T_A = -40°C to +85°C, FGBAT = +2.7V to +4.9V, SYS = +2.8V to 5.5V, C_{SYS} = 10µF, C_{DIG} = 1µF, L_{VLX} = 2.2µH (ESR < 200mΩ), C_{ASYS} = 0.1µF, C_{VPLC} = 8µF, C_{FGBAT} = 10µF, dual-slave configuration if not otherwise specified)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Overtemperature Threshold	TH_{OVER}			150		°C

Note 1: GBD and not production tested.

Note 2: Timing must be fast enough to prevent the IC from entering shutdown mode due to bus low for a period greater than the shutdown timer setting.

Note 3: f_{SCL} must meet the minimum clock low time plus the rise/fall times.

Note 4: The maximum t_{HD_DAT} has only to be met if the device does not stretch the low period (t_{LOW}) of the SCL signal.

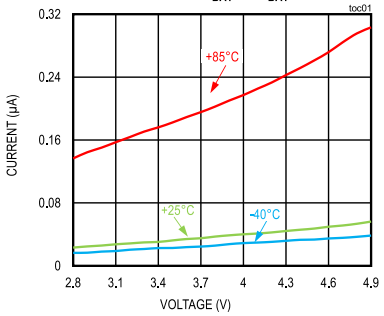
Note 5: This device internally provides a hold time of at least 100ns for the SDA signal (see the minimum V_{IH} of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 6: Filters on SDA and SCL suppress noise spikes at the input buffers and delay the sampling instant.

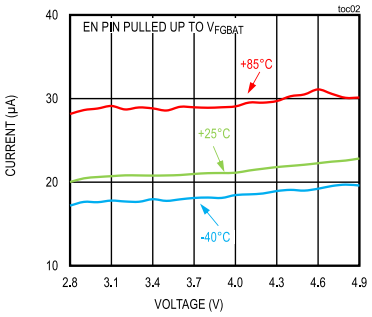
Typical Operating Characteristics

(T_A = +25°C, unless otherwise noted.)

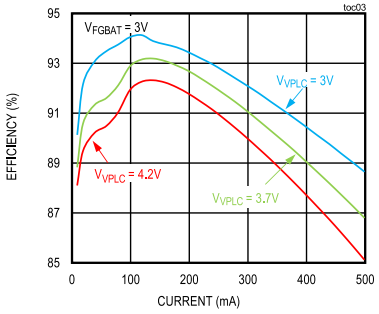
OFF MODE I_{BAT} vs. V_{BAT}



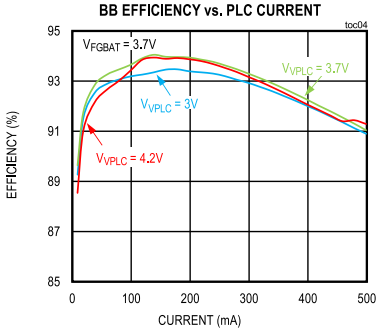
I_{BAT} vs. V_{BAT} PLC DETECTION AND FUEL GAUGE ACTIVE



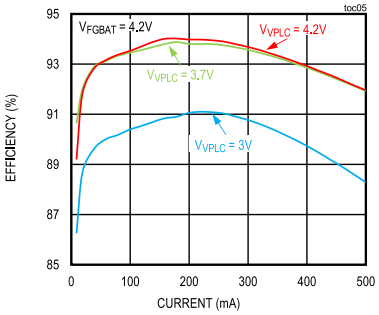
BB EFFICIENCY vs. PLC CURRENT



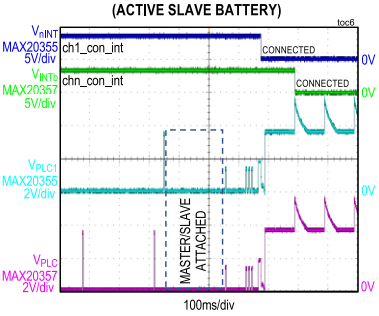
BB EFFICIENCY vs. PLC CURRENT



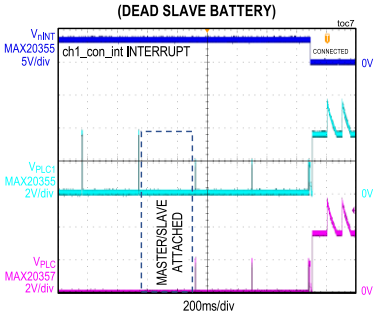
BB EFFICIENCY vs. PLC CURRENT



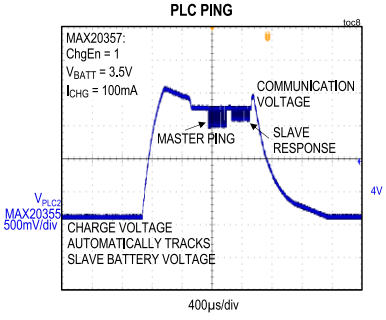
PLC CONNECTION (ACTIVE SLAVE BATTERY)



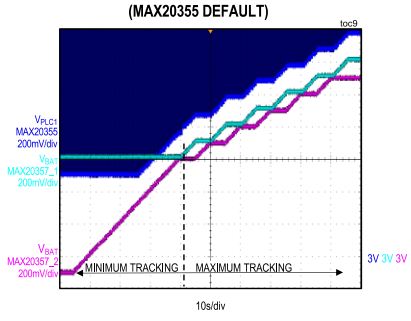
PLC CONNECTION (DEAD SLAVE BATTERY)



PLC PING

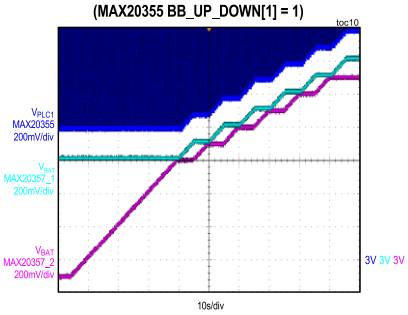


OPTIMIZED TRACKING (MAX20355 DEFAULT)

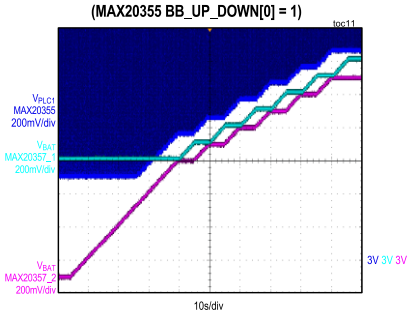


(T_A = +25°C, unless otherwise noted.)

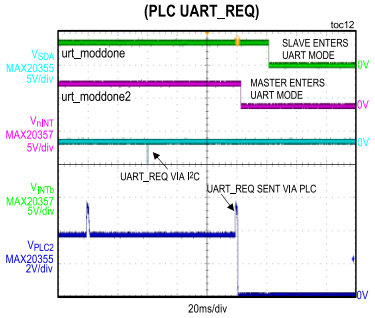
MAXIMUM TRACKING
(MAX20355 BB_UP_DOWN[1] = 1)



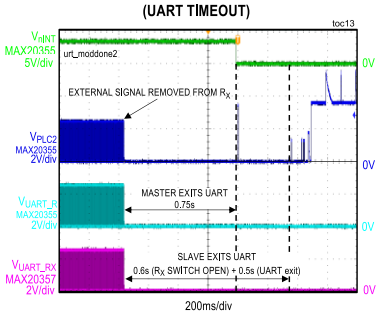
MINIMUM TRACKING
(MAX20355 BB_UP_DOWN[0] = 1)



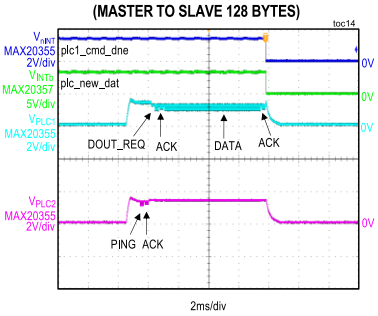
ENTER UART
(PLC UART_REQ)



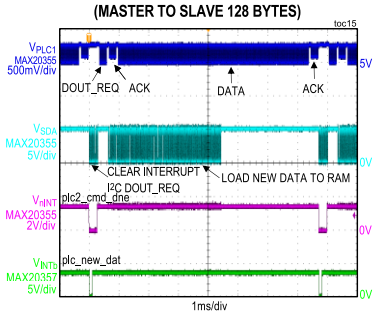
EXIT UART
(UART TIMEOUT)



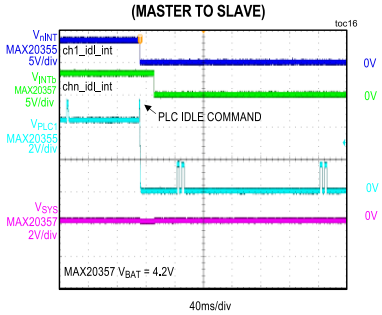
MAILBOX DATA TRANSFER
(MASTER TO SLAVE 128 BYTES)



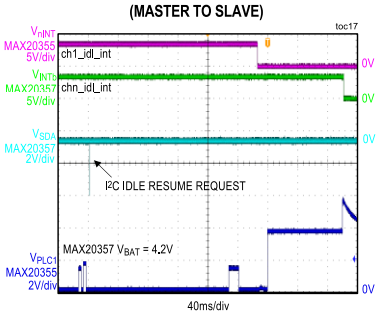
BULK DATA TRANSFER
(MASTER TO SLAVE 128 BYTES)



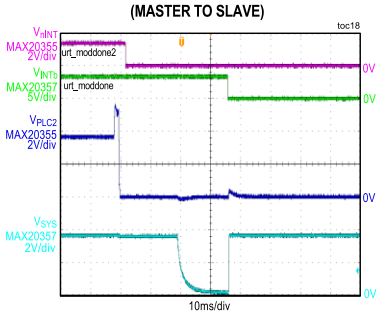
PLC IDLE REQUEST
(MASTER TO SLAVE)



RESUME FROM PLC IDLE
(MASTER TO SLAVE)

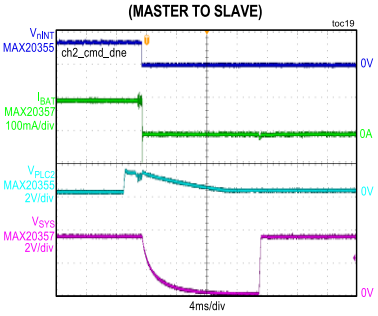


PLC UBOOT
(MASTER TO SLAVE)

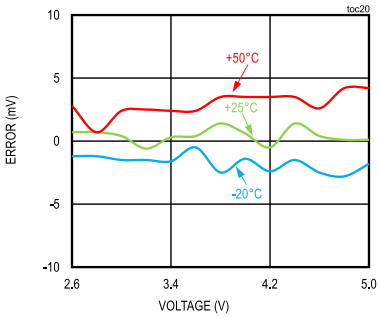


(T_A = +25°C, unless otherwise noted.)

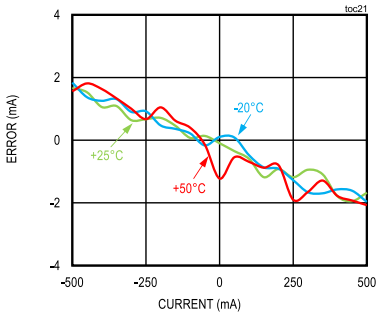
PLC HARD RESET (MASTER TO SLAVE)



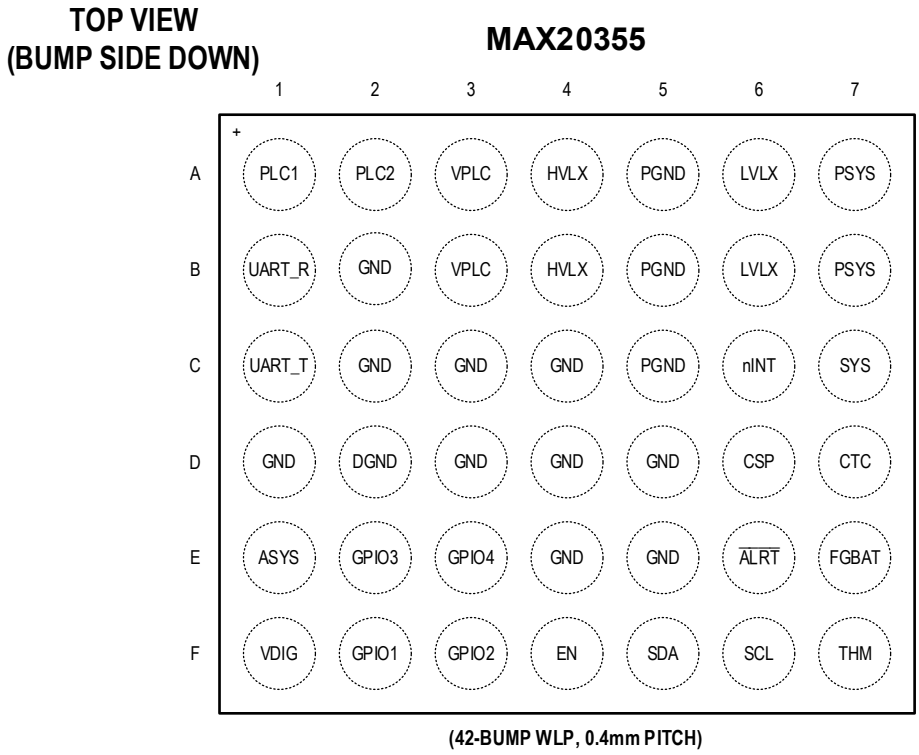
FGBAT VOLTAGE ERROR



FUEL GAUGE CURRENT MEASUREMENT ERROR



Pin Configuration



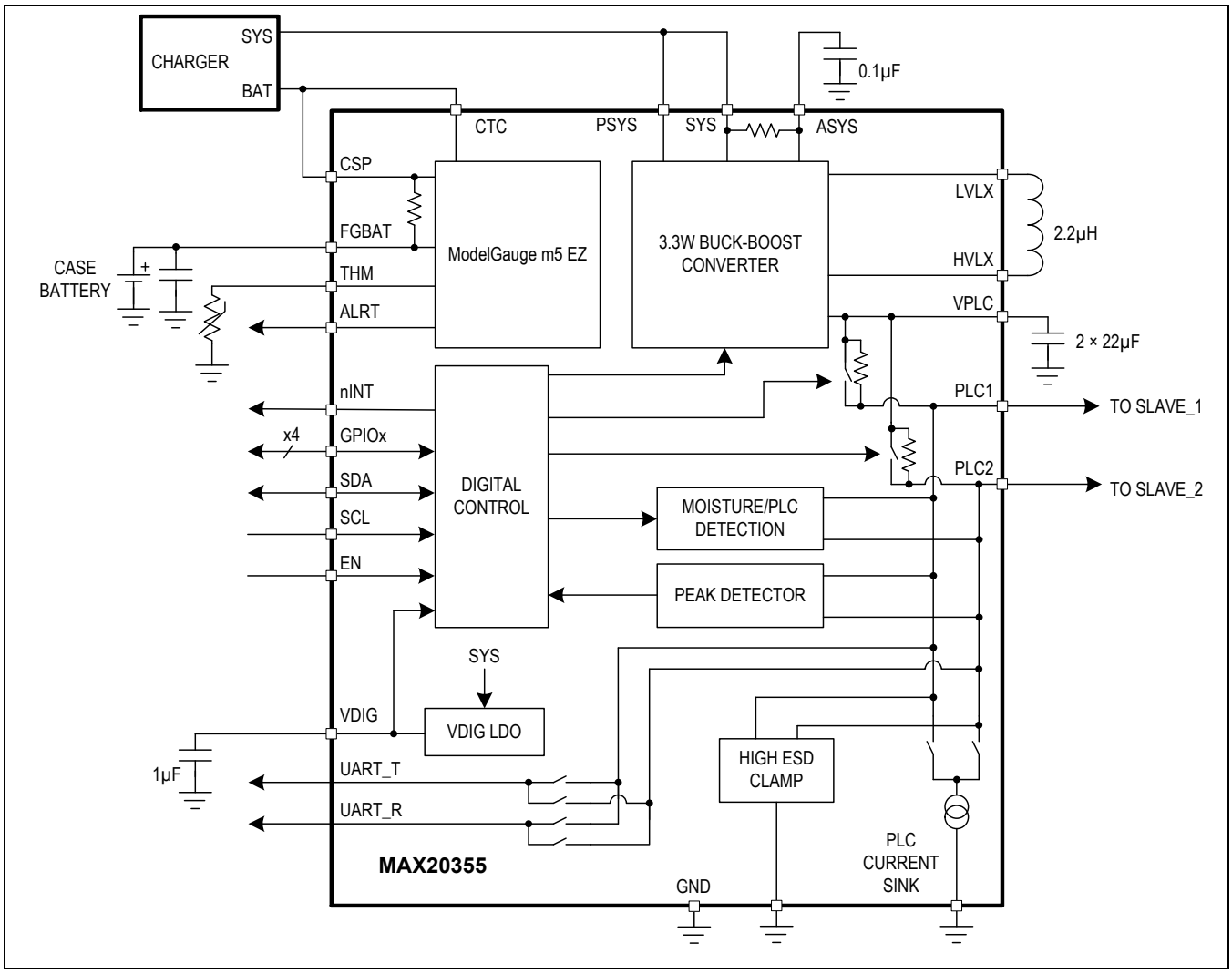
(42-BUMP WLP, 0.4mm PITCH)

Pin Descriptions

PIN	NAME	FUNCTION
A1	PLC1	PLC1 Power Output. Connect to PLC pin on MAX20357.
A2	PLC2	PLC2 Power Output. Connect to PLC pin on MAX20357.
A3, B3	VPLC	PLC Voltage Supply. Bypass with 2 x 22µF capacitor to GND.
A4, B4	HVLX	Buck-Boost High Voltage Switching Node. Connect to inductor.
A5, B5, C5	PGND	Power GND
A6, B6	LVLX	Buck-Boost Low Voltage Switching Node. Connect to inductor.
A7, B7	PSYS	Power SYS. Buck-Boost input. Connect 10µF to GND.
B1	UART_R	UART RX Node. Connect to system processor UART RX.
C1	UART_T	UART_TX Node. Connect to system processor UART TX.
C6	nINT	Open Drain Interrupt. Active low.
C7	SYS	SYS. Connect to PSYS.
D2	DGND	Digital Ground
D6	CSP	Sense Resistor Input Pin
D7	CTC	Short to CSP
E1	ASYS	Analog Sensitive Circuit Supply. Internally connected to SYS by means of a 100Ω resistor. Connect 0.1µF cap to GND.
E2	GPIO3	Programmable GPIO3. May be configured as a digital input or open drain output.

E3	GPIO4	Programmable GPIO4. May be configured as a digital input or open drain output.
E6	ALRT	Alert Output. The ALRT pin is open-drain active-low output which indicates FG alerts. Connect it to GND if not used.
E7	FGBAT	Power Supply and Battery Sense Input. Connect it to positive terminal of a battery cell. Bypass it with 10µF cap.
F1	VDIG	Internal 1.8V Supply. Bypass it with 1µF cap to GND.
F2	GPIO1	Programmable GPIO1. May be configured as a digital input or open drain output.
F3	GPIO2	Programmable GPIO2. May be configured as a digital input or open drain output.
F4	EN	Chip Enable
F5	SDA	I^2C SDA
F6	SCL	I^2C SCL
F7	THM	Battery temperature thermistor measurement input.
C3, C4, D1, D3, D4, D5, E4, E5, C2, B2	GND	Ground

Functional Diagram



Detailed Description

The MAX20355, together with the MAX20357, form an integrated Power Line Communication (PLC) and automated charging system. These devices provide a complete system solution for efficient charging and data transfer between a charging case and wireless earbuds over a single contact.

The MAX20355 and MAX20357 feature data transfers while simultaneously delivering 200mA of charge/system current per output. Both devices support a 166.7kbps bit rate while charging. With a 166.7kbps bit rate, the PLC protocol can achieve up to 130kpbs of throughput using the bulk data transfer feature. Additionally, a 4Mbaud half-duplex data-only UART mode provides an easy and fast method for firmware updates, debug interface, and factory modes. Finally, dedicated hardware PLC commands allow both devices to control GPIOs and enter various power modes, such as ultra-low current ship mode, through the PLC link.

The MAX20355 integrates a completely autonomous charging system that utilizes a 3.3W buck-boost with dynamic voltage scaling (DVS) to provide an optimized charge voltage to the earbuds. An integrated digital state machine automatically manages the dynamic voltage adjustment based on the earbud battery voltage to minimize earbud power dissipation and maximize charging efficiency.

Additional features include comprehensive earbud insertion and removal notifications, even in dead case or bud battery scenarios, and moisture detection on the PLC outputs to prevent contact corrosion. The PLC pins are protected from overcurrent events by an integrated current limiting circuit and ESD events by integrated 8kV contact-rated ESD protection structures.

Power Line Communication (PLC)

The MAX20355/MAX20357 PLC interface offers a means by which charging and exchange of data can occur simultaneously on a single wire connection. The interface can accommodate 100kbps of throughput and can do so while delivering up to 200mA of current per channel in dual slave mode and up to 400mA in mono slave mode. The MAX20355 has an integrated current sink and resistor R_{TX} as shown in *Figure 1*. During communication, the MAX20355 utilizes the pulses generated by the PLC current sink over resistor R_{TX} to create PLC pulses on the PLC line. The MAX20355 uses PLC pulses to transfer information to the MAX20357. At the same time, the charging function is not affected. The MAX20357 uses a similar approach to create PLC pulses and transfer information over the PLC line. When there is no ongoing PLC communication, R_{TX} is shorted by a bypass switch to achieve higher overall power transfer efficiency.

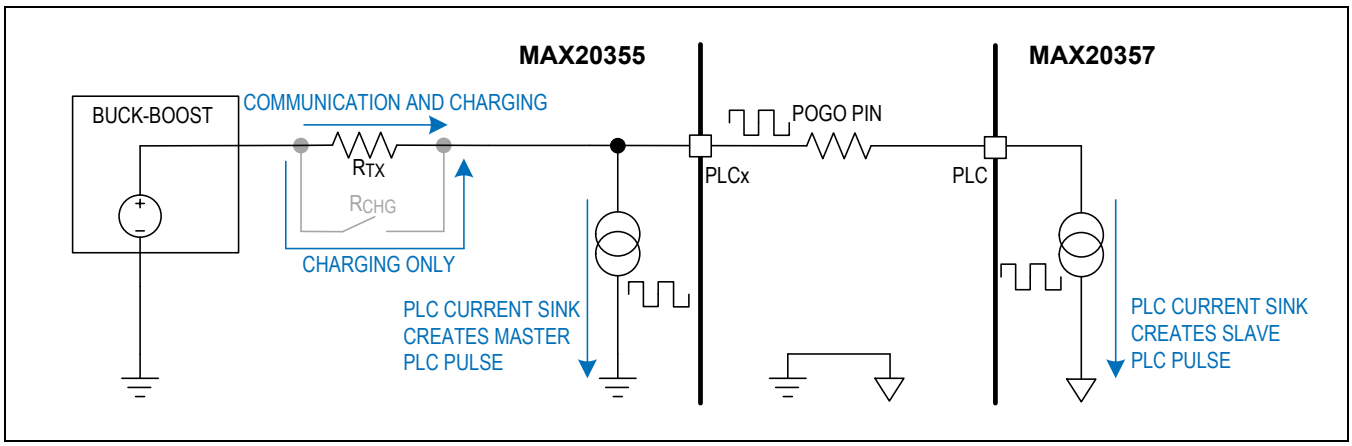


Figure 1. Simplified Scheme of Power Line Communication (PLC)

The PLC interface balances between robustness and throughput. Synchronization mechanisms and checksums offer robust data transfer, but this is also balanced against protocol overhead. With a 166.67kbps bit rate, this careful balance allows a low error rate while achieving throughput up to 130kbps in bulk data transfer mode. To overcome limitations on throughput due to the I^2C interface, the MAX20355/MAX20357 integrates two 128-byte FIFOs. This allows the system to keep I^2C overhead low utilizing bulk writes. The MAX20355 also features a 4Mbaud, data-only, half-duplex UART passthrough mode. This mode uses a simple switch that connects UART ports from master to slave for firmware updates, factory mode, and debug mode over a single PLC line.

The following sections describe the PLC PING (PLC command/mailbox data transfer), FIFO (bulk data transfer), and UART interface in detail.

PLC PING

The PLC interface between master and slave follows a master-initiated scheme. There are two data transfer types—PING and bulk data transfer. PINGs begin with a transfer from master to slave to send data across the interface, request data from the slave, offer the slave a chance to make requests, and sometimes issue commands to the slave. *Figure 2* shows the basic structure of the master to slave PING packet. Master initiates the transmission with a preamble which synchronizes the link and then transmits data. The slave responds in the same manner within a response timeout period. PLC PINGs are automatically generated and transmitted through the PLC line every telemetry period.

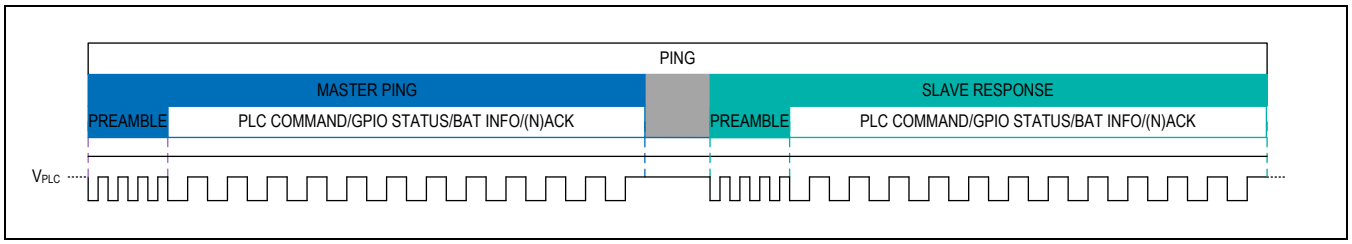


Figure 2. Periodic PING

LISTEN Command and Continuously Updated Information

Each PLC PING contains one PLC command. Most PLC commands are initiated by an I2C write from the system. However, the MAX20355 features automatic LISTEN command, which is sent periodically independent of input from the I2C interface. When there is no customer PLC command issued through I2C, LISTEN command is automatically sent through the PING, the slave has the option to send its command back to the master using its command field. In response to the master's PING, the slave sends the appropriate data based on the contents of the master PING. The contents of the command and data fields are filled according to the requirements of the command sent from the master. The periodic PING offers the following features:

1. Send master PLC command from master to slave (*Table 1*).
2. Give slave an opportunity to send slave PLC commands to master (*Table 2*).
3. Transmit GPIO status between master and slave.
4. Exchange master/slave battery information: master battery voltage, slave battery voltage, slave SoC, and slave charging status.
5. Automatic PLC voltage adjustment to track slave battery voltage and minimize power consumption. Details are described in the *Charger Battery Voltage Tracking Loop* section.
6. Automatic disconnection detection. Master detects slave is disconnected if there is no proper slave response for master PING.

The exchanged GPIO status and battery information between master and slave are ready to read from the master and slave local registers.

Master Commands

In addition to the LISTEN command, which is handled automatically, the master can execute other commands manually as well. Master's command field determines what type of transaction occurs on the PLC interface. Commands over the PLC interface are requested by I2C. *Table 1* describes the various commands master can send and what data and statuses are exchanged. Note that many commands can be sent from master to slave or from slave to master. However, it is only in response to a LISTEN command where the slave may send its commands. Master commands are initiated by writing to the plc_command1/2 and triggered by plc_run_trg1/2 through master I2C. In the command argument register plc_cmd_arg1/2, details on how the command is processed are available for the user to specify.

Table 1. MAX20355 Commands

MASTER COMMANDS	plc_command1/2	plc_cmd_arg1/2	
LISTEN	N/A	N/A	
SET_GPIO	0x3	New slave GPIO setting	
DOUT_REQ	0x5	Number of bytes (up to 128 bytes)	
UART_REQ	0x6	slave_uart_sw[1:0]	
SYST_REQ	0x0	0x00	SEAL request. Puts slave in SEAL mode.
		0x01	Soft reset request. Resets slave's registers.
		0x02	Hard reset request. Cycle the power on the slave's SYS node.
		0x03	Fuel gauge reset request. Resets fuel gauge block.
		0x04	FIFO request. Puts system into bulk data transfer mode.
		0x05	Free request. Stop bulk data transfer and free the data line.
		0x06	IDLE mode request. Sends system into PLC IDLE mode.
		0x07	Hard + Soft reset request. Reset all registers, FSM and cycle the power at SYS.
		{0x3F, slave_uart_tx, slave_uart_rx}	UBOOT request. Hard resets slave and puts slave in UART mode.

Slave Commands

The slave can also issue commands to the master. The slave similarly issues the commands based on I2C input, but it must wait for a master LISTEN command before the command can be sent back to the master. Once the LISTEN command is received, the slave then sends its command using the COMMAND field. *Figure 3* shows the method by which the slave can send a command. Slave commands are initiated with a write to plc_command and triggered by plc_run_trg through master I2C. In the command argument register plc_cmd_arg, details on how to process the PLC commands are available for the user to specify.

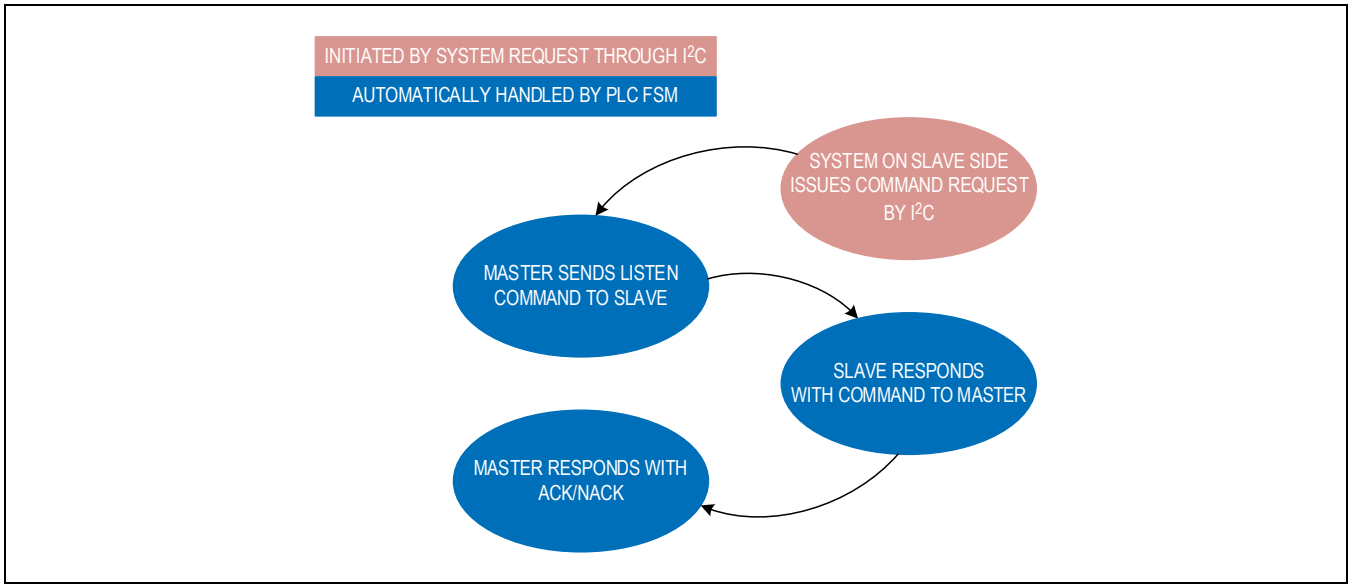


Figure 3. Sending Commands from the Slave

Table 2. Slave Commands

SLAVE COMMAND*	plc_command	plc_cmd_arg (DATA)	
SET_GPIO	0x3	New master GPIO setting	
DOUT_REQ	0x5	Number of bytes (up to 128 bytes)	
UART_REQ	0x6	master_uart_sw[3:0]	
SYST_REQ	0x0	0x01	Soft reset request. Resets master's registers.
		0x03	Fuel gauge reset request. Resets master's fuel gauge block.
		0x04	FIFO request. Puts master and slave into bulk data transfer mode.
		0x05	Free request. Stop bulk data transfer and free the data line.
		0x06	Idle mode request. Sends master into idle mode.

**Always sent in response to the "LISTEN" command from the master shown in Table 1.*

Mailbox Data Transfer (DOUT_REQ)

From Master to Slave

Mailbox data transfers are used to exchange one data packet, up to 128 bytes, from master to slave. A mailbox data transfer begins with a data out request (DOUT_REQ) master command. Fill the data packet into the master RAM, write the number of bytes to be transferred in the command argument field plc_cmd_arg1/2, then trigger the DOUT_REQ command. If slave responds ACK to DOUT_REQ command, data in RAM is transferred to slave automatically. After slave successfully receives data, plc_new_dat interrupt is asserted and RAM_is_full is set to 1. Once RAM_is_full is set to 1, slave RAM is protected from PLC and it sends NACK to master's DOUT_REQ command until RAM_is_full is write cleared. Make sure to read data in RAM before clearing RAM_is_full to avoid data loss. Note that although both PLC devices have two integrated 128-byte RAMs, only one RAM is used in mailbox data transfer. *Figure 4* shows the flow of data during a master to slave write.

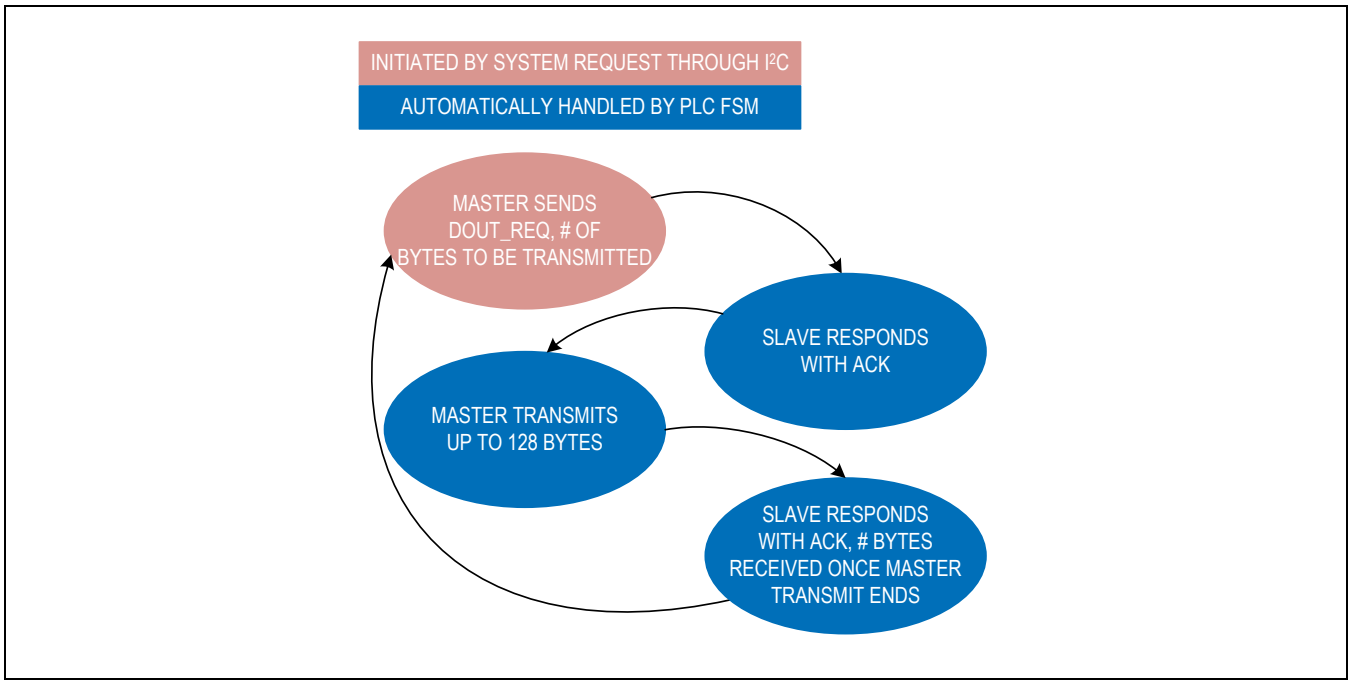


Figure 4. Master to Slave Mailbox Data Transfer

From Slave to Master

A similar scheme is used to send a packet of data, up to 128 bytes, from slave to master. A mailbox data transfer begins with a data out request (DOUT_REQ) slave command. Write the number of bytes to be transferred in the command argument field plc_cmd_arg. Fill the data packet into the slave RAM, write the number of bytes to be transferred in the command argument field plc_cmd_arg, then trigger the DOUT_REQ command. Since the system follows master initiated scheme, the slave triggers the DOUT_REQ command when it receives the master Listen command. After the master receives DOUT_REQ and responds to ACK, data in RAM is transferred to the slave automatically. After master successfully receives data, plc_new_dat interrupt is asserted and RAM_is_full is set to 1. Once RAM_is_full is set to 1, master RAM is protected from PLC and it sends NACK to slave's DOUT_REQ command until RAM_is_full is write cleared. Make sure to read data in RAM before clearing RAM_is_full to avoid data loss. Note that although both PLC devices have two integrated 128-byte RAMs, only one RAM is used in mailbox data transfer. *Figure 5* shows the flow of data during a slave to master write.

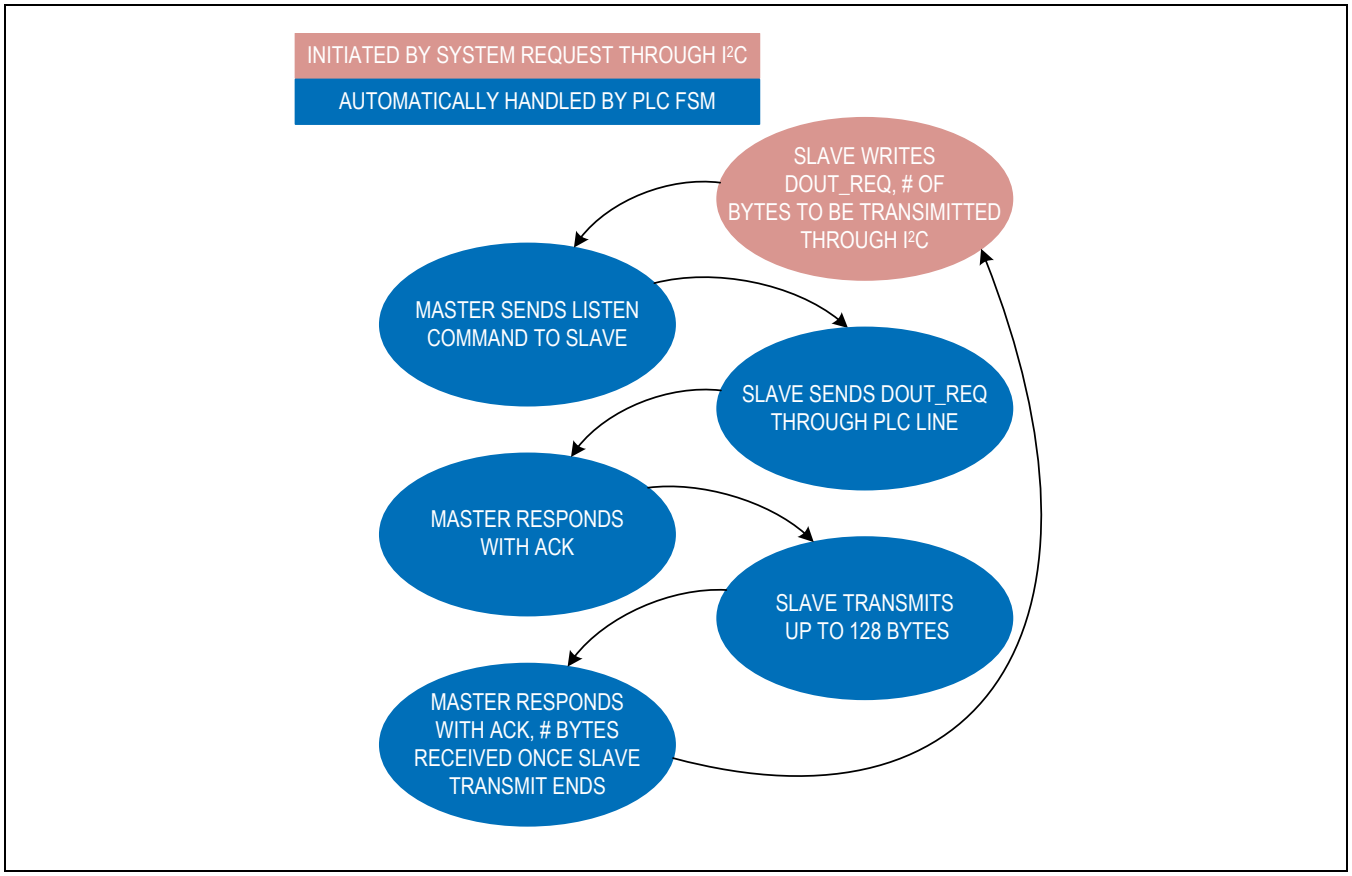


Figure 5. Slave to Master Mailbox Data Transfer

Bulk Data Transfer (FIFO)

To minimize the impact of I²C protocol overhead on the achievable throughput, MAX20355 and MAX20357 integrate two 128-byte RAMs for bulk data transfer. The bulk data transfer scheme is shown in *Figure 6*. The I²C bus connects first to the RAM_1 to load data with I²C bulk writes at data rates up to 400kbps. Once the DOUT_REQ command is triggered, the I²C bus and PLC line are swapped, I²C bus from RAM_1 to RAM_2 and PLC from RAM_2 to RAM_1. While RAM_1 is being emptied by the PLC engine, the next set of data can be loaded by I²C bulk write again into the RAM_2. An interrupt is sent to the system after RAM_1 is completely emptied. Repeat the same process to swap the I²C bus and PLC back and forth to load and transmit data simultaneously. Similar implementation on receiving side swaps I²C bus and PLC back and forth to receive and read data to receiver's microcontroller. Dual-RAM architecture reduces I²C protocol overhead and enhances achievable throughput. Packets of any size up to 128 bytes can be sent by this method.

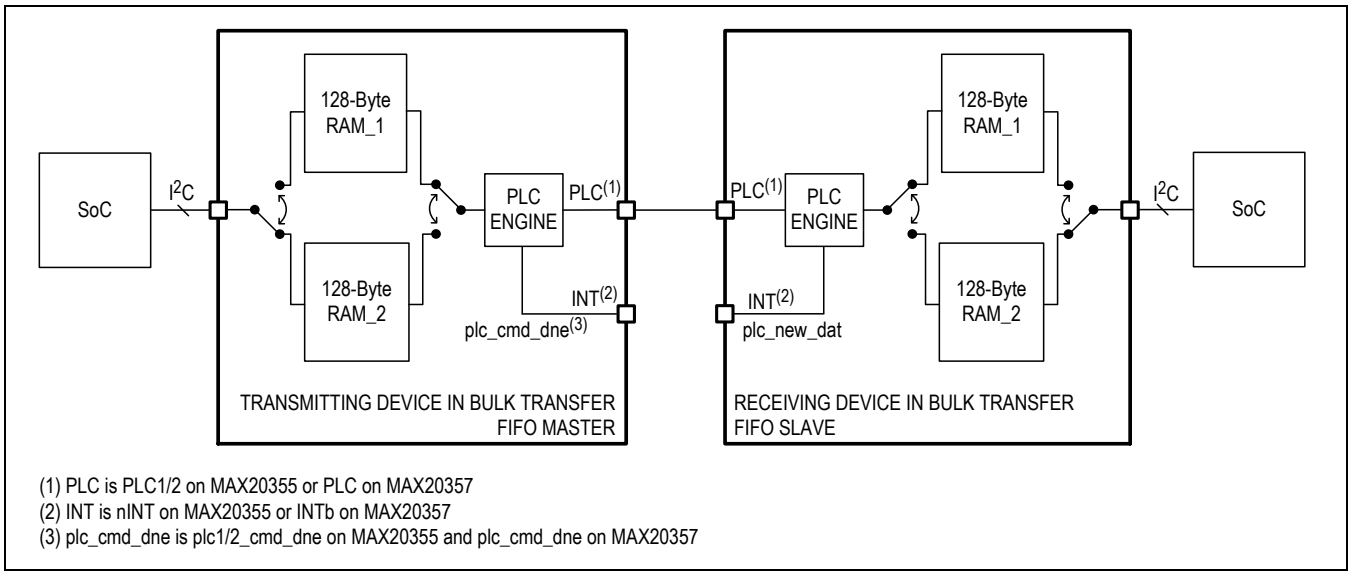


Figure 6. Bulk Data Transfer Implementation

From Master to Slave

Bulk data transfers from master to slave begin with a system request (SYST_REQ) master command. In the command argument of SYST_REQ, the master needs to send a FIFO request. Once receiving an ACK from the slave, PLC voltage is pulled up to communication voltage and the PLC line is locked for bulk data transfer. Before triggering the data transfer, the data packet needs to be filled into the master RAM through I2C. Similar to mailbox data transfer, write the number of bytes to be transferred to plc_cmd_trg1/2, then trigger the DOUT_REQ command and wait for plc1/2_cmd_dne interrupt. Repeat DOUT_REQ and wait for plc1/2_cmd_dne to interrupt multiple times as needed. After all data packets are transferred, issue a FREE request from the master side to unlock the line and return to normal PLC mode. Note that master bulk data transfer is designed to transfer data only from master to slave. *Figure 7* shows the flow of data during a master to slave write.

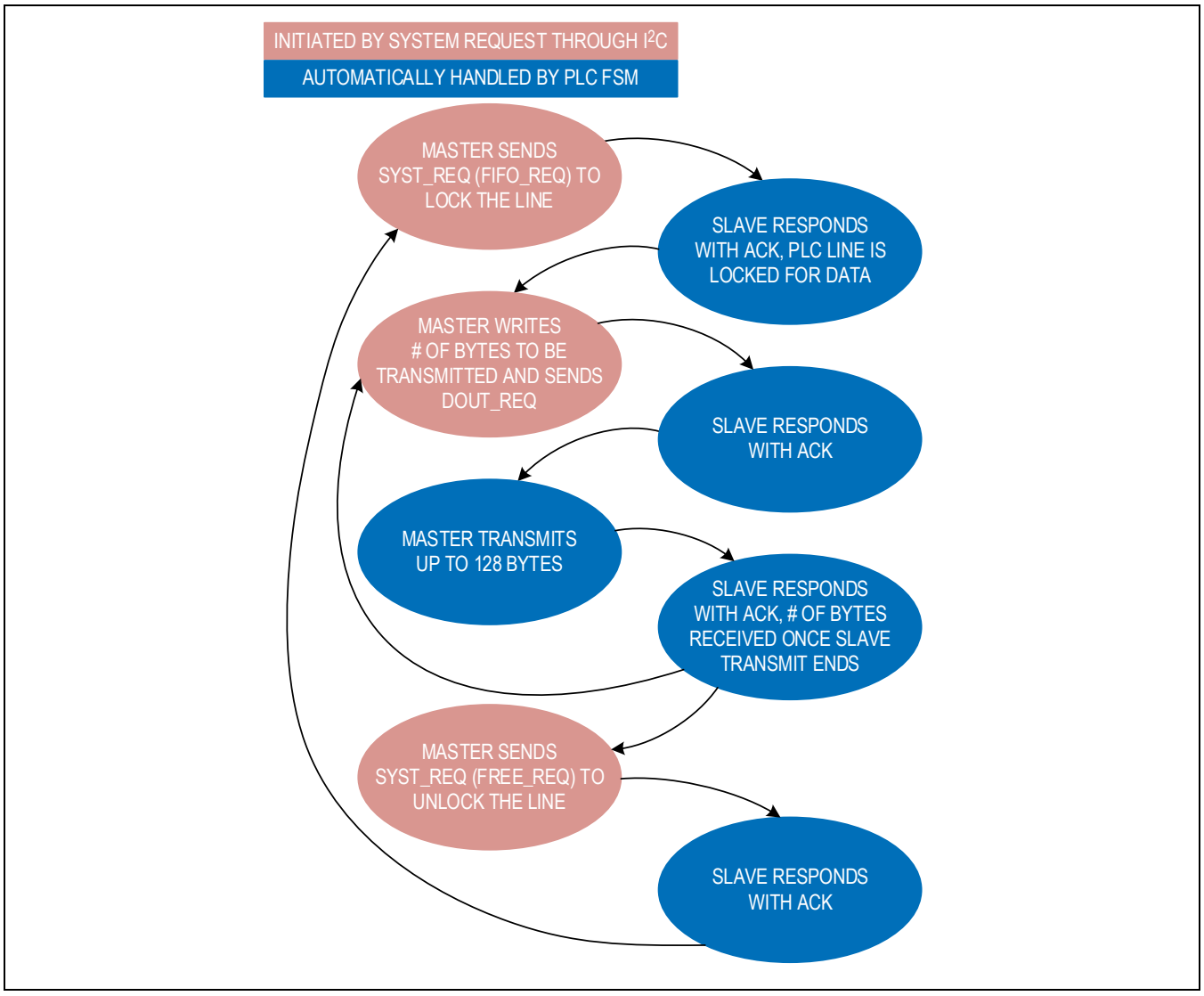


Figure 7. Master to Slave Bulk Data Transfer

From Slave to Master

Bulk data transfers from slave to master begin with a system request (SYST_REQ) slave command. In the command argument of SYST_REQ, the slave needs to write a FIFO request. The slave waits until the master's LISTEN command to send out a FIFO request through the PLC line. Once receiving an ACK from the master, the master locks the PLC line and enters bulk data transfer mode. Before triggering the data transfer, the data packet needs to be filled into the slave RAM. Similar to mailbox data transfer, write the number of bytes to be transferred to plc_cmd_arg, then trigger the DOUT_REQ command and wait for plc_cmd_dne interrupt. Repeat DOUT_REQ and wait for plc_cmd_dne to interrupt multiple times as needed. After all data packets are transferred, issue a FREE request from the slave side to unlock the line and return to normal PLC mode. Note that slave bulk data transfer is designed to transfer data only from slave to master. *Figure 8* shows the flow of data during a slave to master write.

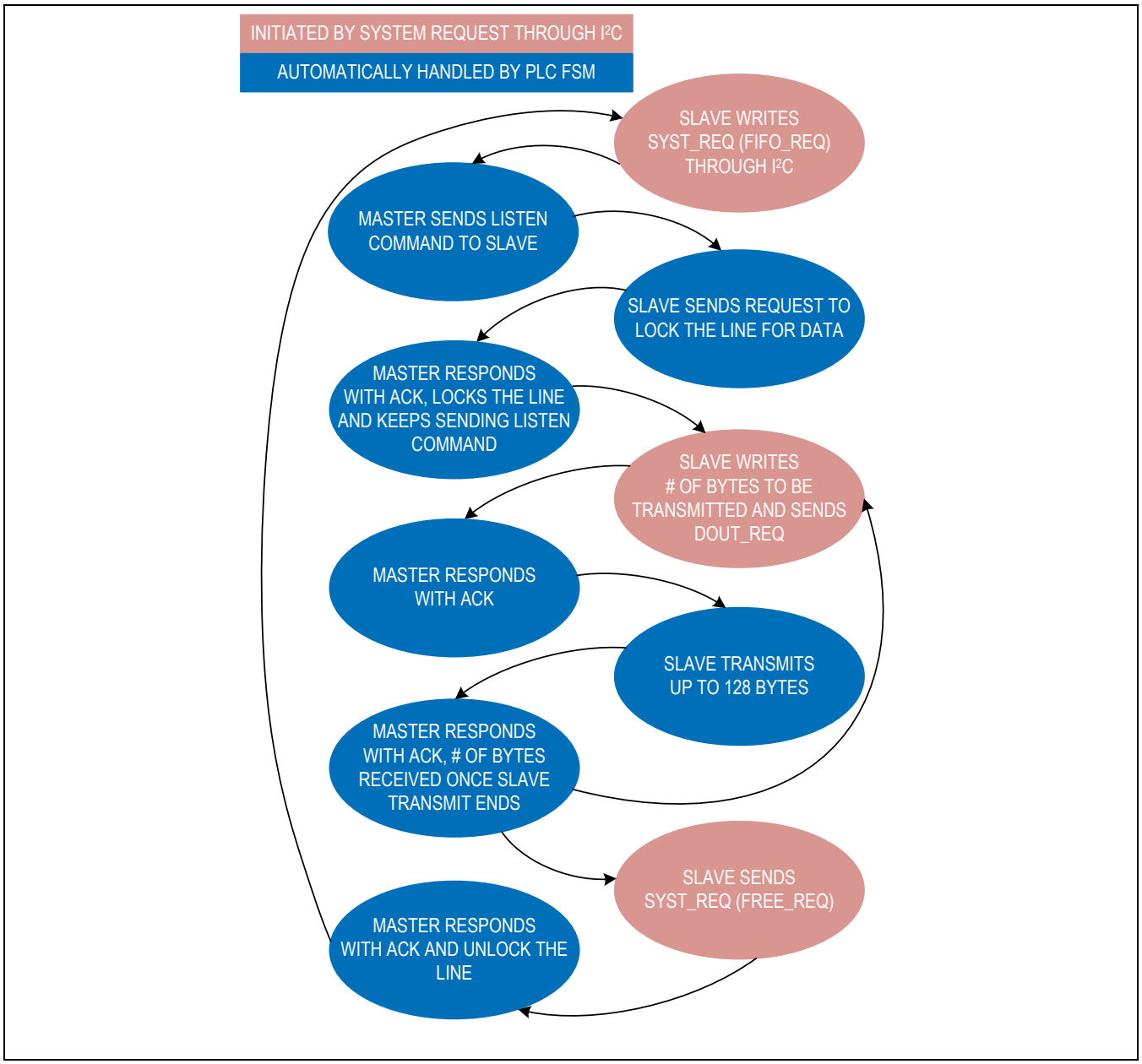


Figure 8. Slave to Master Bulk Data Transfer

Half-Duplex UART Passthrough Interface

The MAX20355 and MAX20357 feature a 4Mbaud, data-only, half-duplex UART passthrough mode. This mode is a simple switch that connects either the TX or RX for firmware updates, factory mode, and debug mode over the single PLC line. A UART command can be sent through the PLC interface to send either device into UART mode in a particular configuration. Internal switches allow the user to separate UART TX and UART RX. The configuration can be seen in *Figure 9*. The user can use the I²C command to switch roles from TX to RX or vice versa on master and slave. The slave provides one additional feature to detect incoming data automatically through its UART_RX pin. In this mode, once the slave enters UART mode, it closes its UART_TX switch by default. Once it detects a rising/falling edge on its UART_RX pin from the slave microcontroller, it toggles the switch from UART_TX to UART_RX automatically. After the data transfer is completed in receiving direction, switch toggles back from UART_RX to UART_TX automatically after a programmable blanking time. This feature allows the user to send and receive data through UART when the slave I²C is not available.

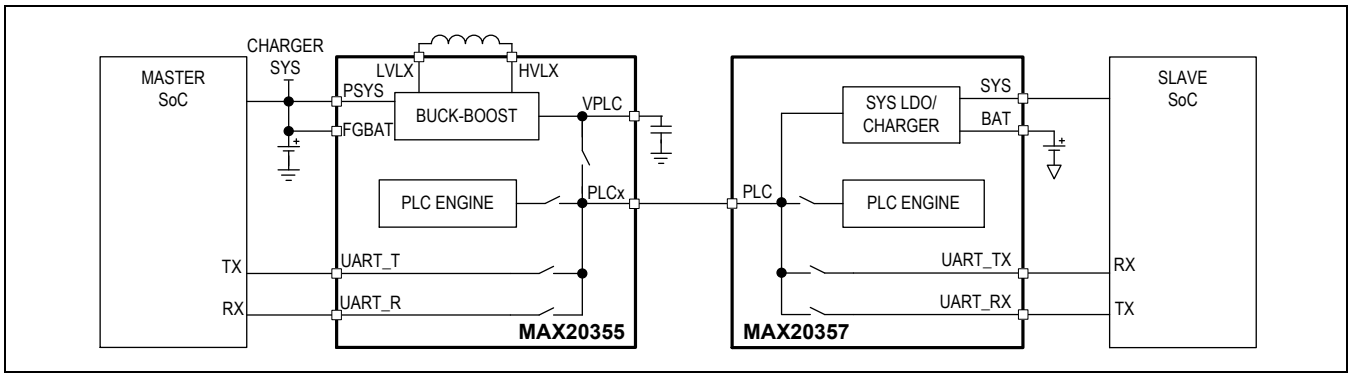


Figure 9. UART Interface

From Master to Slave

The master can put the slave into UART mode by using the UART_REQ master command. The direction of UART communication is set in the command argument. Corresponding master and slave switches are closed once the UART command is successfully triggered. The switch settings are shown in *Table 3*. If the user wants to control UART switches manually in UART mode, all the master/slave UART switch settings can be changed through I2C.

Table 3. UART Switch Settings in Master PLC Command Argument

slave_uart_sw[1:0]	MASTER UART_T	MASTER UART_R	SLAVE UART_TX	SLAVE UART_RX
0b00	×	×	×	×
0b01	×	√	×	√
0b10	√	×	√	×
0b11	√	√	√	√

"√" represents the switch is ON and "×" represents the switch is OFF

From Slave to Master

The slave can put the master into UART mode by using the UART_REQ slave command. Corresponding master switches are closed once the UART command is successfully triggered. The switch settings are shown in *Table 4*. No slave switch is closed after UART_REQ. The slave UART_TX and UART_RX switches need to be closed through the slave I2C write. If the user wants to control UART switches manually in UART mode, all the master/slave UART switch settings can be changed through the master/slave I2C.

Table 4. UART Switch Settings in Slave PLC Command Argument

master_uart_sw[3:0]	MASTER PLC2 UART_T	MASTER PLC2 UART_R	MASTER PLC1 UART_T	MASTER PLC1 UART_R
0b0000	×	×	×	×
0b0001	×	×	×	√
0b0010	×	×	√	×
0b0011	×	×	√	√
0b0100	×	√	×	×
0b1000	√	×	×	×
0b1100	√	√	×	×

"√" represents the switch is ON and "×" represents the switch is OFF

Exit UART Mode

There are two ways to quit UART mode: 1. Master/slave can quit UART mode by disabling UART mode and disconnecting the UART switch through the master/slave I2C; 2. Use master and slave UART timeout counter and wait for UART timeout interrupt. Enable timeout counter through tmo_tmr_ena1/2 on master and tmo_tmr_ena on slave before entering UART mode. Once the TX/RX line is idle for 0.5s, the slave turns off its UART switch, disconnect UART_TX or UART_RX from the PLC line and start a counter to count for 0.6s. After a 0.6s delay, the slave exits UART mode, connects the PLC line to the PLC engine, and returns to the PLC detection mode. Meanwhile, once the line is idle for 0.75s, the master turns off its UART switch, disconnects UART_T or UART_R from the PLC line, connects to the PLC engine, and returns to the

PLC detection mode. Note that once the slave disconnects its UART switch at time point 0.5s, it enters an automatic UART exit process and this process is irreversible. It is not allowed to send data from master through UART after time point 0.5s. If data resumes from the master between time point 0.5s and 0.75s, the master UART timeout counter is cleared, and it holds the master in UART mode for another 0.75s. This causes an issue because the slave exits UART mode at time point 1.1s and starts to send PLC detection pulse immediately. If master is still in UART mode after slave exits UART mode, slave's PLC detection pulse may hold master in UART mode.

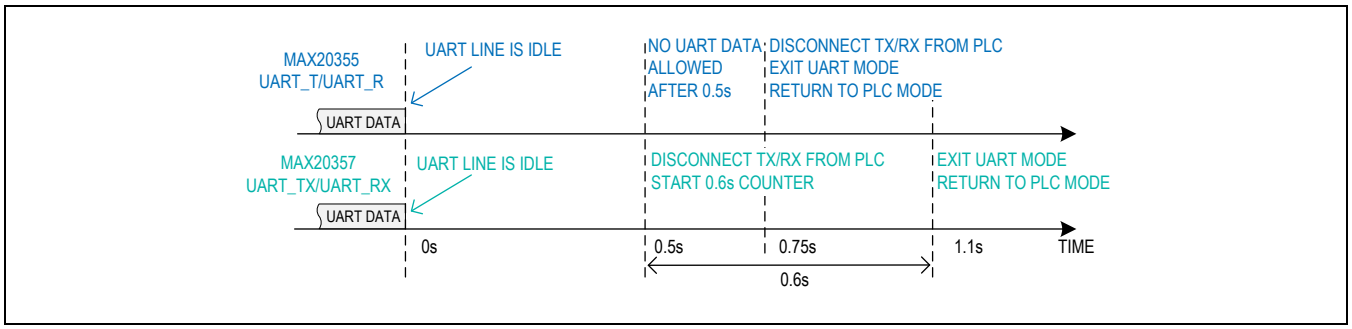


Figure 10. UART Automatic Exit Through UART Timeout

Autonomous Optimized Battery Charging

The MAX20355 interacts with MAX20357 to create an autonomous, closed-loop battery voltage tracking charging system. This system utilizes a dynamic voltage scaling (DVS) buck-boost on the MAX20355 in conjunction with an ultra-low dropout charger (ULDO) on the MAX20357. The PLC interface closes the loop allowing the buck-boost to adjust its output voltage to accommodate the minimum required headroom on the charger. This method allows for excellent 90% efficient energy transfer from the case battery to the earbud battery without the need to place bulky inductors on the tiny form factor ear buds. The excellent efficiency and extremely low heat generation of this charging system offer an increase in the number of charging cycles that can be supported by the case battery as well as the option to increase the rate of charge and deliver faster recharge times to the end customer.

Charger Battery Voltage Tracking Loop

Figure 11 shows the structure of the automatic charger battery voltage tracking algorithm. The charger battery voltage tracking algorithm is accomplished with data exchanges initiated by the master and responded to by the slave(s). Once the connection is built between master and slave(s), the master connects the buck-boost regulator to the PLC line and holds PLC voltage to bb_vlt__def (3.5V by default) for 100ms. Then pull PLC line to communication voltage (5.5V by default) and starts to PING. The communication voltage is programmable through bb_vlt_tran. Slave responds with ACK packet after master sends the first LISTEN command during PING. Slave's ACK packet contains information including slave GPIO status, battery information, charging status, and PLC voltage up/down request. Master adjusts buck-boost output based on the PLC voltage up/down request received from slave(s).

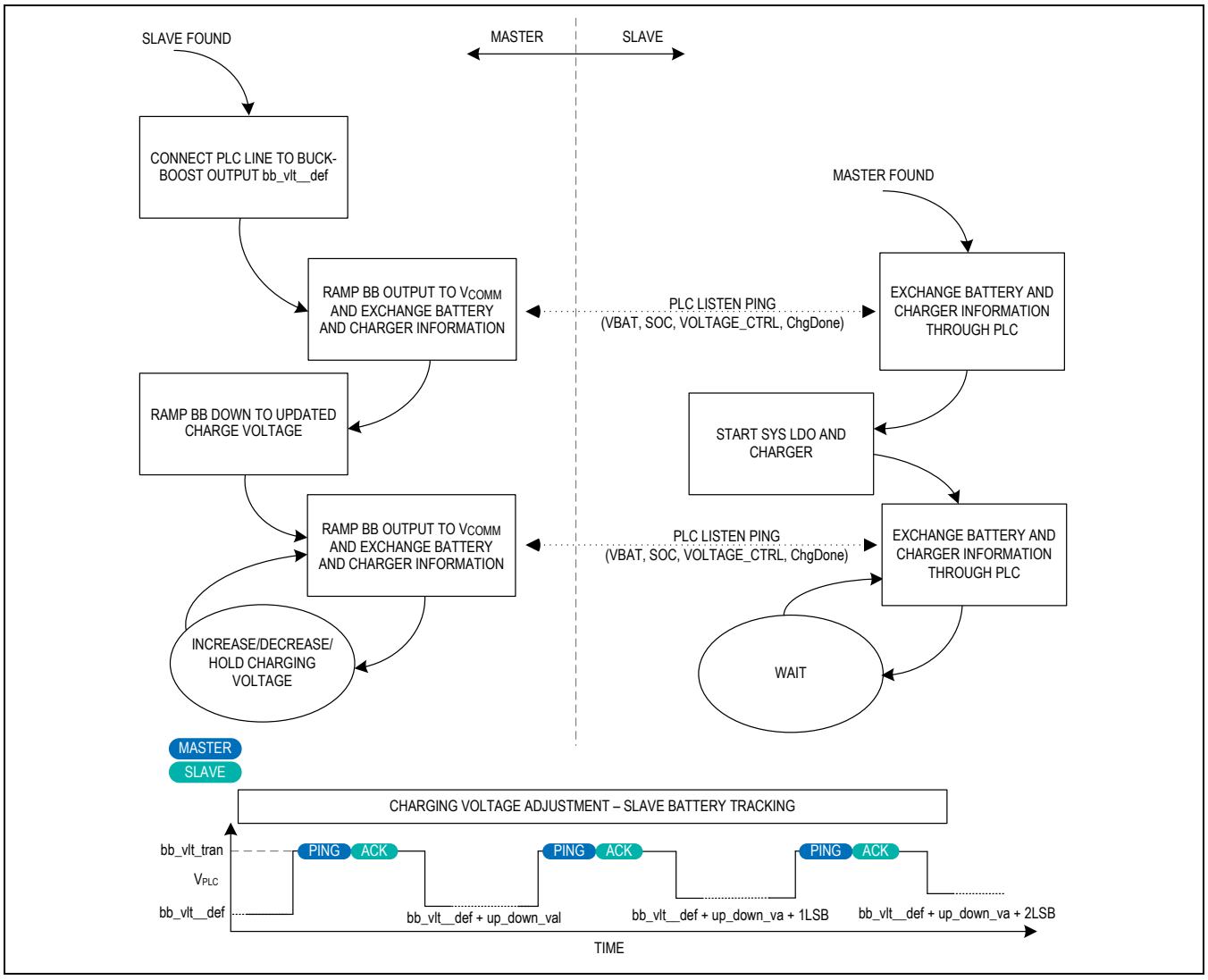


Figure 11. Polling Structure of Automatic Charging Algorithm

Slave compares battery voltage with PLC line voltage. If the PLC input voltage is above the programmed upper threshold (PLC voltage is in decrease zone), the slave sends a “decrease” signal during PING. Similarly, if PLC voltage is below the lower limit (increase zone) or within the limit (hold zone), it sends an “increase” or “hold” signal accordingly. The voltage thresholds are fully programmable through the parameters shown in *Figure 12*. The minimum step of voltage adjustment is programmable through I^2C register bits up_down_val.

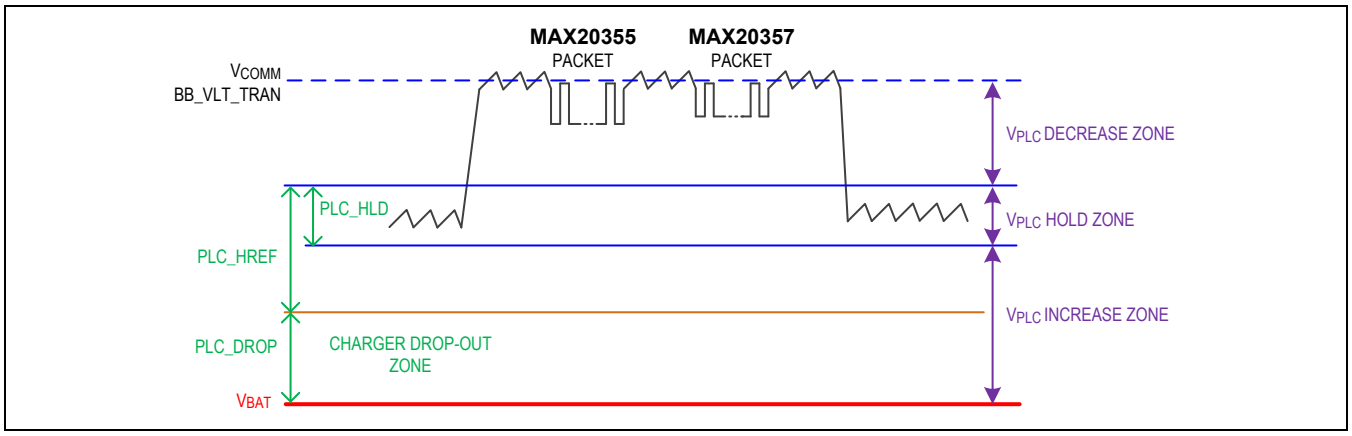


Figure 12. Slave Voltage Control Algorithm

Charger Voltage Tracking Algorithm

When the system works in dual slave mode, the two slaves attached at PLC1 and PLC2 are not guaranteed to have matching battery voltages. The automatic charging voltage adjustment algorithm offers multiple options for which battery should be given preference. System designers have three choices:

- **Minimum Voltage Tracking:** always target the lower of the two voltages. This approach maximizes efficiency since the voltage drop on the charger is minimized in all cases. The cost is that it increases the time it takes to charge the higher voltage battery since it cannot be charged until the two batteries reach the same voltage.
- **Maximum Voltage Tracking:** always target the higher of the two voltages. This approach sacrifices some efficiency in exchange for the fastest charge time for both batteries.
- **Optimized Battery Voltage Tracking:** target the lower battery until the batteries get within a certain voltage avg_t_delta of each other. avg_t_delta is programmable and can be tuned based on designer preferences. This algorithm offers a good balance between efficiency and charge time.

The three algorithm options are demonstrated in *Table 5* and *Table 6*. Note that the battery voltage mentioned above is the average battery voltage reported from the MAX20357 fuel gauge, which is read-only register AVGVCELL_byte_1 and AVGVCELL_byte_0 through I^2C, not real-time slave battery voltage.

Table 5. Maximum and Minimum Battery Voltage Tracking

MINIMUM TRACKING bb__alg_min	MAXIMUM TRACKING bb__alg_max	CHARGER1	CHARGER2	PLC CHARGE VOLTAGE VPLC
1	0	Hold	Hold	Hold
1	0	Hold	Decrease	Decrease
1	0	Hold	Increase	Hold
1	0	Decrease	X	Decrease
1	0	Increase	Hold	Hold
1	0	Increase	Decrease	Decrease
1	0	Increase	Increase	Increase
0	1	Hold	Hold	Hold
0	1	Hold	Decrease	Hold
0	1	Hold	Increase	Increase
0	1	Decrease	Hold	Hold
0	1	Decrease	Decrease	Decrease
0	1	Decrease	Increase	Increase
0	1	Increase	X	Increase

Table 6. Optimized Battery Voltage Tracking

$\|V_{BAT1}-V_{BAT2}\|$ < avg_t_delta	bb__alg_min	bb__alg_max	BATTERY TRACKING ALGORITHM
False	1	1	Minimum Tracking
False	0	0	Minimum Tracking
True	1	1	Maximum Tracking
True	0	0	Maximum Tracking

Mono-Slave and Dual-Slave Mode

The MAX20355 offers mono-slave mode to pair with one slave and dual-slave mode to pair with up to two slaves. In mono-slave mode, system designer needs to tie MAX20355 PLC1 pin and PLC2 pin to double the output current. By setting plc_double register to 1, MAX20355 can be configured to double the output overcurrent threshold up to 800mA. Note that the overcurrent detection sensors are located on PLC1 and PLC2, each one has overcurrent protection threshold up to 400mA (plc1/2_iprog). In order to achieve 800mA of total overcurrent threshold in mono-slave mode, the current flowing through PLC1 and PLC2 needs to be identical. Any non-zero impedance between PLC1 and PLC2 can cause current imbalance. Be sure to keep impedance imbalance between PLC1 and PLC2 pin within 20mΩ, which limits error on the overcurrent threshold within 10% (typ).

In MAX20357, by setting Ichg_x2 to 1, it doubles PLC current limit PLCCurr and charger current registers CC1IFChg/CC2IFChg. MAX20357 achieves up to 400mA charge current in mono-slave mode.

Due to the above reasons, the PLC master and slave devices does not support to toggle between mono-slave and dual-slave mode during operation. Once the system is configured to mono-slave mode, PLC current sink PLCSnkSel and PLC voltage threshold PLCThrSel on both MAX20355 and MAX20357 are doubled automatically to compensate for higher buck-boost output voltage ripple created by PLC1 and PLC2 tied together.

Integrated High Efficiency, Low Noise Buck-Boost Converter

The MAX20355 integrates a high-efficiency, low-noise buck-boost converter block that has been specifically optimized to operate without affecting PLC link performance. The function called dynamic voltage scaling (DVS) allows PLC voltage change with a controlled slew rate. The buck-boost block also provides protection features to limit peak inductor current and monitor temperature. The device supports up to 3.3W maximum power output.

After MAX20355 and MAX20357 build connection and start the charging process, MAX20355 needs to adjust its output voltage periodically to send the Listen command through the PLC telemetry PING. The MAX20355 buck-boost switching regulator outputs can be changed without restarting the regulator. This feature is called dynamic voltage scaling. Dynamic voltage scaling allows PLC voltage to switch between charge voltage and communication voltage with a controlled slew rate and tracks the MAX20357 battery with a 13.7mV increment step. The buck-boost ramp control-related registers are already optimized. It is highly recommended to keep related registers as default.

Buck-boost regulator naturally provides peak current limiting feature since peak current control is used. When the peak current limit is hit, the buck-boost regulator limits the current and lets output voltage on the VPLC pin drop. If load keeps increasing, eventually buck-boost regulator turns off after VPLC pin voltage drops below UVLO threshold. Use bb___l2uh to choose between 1µH and 2.2µH inductor. Roughly half of the ripple is allowed for 2.2µH inductor compared to 1µH inductor to maintain the same amount of core loss. The maximum inductor peak current in boost mode is 2.4A for 2.2µH and 2.75A for 1µH. Note that the MAX20355 has overcurrent protection on PLC1/2 and the PLC overcurrent protection thresholds are lower than buck-boost peak current control. It is not likely buck-boost starts to limit inductor current before PLC overcurrent protection triggers. Buck-boost peak current control provides a second protection for the system.

Buck-boost block also integrated a die temperature sensor. If the sensed temperature is higher than 160°C, MAX20355 turns off the buck-boost converter and set the bb_fault and thm_flt_sts status register to 1 and the corresponding interrupt bits.

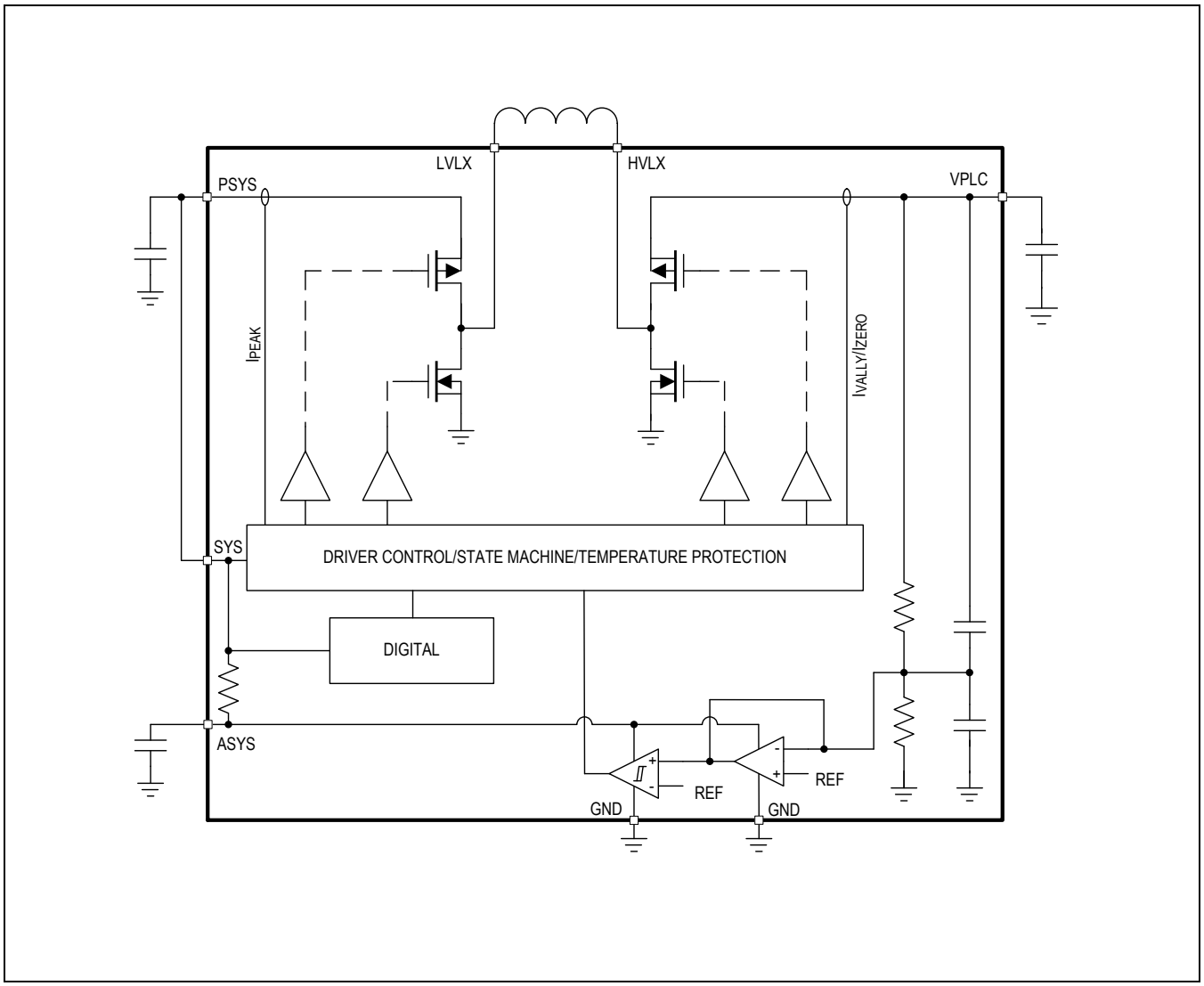


Figure 13. Buck-Boost Converter Block Diagram

PLC Operation Modes

The MAX20355 features multiple modes of operation that are designed to minimize power consumption in the end user's application. These include a low power OFF mode, PLC detection mode, and PLC link mode. The transitions between modes and general operation are shown in the diagram below.

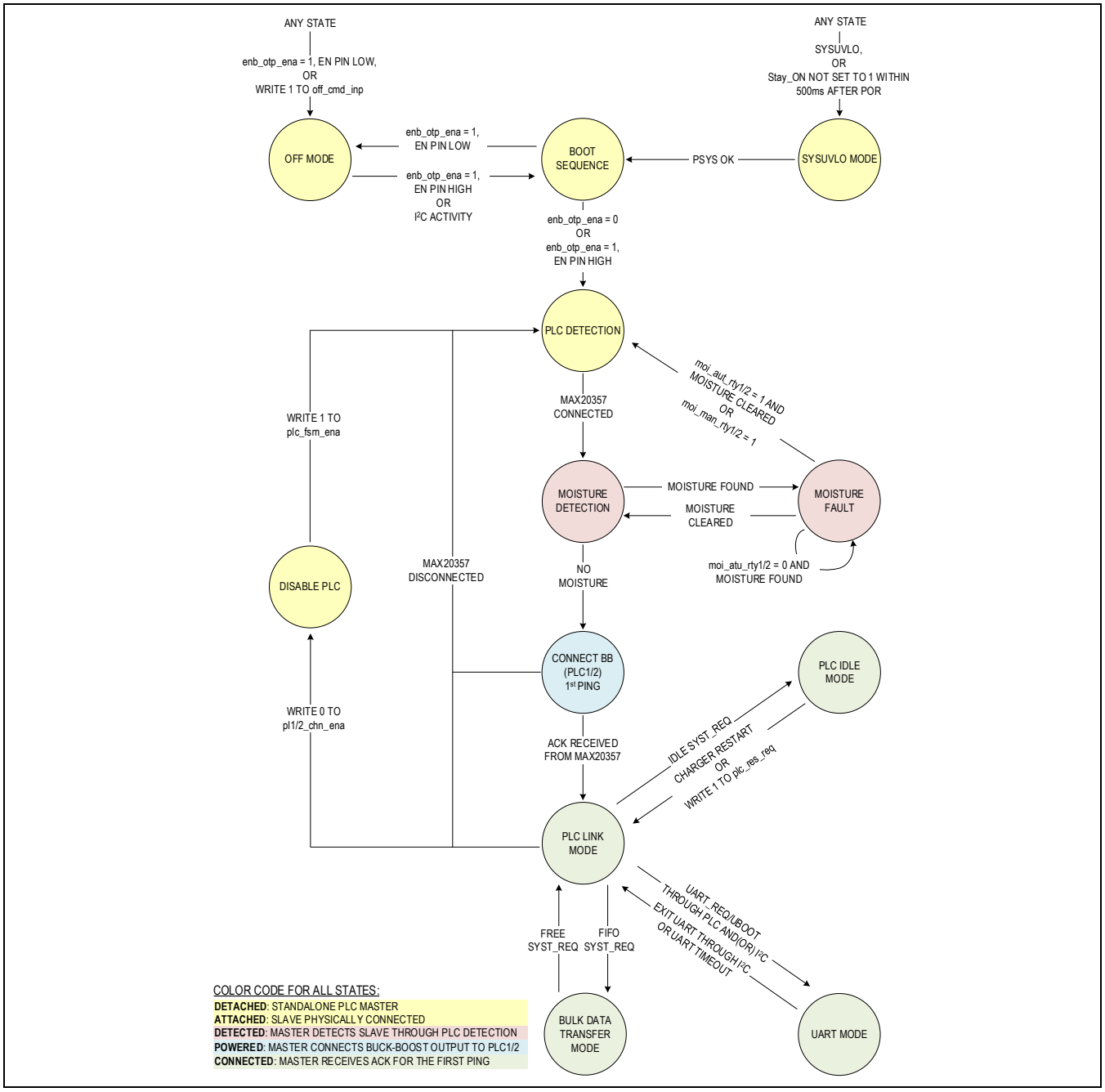


Figure 14. MAX20355 Operating Modes

OFF Mode

The MAX20355 features an OFF mode that is entered by driving the EN low if EN pin function is enabled (enb_otp_ena = 1, default condition), or by issuing OFF mode request through I2C (write 1 to off_cmd_inp). The supply current in OFF mode is reduced to 0.1µA, I2C data is retained, the fuel gauge is shut down, and all blocks are placed into a low-power mode. The device exits OFF mode when the EN pin is driven high in case enb_otp_ena = 1 or the device detects I2C activity. OTP registers are reloaded after the device exits OFF mode. Note that enb_otp_ena is also reloaded to its default OTP setting after the device exits OFF mode.

PLC Detection Mode

Standalone master or slave operates in PLC detection mode when the device is active. The PLC and moisture detection block diagram is shown in *Figure 15*. In PLC detection mode, master sends PLC detection pulse through an internal pullup/pulldown resistor every 380ms. Slave sends PLC detection pulse through a similar pullup/pulldown resistor every 240ms. Master PLC is active, fuel gauge is on, and other blocks are in low-power mode; slave PLC is active, fuel gauge is on, power path FET is fully on, and other blocks are in low-power state. Both the master and slave contain a passive impedance clamp network, which allows the PLC device to detect its counterpart device even when the counterpart device has a dead battery. See the following sections for a detailed description of the master and slave connection detection scheme under various circumstances. PLC detection pulse or moisture measurement launched by the counterpart device is shown in a dashed line.

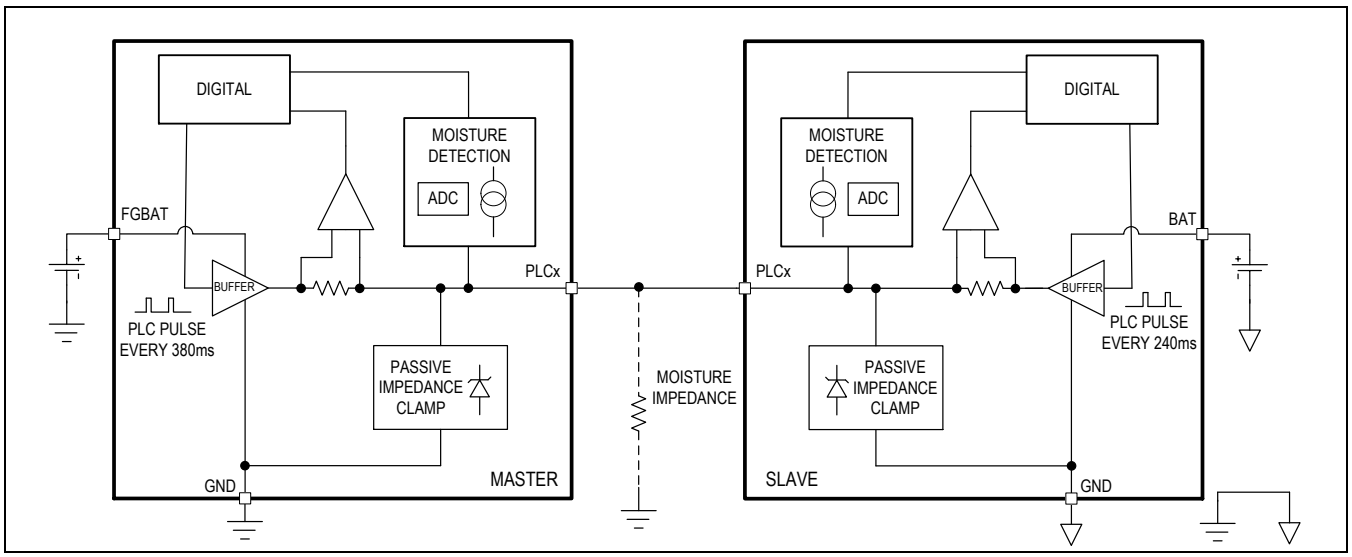


Figure 15. PLC and Moisture Detection Block Diagrams

Active Master and Active Slave

As shown in *Figure 16*, both the master and slave are in the PLC detection mode before the slave attached. After receiving slave's PLC detection pulse on the PLC line, the master sends a signature pulse (3x pulse) and automatically launches a moisture measurement. If there is no moisture, master connects the PLC line to a buck-boost output and holds PLC voltage to default charge voltage (default 3.5V) for 100ms. Then master increases PLC voltage to communication voltage (default 5.5V) and PING the slave. If master received ACK from slave, both devices enter PLC Link mode. After the master and slave build connection, ch1/2_con_sts status bit of MAX20355 and chn_con_sts status bit of MAX20357 are set, the corresponding interrupt bits are asserted accordingly.

If there is moisture found, the master does not turn on buck-boost to supply PLC1/2 pin and no power is applied to the slave. Moisture-related status and interrupt registers in MAX20355 are set indicating moisture detected. If the slave does not detect power applied through the PLC line within 200ms, it launches moisture measurement automatically. If moisture is not cleared, the slave stays in moisture detection mode and launches automatic moisture detection every 16s. Once the slave detects moisture is cleared, it proceeds to PLC detection mode to connect with the master. See the *Moisture Detection* section for a detailed description of the moisture feature.

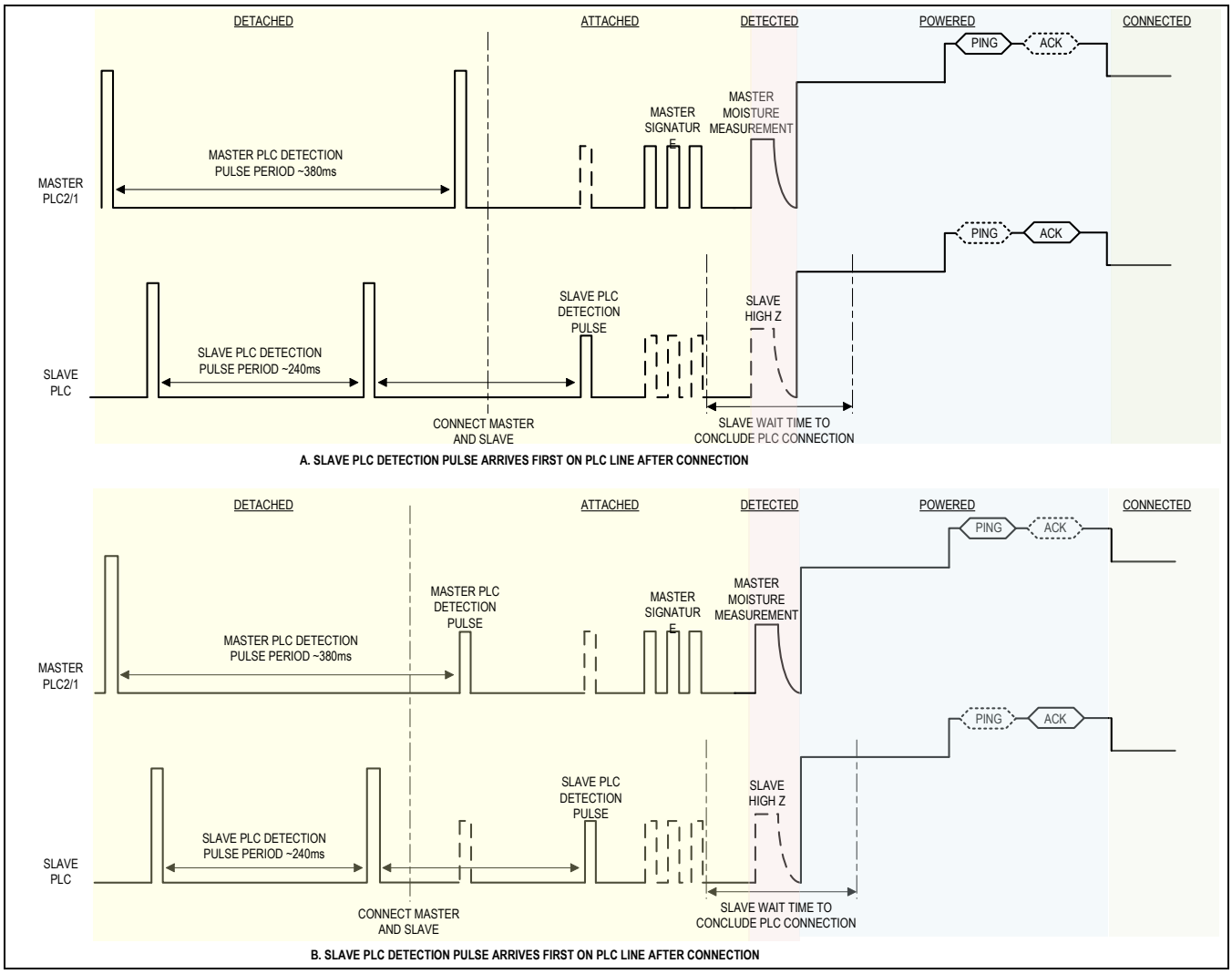


Figure 16. Active Master and Active Slave Connection

Active Master and Dead Slave

Both master and slave offer passive impedance clamp networks for dead slave battery detection and moisture detection. As shown in *Figure 17*, before connection, master is in PLC detection mode and slave is connected to a dead battery. After the slave is attached to the master, the amplitude of the master's PLC detection pulse is clamped to 1.8V by slave's internal passive impedance clamp network. Master perceives its PLC detection pulse with reduced amplitude as 'something connected'. After 3x consecutive 'something connected' pulse, the master launches moisture measurement automatically. If there is no moisture, the master connects the PLC line to the buck-boost output and holds the PLC voltage to default charge voltage (default 3.5V) for 100ms. Then the master increases PLC voltage to communication voltage (default 5.5V) and PING the slave. If master received ACK from slave, both devices enter the PLC Link mode. After the master and slave build connection, ch1/2_con_sts status bit of MAX20355 and chn_con_sts status bit of MAX20357 are set, the corresponding interrupt bits are asserted accordingly.

Note that not only does the passive impedance clamp on slave creates a 'something connected' pulse on the master, but moisture between the PLC line and the ground can also create similar behavior. Master is not able to determine if it is real slave or moisture impedance before launching moisture detection. If there is moisture found, master does not connect the buck-boost output to PLC1/2 pin, and power is not applied to the slave. Moisture-related status and interrupt registers in MAX20355 are set indicating moisture detected. MAX20357 remains in SYSUVLO mode.

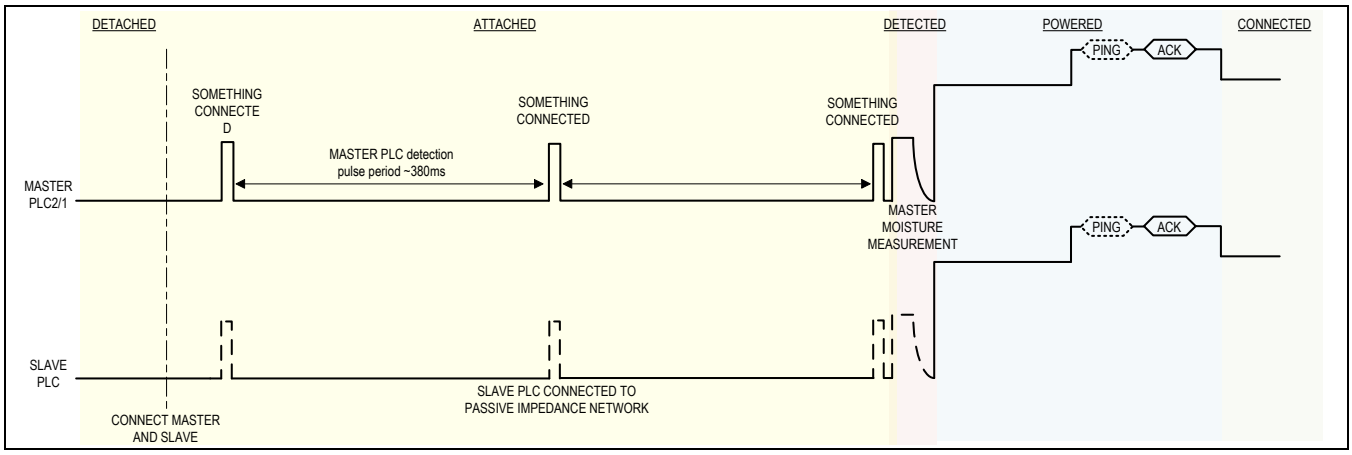


Figure 17. Active Master and Dead Slave Connection

Dead Master and Active Slave

As shown in *Figure 18* before connection, the slave is in PLC Detection mode and master is connected to a dead battery. After slave is attached to the master, the slave's PLC detection pulse is clamped to 1.8V by master's passive impedance clamp network. The slave perceives its PLC detection pulse with reduced amplitude as 'something connected'. After 3x consecutive 'something connected' pulse, dead_found interrupt is asserted in MAX20357 signaling dead master is detected, MAX20357 automatically launches a moisture measurement and continues sending PLC detection pulse. After master wakes up, master and slave follow the procedure described in 'Active Master and Active Slave' to build a connection.

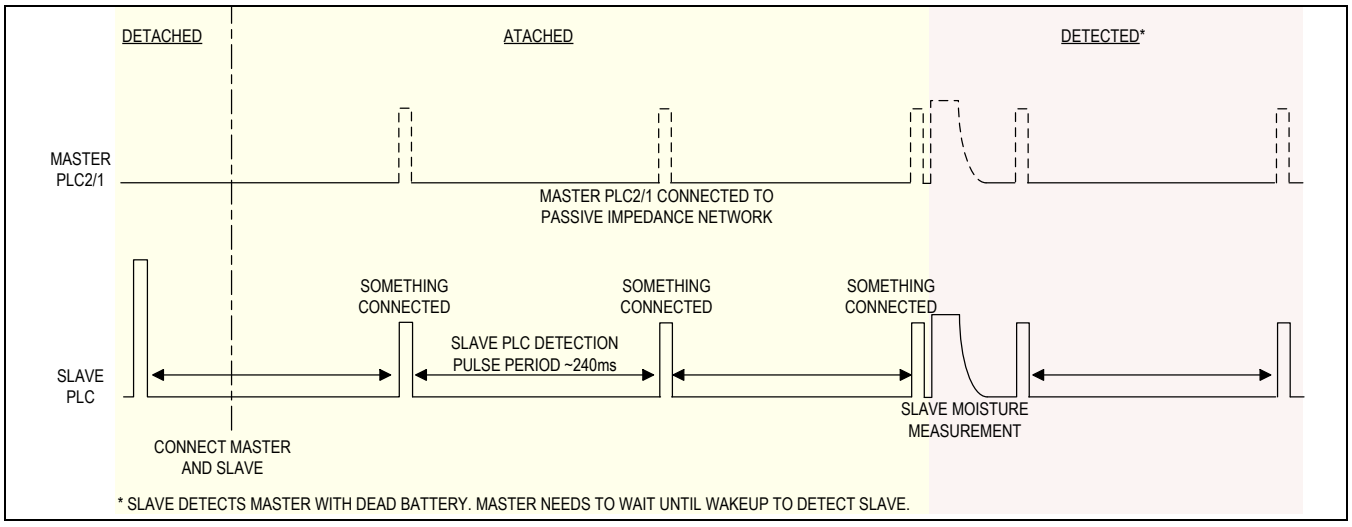


Figure 18. Dead Master and Active Slave Connection

Moisture Detection

To prevent corrosion of the contacts if moisture is present on the PLC outputs, both the MAX20355 and MAX20357 offer a build-in moisture detection feature to check the resistance from the PLC output to GND. If the impedance detected is less than the I^2C programmable moisture threshold, corresponding interrupts are flagged on both devices. The MAX20355 does not connect the buck-boost output to the PLC lines to start the charging process whenever moisture is detected. For both MAX20355 and MAX20357, moisture target current is set by RaccDetMIp (default 1µA) and voltage threshold is set by RaccDetMThr (default 700mV). The moisture threshold is then calculated as RaccDetMThr/RaccDetMIp (default 700kΩ). Both the MAX20355 and MAX20357 can launch moisture detection before the PLC line is connected to the buck-boost output. After MAX20355 and MAX20357 build the PLC connection and enters the PLC link mode, manual moisture request is queued and proceeded after the PLC disconnection event. Note that after MAX20355 and MAX20357 build connection and starts charging, it is not able to measure moisture by applying source current and measuring voltage from ADC, since PLC voltage is already regulated by the buck-boost converter. The MAX20355 offers overcurrent protection

for this use case to maintain system safety. The detailed description is in the overcurrent protection section. After POR, if no moisture is found previously, MAX20355 launches moisture detection under the following conditions:

1. After MAX20355 detects PLC pulse from the MAX20357 and sends signature pulse.
2. After MAX20355 detects 3x consecutive 'something connected' pulse. If MAX20355 detects an equivalent resistance less than 17kΩ (typ) applied between PLC line and ground, the amplitude of PLC detection pulse reduces and MAX20355 recognizes it as 'something connected' pulse.
3. After manual moisture detection request through I2C bit moi_man_pl1/2. Manuel moisture request is queued when buck-boost voltage is applied.
4. Automatic moisture detection every 16s if it is enabled through I2C bit moi_det_aut1/2. By default, moi_det_aut1/2 is set to 0 and automatic moisture detection is disabled.

After POR, if no moisture is found previously, the MAX20357 launches moisture detection under the following conditions:

1. After MAX20357 detects 3x consecutive 'something connected' pulse. If MAX20357 detects an equivalent resistance less than 17kΩ (typ) applied between PLC line to ground, the amplitude of PLC detection pulse reduces and MAX20357 recognizes it as 'something connected' pulse.
2. After manual moisture detection request through I2C bit moi_man_pl. Manual moisture request is queued when buck-boost voltage is applied.
3. Automatic moisture detection every 16s. Automatic moisture detection is enabled by default.

During moisture detection, MAX20355 and MAX20357 source a current through PLC pin and measure PLC voltage with built-in ADC. Moisture detection logic always starts from 1µA source current. If the ADC voltage reading RaccDetMThr is less than 0x3F – AdcRng, it increases current to 4 times and repeats the same measurement until the measured value is larger than 0x3F – AdcRng or maximum source current 64µA is reached. The moisture measurement algorithm ensures that the target moisture sink current RaccDetMIp can be reached for the best accuracy. The ADC voltage readings are reported in ADCAvg and source current is reported in IP_RES_DET. Moisture resistance is calculated as ADCAvg/IP_RES_DET. Once the moisture measurement is completed, based on the measured result, corresponding interrupts are asserted as shown in *Figure 18*. Note that the designed chip-level impedance measurement error is ±10%. However, the actual measurement error in the application can be higher than ±10% due to leakage current on PCB. Especially when moisture impedance is approaching 1.5MΩ, the leakage current is comparable to ADC source current. Since the leakage current varies depending on application schematic and layout design, the actual measurement error also varies from design to design.

Table 7. Moisture Interrupt

MOISTURE INTERRUPT	MOISTURE RESISTANCE
res_det_gnd	R_{MOI} < 458Ω ± 92Ω
moi_irq_det	R_{MOI} < RaccDetMThr / RaccDetMIp (default 700kΩ)
moi_dne_int	RaccDetMThr / RaccDetMIp (default 700kΩ) < R_{MOI} < 1.5MΩ
res_det_opn	R_{MOI} > 1.5MΩ
res_det_abr	Significant variation during moisture measurement
plc2/1_moi_det (MAX20355)	R_{MOI} < RaccDetMThr / RaccDetMIp (default 700kΩ) Set by automatic moisture detection conditions 1, 2 and 4
plc_moi_det (MAX20357)	R_{MOI} < RaccDetMThr / RaccDetMIp (default 700kΩ) Set by automatic moisture detection conditions 1 and 3

If moisture is detected, the MAX20355/MAX20357 continues sending PLC detection pulse by default to check resistance on the PLC line through its internal pullup resistor. If the equivalent moisture resistance measured is larger than 17kΩ, MAX20355 and MAX20357 perceive it as moisture removed and clears corresponding interrupts. Note that even moisture is detected in one standalone PLC device, if it is attached to its counterpart device, they are still able to pass the ATTACHED phase of the PLC connection detection process. After entering the DETECTED phase, MAX20355 launches moisture detection and fails to build a connection with the MAX20357. The system cycles between ATTACHED and DETECTED back and forth until moisture condition is removed. Automatic moisture retry on MAX20355 can be disabled by setting moi_aut_rty1/2 = 0. Automatic retry on MAX20357 cannot be disabled.

PLC IDLE Mode

Once the MAX20357 battery is fully charged, either MAX20355 or MAX20357 can put the system into PLC IDLE mode by IDLE SYST_REQ PLC command to reduce unnecessary power losses. The current consumption of the MAX20357 in PLC IDLE mode is reduced to 11µA. In PLC IDLE mode, master and slave suspend charging by removing power from the PLC line and starts handshaking pulsing protocol to check the presence of the other device. Both master and slave can request to resume PLC communication from PLC IDLE mode by writing 1 to plc_res_req through I2C. The system automatically resumes from PLC IDLE mode if MAX20357 charger auto-restart is enabled and MAX20357 battery voltage falls below charger restart threshold. *Figure 19* shows the waveform of MAX20355 issuing IDLE SYST_REQ PLC command to put the system into PLC IDLE mode.

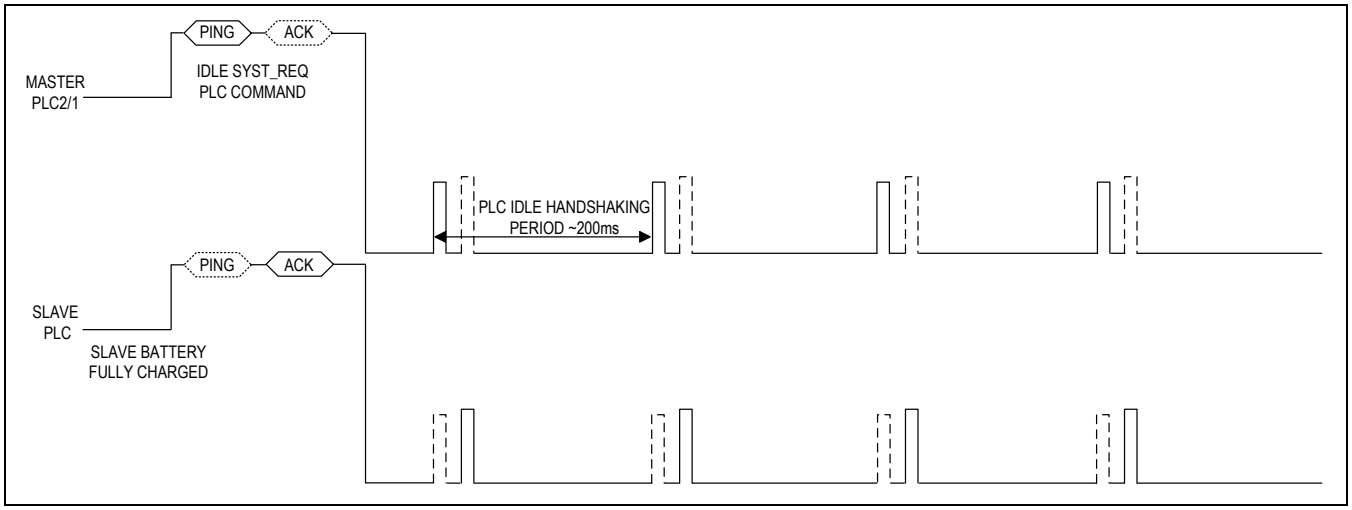


Figure 19. Enter PLC IDLE Mode After Slave Battery is Fully Charged.

Both MAX20355 and MAX20357 could detect the PLC disconnection (earbud removal) while in PLC IDLE mode, as shown in *Figure 20*.

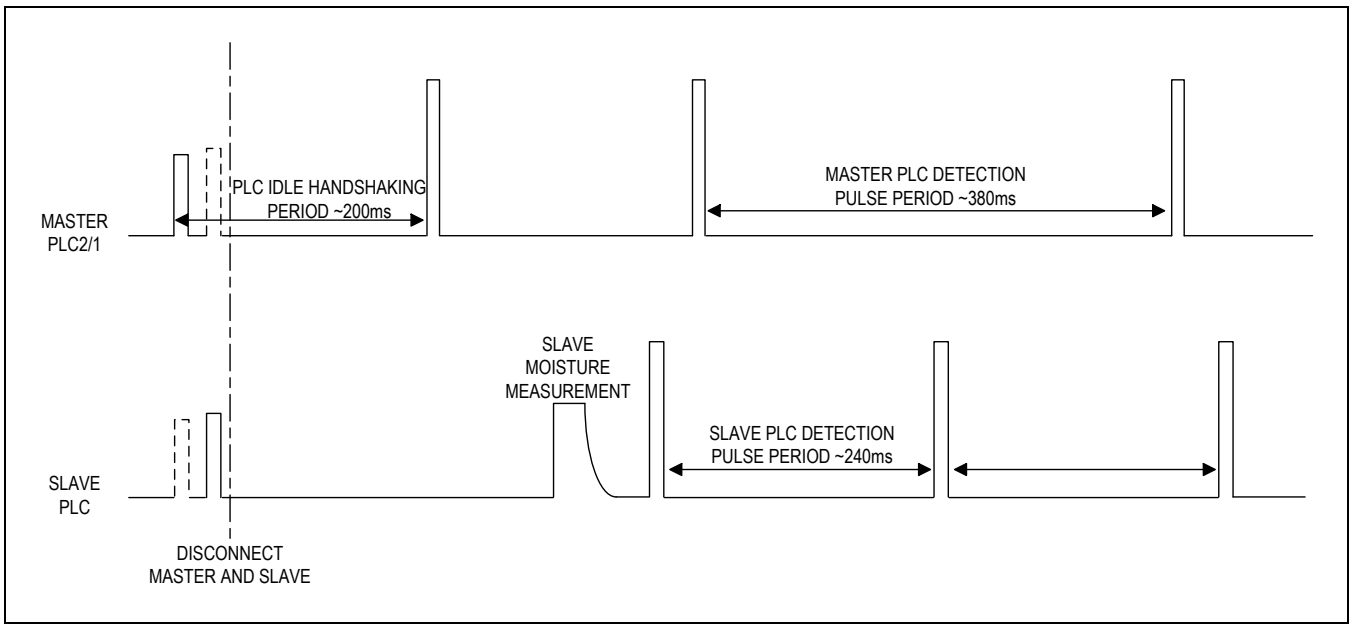


Figure 20. Master and Slave Disconnection While in PLC IDLE Mode

Master/Slave Resets and UBOOT Mode

The MAX20357 has PLC and I2C controllable resets. All the resets can be performed by local I2C write or by PLC command from its counterpart device. The MAX20355/MAX20357 performs reset for its counterpart device through SYST_REQ PLC command. See *Table 1* and *Table 2* for a detailed decode chart for reset commands. Different reset types are summarized in *Table 8*.

Table 8. Reset Types and Description

	DESCRIPTION	MASTER		SLAVE	
		PLC*	I2C	PLC*	I2C
Fuel Gauge Reset	Reset fuel gauge register	SYST_REQ FG reset	fg_ena_byp fg_ena_val	SYST_REQ FG reset	fg_ena_byp fg_ena_val
Soft Reset	Reset internal registers and FSMs. Only PLC, not including RAM or fuel gauge.	SYST_REQ Soft reset	soft_reset	SYST_REQ Soft reset	soft_reset
Hard Reset	Cycle the power at MAX20357 SYS pin. MAX20357 charger is temporarily turned off.	SYST_REQ Hard reset	—	—	hard_reset
Hard + Soft Reset	Soft reset + Hard reset.	SYST_REQ X reset	—	—	sft_hrd_rst
UBOOT	Hard reset and enters UART mode.	SYST_REQ UBOOT	—	—	uboot_i2c_cmd
Off Mode	Off mode	—	off_cmd_inp	—	off_cmd_inp
Seal Mode	MAX20357 Seal mode	SYST_REQ SEAL	—	—	seal_i2c_cmd

**PLC column lists the PLC commands that put its counterpart into the corresponding reset state*

General Purpose Input Output (GPIO)

MAX20355/MAX20357each features four general purpose input/outputs (GPIO) controllable by its counterpart through PLC or local I2C write. Set GPIOPLCCtrx = 1 to control GPIOx by PLC, set GPIOPLCCtrx = 0 to control GPIO by I2C. When GPIOx is controlled by I2C, set GPIOEnResx = 1 to configure GPIOx as general-purpose input, set GPIOEnResx = 0 to configure GPIOx as general-purpose output.

When GPIOx is configured as general-purpose input (GPI), set GPIOEnPupx = 1 to have the GPIOx internally pulled up to the maximum of PLC line voltage and battery voltage, set GPIOEnPupx = 0 to have the GPIOx internally pulled down.

When GPIOx is configured as a general-purpose output (GPO), the GPO is in open-drain mode and requires an external pullup resistor (typically 10kΩ–100kΩ). Connect the external pullup resistor to a voltage rail that is higher than 1.4V (TYP). Set GPIODoutx = 1 to turn on the open-drain FET and output low, set GPIODoutx = 0 to turn off the open-drain FET and output high.

The GPI input status GPIODAInp4–GPIODAInp1 still functions properly and does not collide when the GPIO is configured as an output. In other words, GPIO status, either configured as input or as output, is stored in GPIODAInp4–GPIOAInp1.

Integrated Protection Features

The MAX20355 features integrated protection features on the PLC pins to protect the device from short-circuit faults and ESD events.

Current Limited PLC Outputs

The PLC outputs of the MAX20355 feature a current limit block to protect the device from damage in the event of a hard short to ground, or to protect the battery from unwanted discharge in the case of a soft short. The current limit is programmable from 100mA–400mA through plc1/2_iprog, with an accuracy of ±20%. When the current limit is exceeded, MAX20355 disconnects buck-boost from PLC pins, and sys_err_int status and sys_err_intIrq interrupt are flagged.

High ESD Protected ESD Outputs

High-ESD protection on the PLC outputs of MAX20355 protect the device from high energy ESD damage up to 8kV contact discharge.

ModelGauge M5 EZ Fuel Gauge with Integrated Sense Resistor

The MAX20355 and MAX20357 implement the Maxim ModelGauge m5 algorithm. The IC measures voltage, current, and temperature accurately to produce fuel gauge results. The ModelGauge m5 robust algorithm provides tolerance against battery diversity. This additional robustness enables simpler implementation for most applications and batteries by avoiding time-consuming battery characterization.

The ModelGauge m5 algorithm combines the short-term accuracy and linearity of a coulomb-counter with the long-term stability of a voltage-based fuel gauge, along with temperature compensation to provide industry-leading fuel gauge accuracy. The IC automatically compensates for aging, temperature, and discharge rate and provides an accurate state of charge (SOC) in percentage (%) and remaining capacity in milliampere-hours (mAhr) over a wide range of operating conditions. Fuel gauge error always converges to 0% as the cell approaches empty. The IC provides accurate estimation of time-to-empty and time-to-full and provides three methods for reporting the age of the battery: reduction in capacity, increase in battery resistance, and cycle odometer.

The IC contains a unique serial number. It can be used for cloud-based authentication. See the *Serial Number Feature* section for more information.

Communication to the host occurs over the standard I^2C interface.

ModelGauge m5 EZ Performance

ModelGauge m5 EZ performance provides plug-and-play operation when the IC is connected to most lithium batteries. While the IC can be custom-tuned to the application's specific battery through a characterization process for ideal performance, the IC can provide good performance for most applications with no custom characterization required. *Table 9* and *Figure 21* show the performance of the ModelGauge m5 algorithm in applications using the ModelGauge m5 EZ configuration.

The ModelGauge m5 EZ provides good performance for most cell types. For some chemistries, such as lithium-iron-phosphate ($LiFePO_4$) and Panasonic NCR/NCA series cells, it is suggested that the customer request a custom model from Maxim for best performance.

For even better fuel-gauging accuracy than ModelGauge m5 EZ, contact Maxim for information regarding cell characterization.

Table 9. ModelGauge m5 EZ Performance

DESCRIPTION	AFTER FIRST CYCLE* (%)	AFTER SECOND CYCLE* (%)
Tests with less than 3% error	95	97
Tests with less than 5% error	98.7	99
Tests with less than 10% error	100	100

**Test conditions: +20°C and +40°C, run time of > 3 hours.*

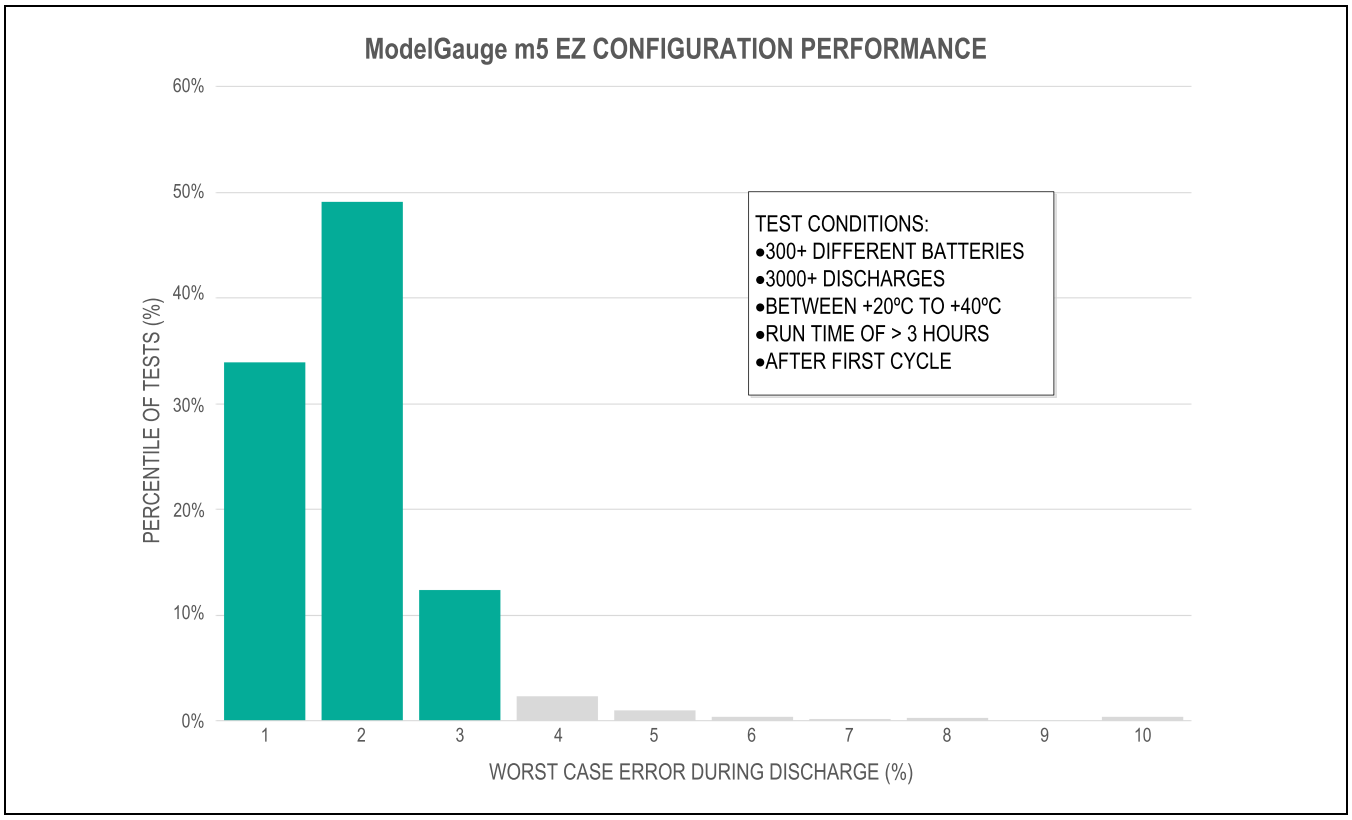


Figure 21. ModelGauge m5 EZ Configuration Performance

Application Notes

Refer to the following application notes for additional reference material:

User Guide 6597: MAX1726x ModelGauge m5 EZ User Guide

- Documents full register set
- More details about ModelGauge m5 algorithm
- Discusses additional applications

User Guide 6595: MAX1726x Software Implementation Guide

- Guidelines for software drivers including example code

Standard Register Formats

Unless otherwise stated during a given register's description, all fuel gauge registers of the MAX77658 follow the same format depending on the type of register. See *Table 10* for the resolution and range of any register described hereafter.

Table 10. ModelGauge m5 Register Standard Resolutions

REGISTER TYPE	LSB SIZE	MINIMUM VALUE	MAXIMUM VALUE	NOTES
Capacity	0.33mAh	0.0mAh	21845mAh	
Percentage	1/256%	0.0%	255.9961%	1% LSb when reading only the upper byte.
Voltage	1.25mV/16	0.0V	5.11992V	
Current	104.17µA	-3.41333A	3.41322A	Signed two's-complement format.
Temperature	1/256°C	-128.0°C	127.996°C	Signed two's-complement format. 1°C LSb when reading only the upper byte.
Resistance	1/4096Ω	0.0Ω	15.99976Ω	
Time	5.625s	0.0s	102.3984h	
Special				Format details are included with the register description.

ModelGauge m5 Algorithm

Classical coulomb-counter-based fuel gauges have excellent linearity and short-term performance. However, they suffer from drift due to the accumulation of the offset error in the current-sense measurement. Although the offset error is often very small, it cannot be eliminated. It causes the reported capacity error to increase over time and requires periodic corrections. Corrections are traditionally performed at full or empty. Some other systems also use the relaxed battery voltage to perform corrections. These systems determine the true state-of-charge (SOC) based on the battery voltage after a long time of no current flow. Both have the same limitation: if the correction condition is not observed over time in the actual application, the error in the system is boundless. The performance of classic coulomb counters is dominated by the accuracy of such corrections. Voltage measurement-based SOC estimation has accuracy limitations due to imperfect cell modeling but does not accumulate offset error over time.

The MAX20355/MAX20357 includes an advanced voltage fuel gauge (VFG) that estimates open-circuit voltage (OCV), even during current flow, and simulates the nonlinear internal dynamics of a Li+ battery to determine the SOC with improved accuracy. The model considers the time effects of a battery caused by the chemical reactions and impedance in the battery to determine SOC. This SOC estimation does not accumulate offset error over time. The IC performs a smart empty compensation algorithm that automatically compensates for the effect of temperature condition and load condition to provide accurate state-of-charge information. The converge-to-empty function eliminates error toward an empty state. The IC learns battery capacity over time automatically to improve long-term performance. The age information of the battery is available in the output registers.

The ModelGauge m5 algorithm combines a high-accuracy coulomb counter with a VFG. See *Figure 22*. The complementary combined result eliminates the weaknesses of both the coulomb counter and the VFG while providing the strengths of both. A mixing algorithm combines the VFG capacity with the coulomb counter and weighs each result so that both are used optimally to determine the battery state. In this way, the VFG capacity result is used to continuously make small adjustments to the battery state, cancelling the coulomb-counter drift.

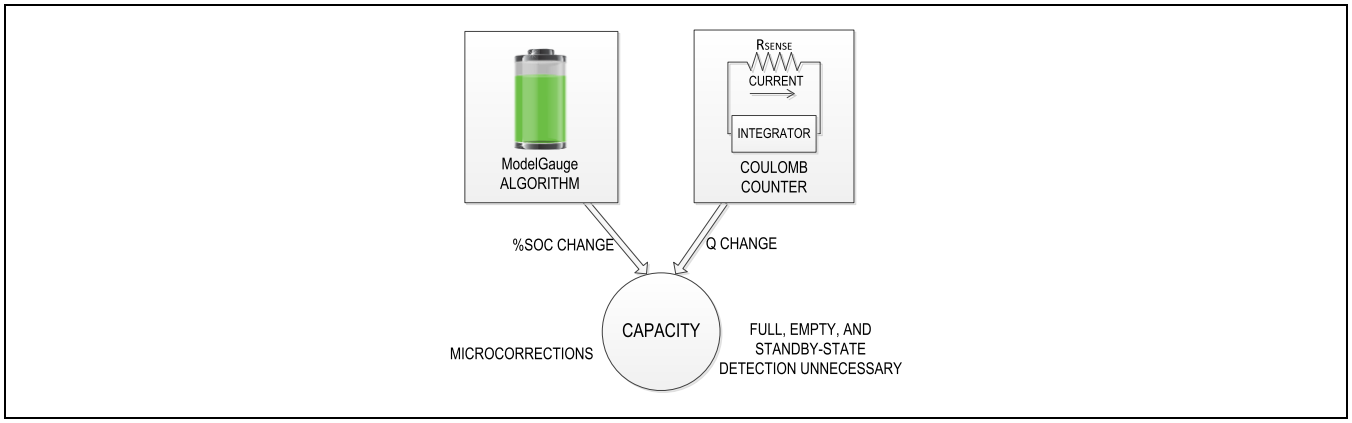


Figure 22. ModelGauge m5 Algorithm

The ModelGauge m5 algorithm uses this battery state information and accounts for temperature, battery current, age, and application parameters to determine the remaining capacity available to the system. As the battery approaches the critical region near empty, the ModelGauge m5 algorithm invokes a special error correction mechanism that eliminates any error.

The ModelGauge m5 algorithm continually adapts to the cell and application through independent learning routines. As the cell ages, its change in capacity is monitored and updated and the voltage-fuel-gauge dynamics adapt based on cell-voltage behavior in the application.

Analog Measurements

Voltage Measurement

VCell Register (0x09)

Register Type: Voltage
VCell reports the voltage measured between BATT and GND

AvgVCell Register (0x19)

Register Type: Voltage
The AvgVCell register reports an average of the VCell register readings.

MaxMinVolt Register (0x1B)

Register Type: Special

Initial Value: 0x00FF

The MaxMinVolt register maintains the maximum and minimum of VCell register values since the device reset. At power-up, the maximum voltage value is set to 0x00 (the minimum) and the minimum voltage value is set to 0xFF (the maximum). Therefore, both values are changed to the voltage register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x00FF. The maximum and minimum voltages are each stored as 8-bit values with a 20mV resolution.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MaxVCELL								MinVCELL							

MaxVCELL: Maximum VCell register reading

MinVCELL: Minimum VCell register reading

Current Measurement

Current Register (0x0A)

Register Type: Current

The MAX20355/MAX20357 uses internal current sensing to monitor the current through the SYS FG pin. The measurement value is stored in two's-complement format. Measurement that exceeds maximum and minimum current range is stored as maximum and minimum values. The current register has a LSB value of 31.25µA, a register scale range of ±1.024A, and an allowable measurement range as described in the *Absolute Maximum Ratings*.

AvgCurrent Register (0x0B)

Register Type: Current

The AvgCurrent register reports an average of current register readings.

MaxMinCurr Register (0x1C)

Register Type: Special

Initial Value: 0x807F

The MaxMinCurr register maintains the maximum and minimum current register values since the last IC reset or until cleared by host software. At power-up, the maximum current value is set to (most negative) and the minimum current value is set to 7Fh (most positive). Therefore, both values are changed to the current register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x807F. The maximum and minimum currents are each stored as two's complement 8-bit values with 160mA resolution.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MaxCurrent								MinCurrent							

MaxCurrent: Maximum Current register reading

MinCurrent: Minimum Current register reading

Temperature Measurement

Temp Register (0x08)

Register Type: Temperature
The Temp register provides the temperature measured by the thermistor or die temperature based on the Conifg register setting.

MaxMinTemp Register (0x1A)

Register Type: Special Initial Value: 0x807F

The MaxMinTemp register maintains the maximum and minimum Temp register (0x08) values since the last fuel-gauge reset or until cleared by host software. At power-up, the maximum value is set to 0x80 (most negative) and the minimum value is set to 0x7F (most positive). Therefore, both values are changed to the Temp register reading after the first update. Host software can reset this register by writing it to its power-up value of 0x807F. The maximum and minimum temperatures are each stored as two's complement 8-bit values with 1°C resolution.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MaxTemperature								MinTemperature							

MaxTemperature: Maximum Temp register reading

MinTemperature: Minimum Temp register reading

DieTemp Register (0x34)

Register Type: Temperature

The DieTemp register provides the internal die temperature measurement. If Config.TSel = 0, DieTemp and Temp registers have the value of the die temperature.

Power Measurement

Power Register (0xB1)

Instant power calculation from immediate current and voltage. The LSB is 1.6mW.

AvgPower Register (0xB3)

Filtered Average Power from the Power register. LSB is 1.6mW.

Alert Function

The Alert Threshold registers allow interrupts to be generated by detecting a high or low voltage, current, temperature, or state-of-charge. Interrupts are generated on the ALRT pin open-drain output driver. An external pullup is required to generate a logic-high signal. Alerts can be triggered by any of the following conditions:

- Battery removal: ($V_{TH} > V_{BAT} - V_{DET}$) and battery removal detection enabled (Ber = 1).
- Battery insertion: ($V_{TH} < V_{BAT} - V_{DET-HYS}$) and battery insertion detection enabled (Bei = 1).
- Over/undervoltage: VAlrtTh register threshold violation (upper or lower) and alerts enabled (Aen = 1).
- Over/undertemperature: TAlrtTh register threshold violation (upper or lower) and alerts enabled (Aen = 1).
- Over/undercurrent: IAlrtTh register threshold violation (upper or lower) and alerts enabled (Aen = 1).
- Over/under SOC: SAlrtTh register threshold violation (upper or lower) and alerts enabled (Aen = 1).
- 1% SOC change: RepSOC register bit d8 (1% bit) changed (dSOCen = 1).

To prevent false interrupts, the threshold registers should be initialized before setting the Aen bit. Alerts generated by battery insertion or removal can only be reset by clearing the corresponding bit in the Status (0x00) register. Alerts generated by a threshold-level violation can be configured to be cleared only by software or cleared automatically when the threshold level is no longer violated. See the Config (1Dh) and Config2 (BBh) register descriptions for details of the alert function configuration.

Serial Number Feature

Each IC provides a unique serial number ID. To read this serial number, clear the AtRateEn and the DPEn bit in the Config2 register. The 128-bit serial information overwrites the Dynamic Power and AtRate output registers. To continue Dynamic Power and AtRate operations after reading the serial number, the host should set Config2.AtRateEn and Config2.DPEn to 1.

Table 11. Serial Number Format

ADDRESS	Config2.AtRateEn = 1 || Config2.DPEn = 1	Config2.AtRateEn = 0 && Config2.DPEn = 0
0xD4	MaxPeakPower	Serial Number Word0
0xD5	SusPeakPower	Serial Number Word1
0xD9	MPPCurrent	Serial Number Word2
0xDA	SPPCurrent	Serial Number Word3
0xDC	AtQResidual	Serial Number Word4
0xDD	AtTTE	Serial Number Word5
0xDE	AtAvSoc	Serial Number Word6
0xDF	AtAvCap	Serial Number Word7

ModelGauge m5 Memory Space

Registers that relate to functionality of the ModelGauge m5 fuel gauge are located on pages 0h-4h and are continued on pages Bh and Dh. See the *ModelGauge m5 Algorithm* section for details of specific register operation. Register locations marked reserved should not be written to.

Table 12. ModelGauge m5 Register Memory Map

PAGE/WORD	00h	10h	20h	30h	40h	B0h	D0h
0h	Status	FullCapRep	TTF	Reserved	Reserved	Status2	RSense / UserMem3
1h	VAlrtTh	TTE	DevName	Reserved	Reserved	Power	ScOcvLim
2h	TAlrtTh	QRTable00	QRTable10	QRTable20	QRTable30	ID / UserMem2	VGain
3h	SAlrtTh	FullSocThr	FullCapNom	Reserved	RGain	AvgPower	SOCHold
4h	AtRate	RCell	Reserved	DieTemp	Reserved	IAlrtTh	MaxPeakPower
5h	RepCap	Reserved	Reserved	FullCap	dQAcc	TTFCfg	SusPeakPower
6h	RepSOC	AvgTA	Reserved	Reserved	dPAcc	CVMixCap	PackResistance
7h	Age	Cycles	AIN	Reserved	Reserved	CVHalfTime	SysResistance
8h	Temp	DesignCap	LearnCfg	RComp0	Reserved	CGTempCo	MinSysVoltage
9h	VCell	AvgVCell	FilterCfg	TempCo	ConvgCfg	Curve	MPPCurrent
Ah	Current	MaxMinTemp	RelaxCfg	VEmpty	VFRemCap	HibCfg	SPPCurrent
Bh	AvgCurrent	MaxMinVolt	MiscCfg	Reserved	Reserved	Config2	ModelCfg
Ch	QResidual	MaxMinCurr	TGain	Reserved	Reserved	VRipple	AtQResidual
Dh	MixSOC	Config	TOff	FStat	QH	RippleCfg	AtTTE
Eh	AvSOC	IChgTerm	CGain	Timer	Reserved	TimerH	AtAvSOC
Fh	MixCap	AvCap	COff	ShdnTimer	Reserved	Reserved	AtAvCap

I²C Serial Communication

General Description

The IC features a revision 3.0 I²C-compatible, 2-wire serial interface consisting of a bidirectional serial data line (SDA) and a serial clock line (SCL). This device acts as a slave-only device, relying on the master to generate a clock signal. SCL clock rates from 0Hz to 400kHz are supported.

I²C is an open-drain bus and therefore SDA and SCL require pullups. Optional resistors (24Ω) in series with SDA and SCL protect the device inputs from high-voltage spikes on the bus lines. Series resistors also minimize crosstalk and undershoot on bus signals.

Figure 23 shows the functional diagram for the I²C based communications controller. For additional information on I²C, refer to the I²C Bus Specification and User Manual which is available for free through the internet.

Features

- I²C Revision 3.0 compatible serial communications channel
- Compatible with any bus timing up to 400kHz
- Does not utilize I²C clock stretching

I²C Simplified Block Diagram

There are three pins (aside from GND) for the I²C-compatible interface. V_{IO} determines the logic level, SCL is the clock line, and SDA is the data line. Note that the interface cannot drive the SCL line.

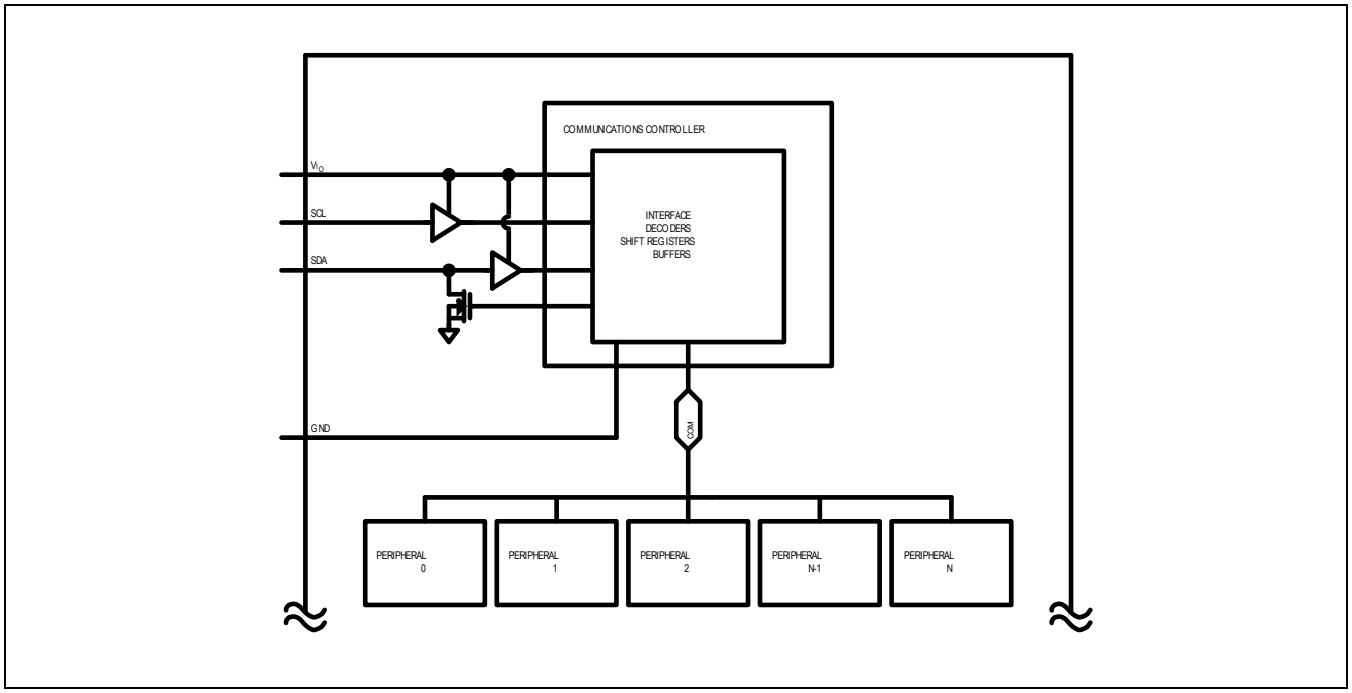


Figure 23. I²C Simplified Block Diagram

I²C System Configuration

The I²C-compatible interface is a multimaster bus. The maximum number of devices that can attach to the bus is only limited by bus capacitance.

A device on the I²C bus that sends data to the bus is called a transmitter. A device that receives data from the bus is called a receiver. The device that initiates a data transfer and generates the SCL clock signals to control the data transfer is a master. Any device that is being addressed by the master is considered a slave. The I²C-compatible interface operates as a slave on the I²C bus with transmit and receive capabilities.

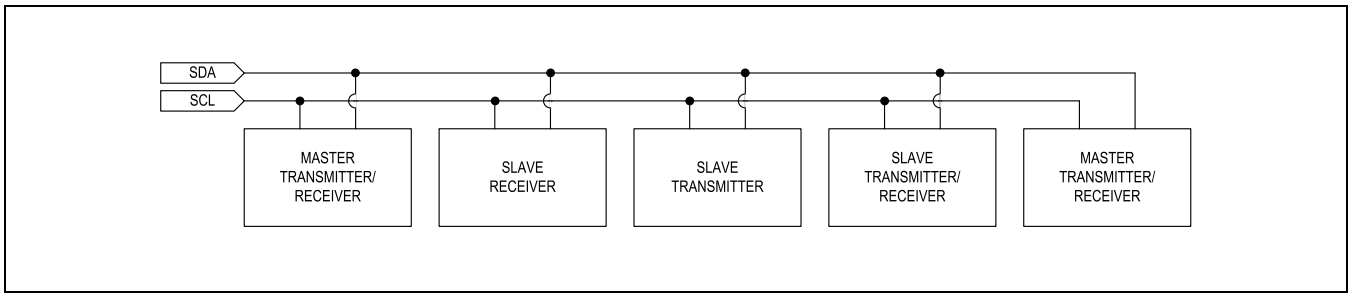


Figure 24. I²C System Configuration

I²C Interface Power

The I²C interface derives its power from V_{IO}. Typically, a power input such as V_{IO} would require a local 0.1µF ceramic bypass capacitor to ground. However, in highly integrated power distribution systems, a dedicated capacitor might not be necessary. If the impedance between V_{IO} and the next closest capacitor (≥ 0.1µF) is less than 100mΩ in series with 10nH, then a local capacitor is not needed. Otherwise, bypass V_{IO} to GND with a 0.1µF ceramic capacitor.

V_{IO} accepts voltages from 1.7V to 3.6V (V_{IO}). Cycling V_{IO} does not reset the I²C registers. When V_{IO} is less than V_{IOUVLO} and V_{SYSA} is less than $V_{SYSAUVLO}$, SDA and SCL are high impedance.

I²C Data Transfer

One data bit is transferred during each SCL clock cycle. The data on SDA must remain stable during the high period of the SCL clock pulse. Changes in SDA, while SCL is high, are control signals. See the *I2C Start and Stop Conditions* section. Each transmit sequence is framed by a START (S) condition and a STOP (P) condition. Each data packet is nine bits long: eight bits of data followed by the acknowledge bit. Data is transferred with the MSB first.

I²C Start and Stop Conditions

When the serial interface is inactive, SDA and SCL idle high. A master device initiates communication by issuing a START condition. A START condition is a high-to-low transition on SDA with SCL high. A STOP condition is a low-to-high transition on SDA, while SCL is high. See *Figure 25*.

A START condition from the master signals the beginning of a transmission to the device. The master terminates transmission by issuing a not-acknowledge followed by a STOP condition (see the *I2C Acknowledge Bit* section for information on not-acknowledge). The STOP condition frees the bus. To issue a series of commands to the slave, the master can issue repeated start (Sr) commands instead of a STOP command to maintain control of the bus. In general, a repeated start command is functionally equivalent to a regular start command.

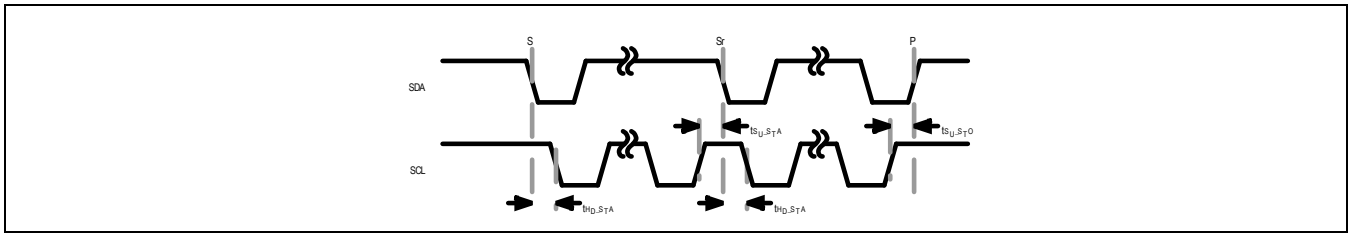


Figure 25. I²C Start and Stop Conditions

I²C Acknowledge Bit

Both the I²C bus master and slave devices generate acknowledge bits when receiving data. The acknowledge bit is the last bit of each ninth-bit data packet. To generate an acknowledge (A), the receiving device must pull SDA low before the rising edge of the acknowledge-related clock pulse (ninth pulse) and keep it low during the high period of the clock pulse. See *Figure 26*. To generate a not-acknowledge (nA), the receiving device allows SDA to be pulled high before the rising edge of the acknowledge-related clock pulse and leaves it high during the high period of the clock pulse.

Monitoring the acknowledge bits allows for the detection of unsuccessful data transfers. An unsuccessful data transfer occurs if a receiving device is busy or if a system fault has occurred. In the event of an unsuccessful data transfer, the bus master should reattempt communication at a later time.

This device issues an ACK for all register addresses in the possible address space even if the particular register does not exist.

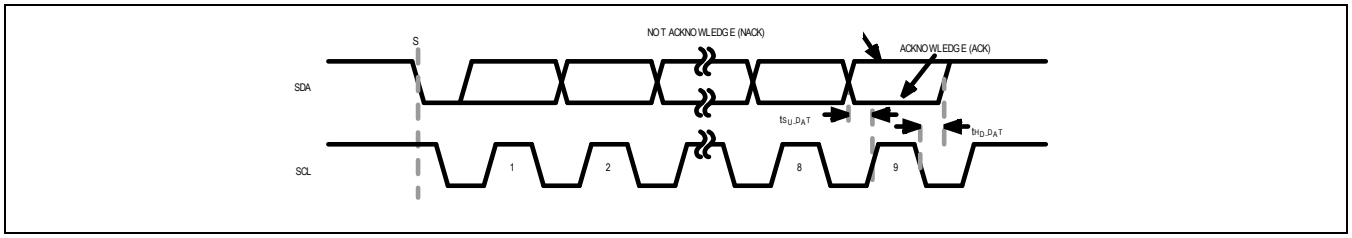


Figure 26. Acknowledge Bit

I²C Slave Address

The I²C controller implements 7-bit slave addressing. The registers of the MAX20355/MAX20357 are divided into three blocks with separate slave addresses:

- The main block includes all the registers for the global resource, the power line communication, and the buck-boost regulator (MAX20355)/the charger (MAX20357). All the registers in the main block are 8-bit registers.
- The fuel gauge block includes all the registers for the fuel gauging. The fuel gauge registers are in 16-bit word.
- The RAM block is 128-byte space for mailbox and bulk data transfer. The RAM registers are 8-bit.

MAX20355 ADDRESS	7-BIT SLAVE ADDRESS	8-BIT WRITE ADDRESS	8-BIT READ ADDRESS
PLC Address	0x28, 0b 010 1000	0x50, 0b 0101 0000	0x51, 0b 0101 0001
Fuel Gauge Address	0x36, 0b 011 0110	0x6C, 0b 0110 1100	0x6D, 0b 0110 1101
RAM Address	0x40, 0b 100 0000	0x80, 0b 1000 0000	0x81, 0b 1000 0001
MAX20357 ADDRESS	**7-BIT SLAVE ADDRESS**	**8-BIT WRITE ADDRESS**	**8-BIT READ ADDRESS**
PLC Address	0x15, 0b 001 0101	0x2A, 0b 0010 1010	0x2B, 0b 0010 1011
Fuel Gauge Address	0x36, 0b 011 0110	0x6C, 0b 0110 1100	0x6D, 0b 0110 1101
RAM Address	0x55, 0b 101 0101	0xAA, 0b 1010 1010	0xAB, 0b 1010 1011

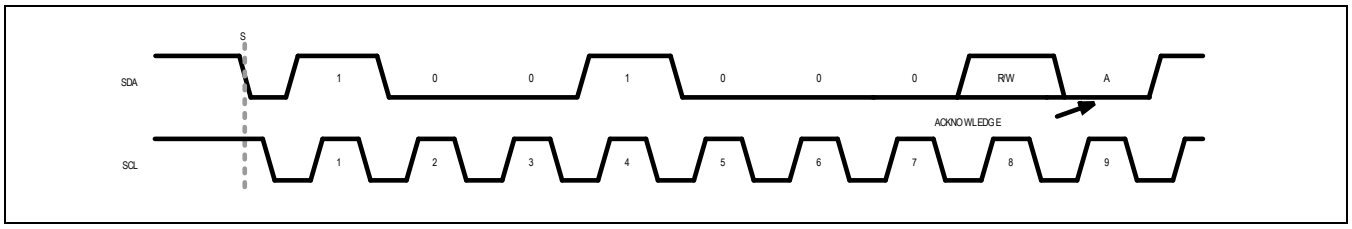


Figure 27. Slave Address Example

I²C Clock Stretching

In general, the clock signal generation for the I²C bus is the responsibility of the master device. The I²C specification allows slow slave devices to alter the clock signal by holding down the clock line. The process in which a slave device holds down the clock line is typically called clock stretching. The IC does not use any form of clock stretching to hold down the clock line.

I²C General Call Address

This device does not implement the I²C specifications general call address and does not acknowledge the general call address (0b0000_0000).

I²C Device ID

This device does not support the I²C Device ID feature.

I²C Communication Speed

This device is compatible with any bus timing up to 400kHz. The main consideration when changing bus speed through this range is the combination of the bus capacitance and pullup resistors. Larger values of bus capacitance and pullup resistance increase the time constant (C x R), slowing bus operation. Therefore, when increasing bus speeds, the pullup resistance must be decreased to maintain a reasonable time constant. Refer to the *Pullup Resistor Sizing* section of the I²C Bus Specification and User Manual (available for free on the internet) for detailed guidance on the pullup resistor

selection. In general, for bus capacitances of 200pF, a 100kHz bus needs 5.6kΩ pullup resistors, and a 400kHz bus needs about 1.5kΩ pullup resistors. Remember that, while the open-drain bus is low, the pullup resistor is dissipating power, and lower value pullup resistors dissipate more power (V^2/R).

Operating in high-speed mode requires some special considerations. For a full list of considerations, refer to the publicly available I²C Bus Specification and User Manual. Major considerations concerning this part are:

- The I²C bus master uses current source pullups to shorten the signal rise.
- The I²C slave must use a different set of input filters on its SDA and SCL lines to accommodate for the higher bus.
- The communication protocols need to utilize the high-speed master code.

At power-up and after each stop condition, the bus input filters are set for standard mode, fast mode, and fast-mode plus (i.e., 0Hz to 1MHz). To switch the input filters for high-speed mode, use the high-speed master code protocols that are described in the *I2C Communication Protocols* section.

I²C Communication Protocols

Both writing to and reading from registers are supported as described in the following subsections.

Writing to a Single 8-bit Register

Figure 28 shows the protocol for the I²C master device to write one byte of data to this device. This protocol is the same as the SMBus specification's write-byte protocol.

The write byte protocol is as follows:

1. The master sends a start command (S).
2. The master sends the 7-bit slave address followed by a write bit ($R/\overline{W} = 0$).
3. The addressed slave asserts an acknowledge (A) by pulling SDA low.
4. The master sends an 8-bit register pointer.
5. The slave acknowledges the register pointer.
6. The master sends a data byte.
7. The slave updates with the new data.
8. The slave acknowledges or not acknowledges the data byte. The next rising edge on SDA loads the data byte into its target register and the data becomes active.
9. The master sends a stop condition (P) or a repeated start condition (Sr). Issuing a P ensures that the bus input filters are set for 1MHz or slower operation. Issuing an Sr leaves the bus input filters in their current state.

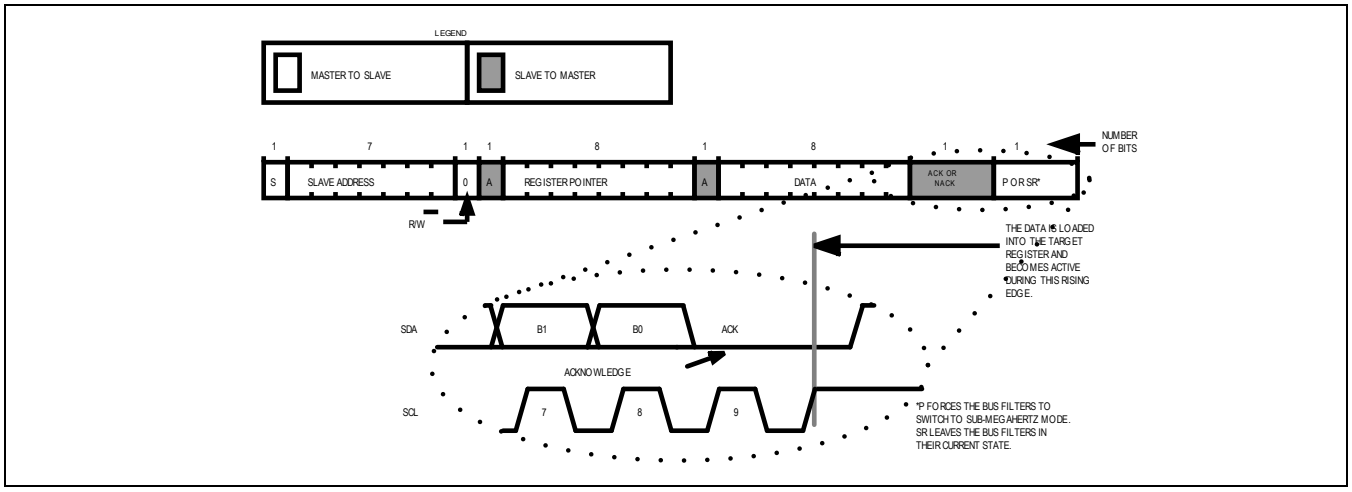


Figure 28. Writing to a Single 8-bit Register with the Write Byte Protocol

Writing Multiple Bytes to Sequential Registers

Figure 29 shows the protocol for writing to sequential registers. This protocol is similar to the write byte protocol above, except the master continues to write after it receives the first byte of data. When the master is done writing, it issues a stop or repeated start.

The writing to sequential registers protocol is as follows:

1. The master sends a start command (S).
2. The master sends the 7-bit slave address followed by a write bit (R/$\overline{W}$ = 0).
3. The addressed slave asserts an acknowledge (A) by pulling SDA low.
4. The master sends an 8-bit register pointer.
5. The slave acknowledges the register pointer.
6. The master sends a data byte.
7. The slave acknowledges the data byte. The next rising edge on SDA loads the data byte into its target register and the data becomes active.
8. Steps 6 to 7 are repeated as many times as the master requires.
9. During the last acknowledge-related clock pulse, the master can issue an acknowledge or a not acknowledge.
10. The master sends a stop condition (P) or a repeated start condition (Sr). Issuing a P ensures that the bus input filters are set for 1MHz or slower operation. Issuing an Sr leaves the bus input filters in their current state.

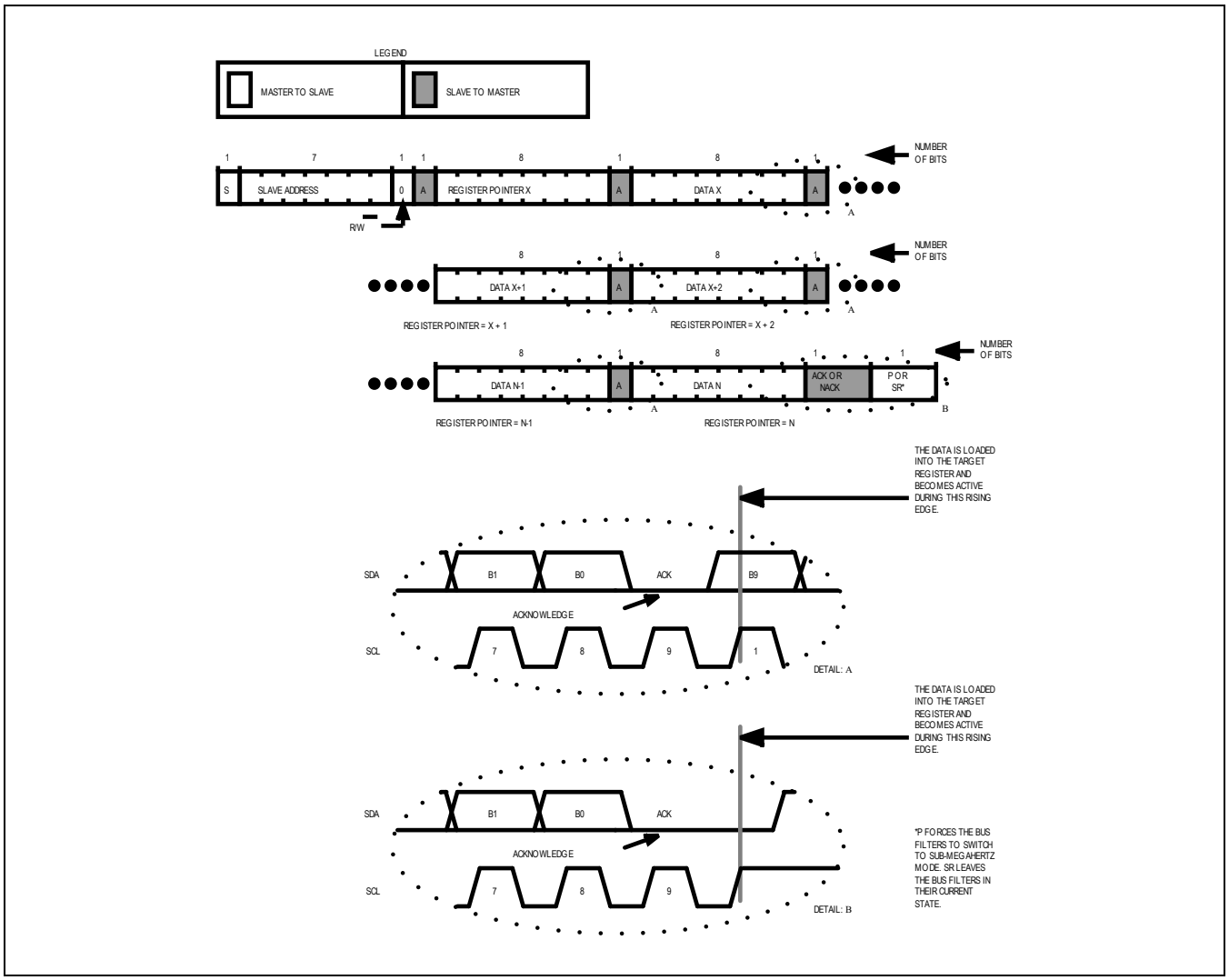


Figure 29. Writing to Sequential Registers X to N

Writing to 16-bit Registers

The Write Data protocol is used to transmit data to the registers of the fuel gauge at memory addresses from 00h to FFh. Addresses 00h to FFh can be written as a block. The memory address is sent by the bus master as a single byte value immediately after the slave address. The LSB of the data to be stored is written immediately after the memory address byte is acknowledged. Because the address is automatically incremented after the last bit of each 16-bit word received by the IC, the LSB of the data at the next memory address can be written immediately after the acknowledgment of the MSB of data at the previous address. The master indicates the end of a write transaction by sending a STOP or Repeated START after receiving the last acknowledge bit. If the bus master continues an auto-incremented write transaction beyond address FFh, the IC ignores the data. Data is also ignored on writes to read-only addresses but not reserved addresses. Do not write to reserved address locations. See *Figure 30* for an example of the Write Data communication sequence.

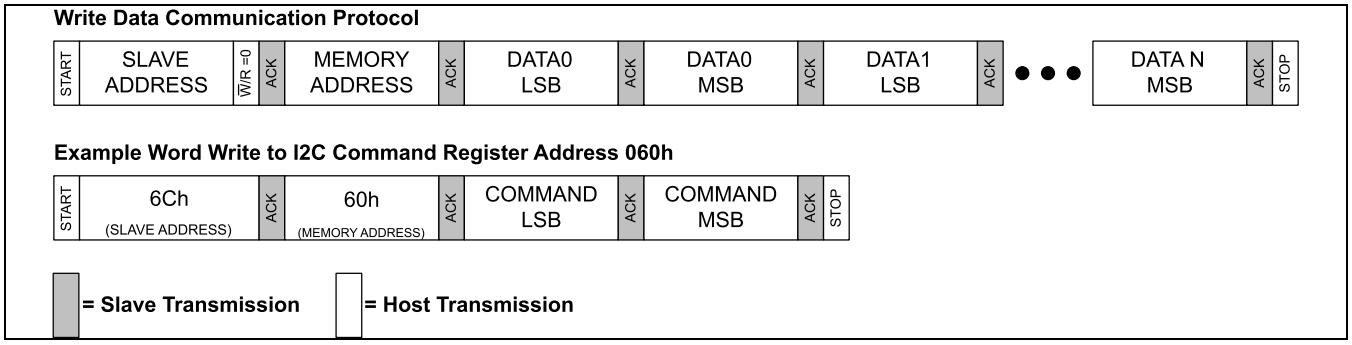


Figure 30. Example I²C Write 16-bit Data Communication Sequence

Reading from a Single Register

Figure 31 shows the protocol for the I²C master device to read one byte of data. This protocol is the same as the SMBus specification's read-byte protocol.

The read byte protocol is as follows:

1. The master sends a start command (S).
2. The master sends the 7-bit slave address followed by a write bit (R/$\overline{W}$ = 0).
3. The addressed slave asserts an acknowledge (A) by pulling SDA low.
4. The master sends an 8-bit register pointer.
5. The slave acknowledges the register pointer.
6. The master sends a repeated start command (Sr).
7. The master sends the 7-bit slave address followed by a read bit (R/$\overline{W}$ = 1).
8. The addressed slave asserts an acknowledge by pulling SDA low.
9. The addressed slave places 8-bits of data on the bus from the location specified by the register pointer.
10. The master issues a not acknowledge (nA).
11. The master sends a stop condition (P) or a repeated start condition (Sr). Issuing a P ensures that the bus input filters are set for 1MHz or slower operation. Issuing an Sr leaves the bus input filters in their current state.

Note that when this device receives a stop, the register pointer is not modified. Therefore, if the master re-reads the same register, it can immediately send another read command, omitting the command to send a register pointer.

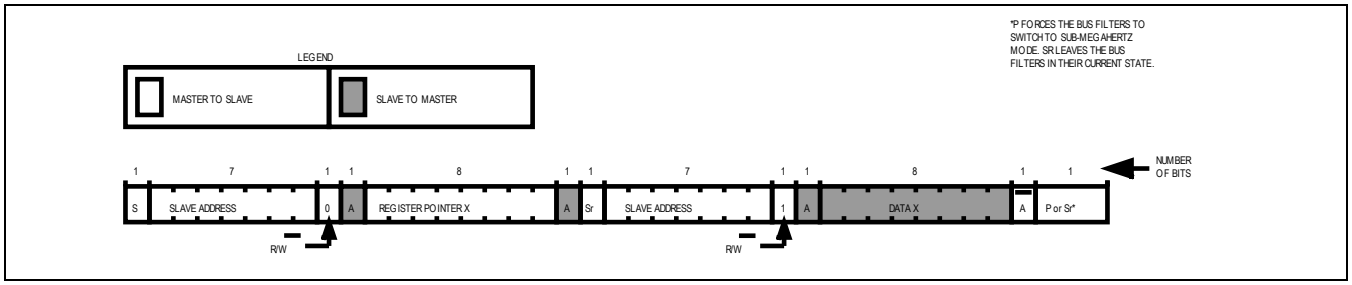


Figure 31. Reading from a Single Register with the Read Byte Protocol

Reading from Sequential Registers

Figure 32 shows the protocol for reading from sequential registers. This protocol is similar to the read byte protocol except the master issues an acknowledge to signal the slave that it wants more data: when the master has all the data it requires it issues a not acknowledge (nA) and a stop (P) to end the transmission.

The continuous read from sequential registers protocol is as follows:

1. The master sends a start command (S).
2. The master sends the 7-bit slave address followed by a write bit (R/$\overline{W}$ = 0).
3. The addressed slave asserts an acknowledge (A) by pulling SDA low.
4. The master sends an 8-bit register pointer.
5. The slave acknowledges the register pointer.
6. The master sends a repeated start command (Sr).
7. The master sends the 7-bit slave address followed by a read bit (R/$\overline{W}$ = 1).
8. The addressed slave asserts an acknowledge by pulling SDA low.
9. The addressed slave places 8-bits of data on the bus from the location specified by the register pointer.
10. The master issues an acknowledge (A) signaling the slave that it wishes to receive more data.
11. Steps 9 to 10 are repeated as many times as the master requires. Following the last byte of data, the master must issue a not acknowledge (nA) to signal that it wishes to stop receiving data.
12. The master sends a stop condition (P) or a repeated start condition (Sr). Issuing a stop (P) ensures that the bus input filters are set for 1MHz or slower operation. Issuing an Sr leaves the bus input filters in their current state.

Note that when this device receives a stop it does not modify its register pointer. Therefore, if the master re-reads the same register, it can immediately send another read command, omitting the command to send a register pointer.

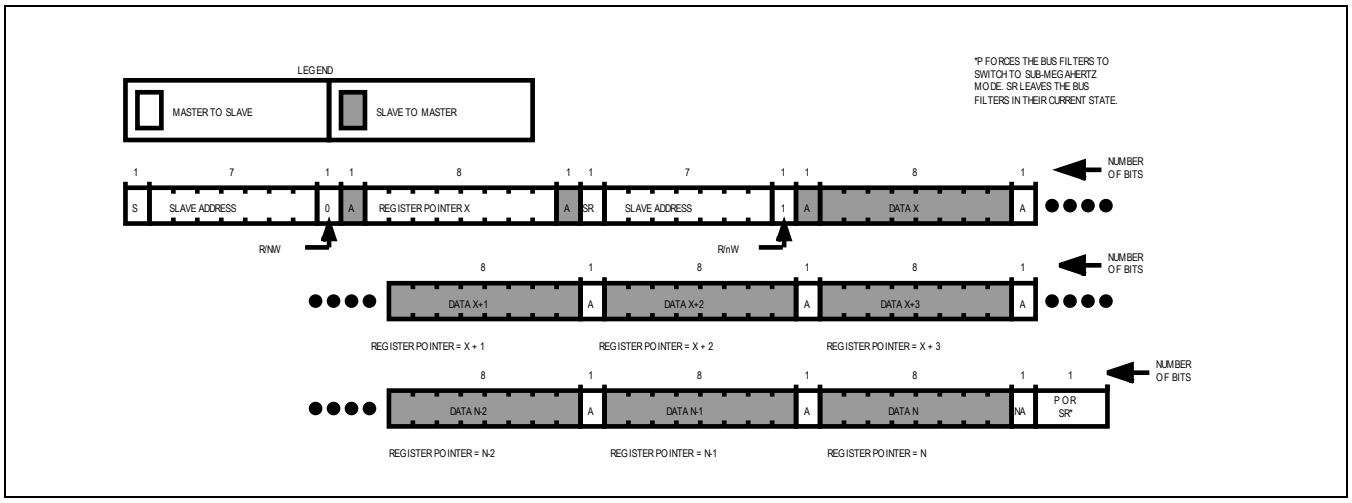


Figure 32. Reading Continuously from Sequential Registers X to N

Applications Information

PLC Current Sink and Detection Threshold

The MAX20355 and MAX20357 PLC sink current and detection threshold are specially paired to maintain good performance. Buck-boost ripple is the major source of noise inside the system. MAX20355 self-cancels the effect of the buck-boost ripple since the noise is created by itself. The MAX20357 is not able to cancel the noise. To compensate for the ripple noise, all the PLC sink current PLCSnkSel options of the MAX20355 are higher than MAX20357 PLC sink current as shown in *Table 13* and *Table 14*.

Table 13. PLC Detection Pair (MAX20355 Data Packet)

DECODE	MAX20355 **PLCSnkSel**	MAX20357 **PLCThrSel**
0b00	88mA	75mV
0b01	105.4mA	90mV
0b10	123mA	105mV
0b11	140.5mA	120mV

Table 14. PLC Detection Pair (MAX20357 Data Packet)

DECODE	MAX20357 **PLCSnkSel**	MAX20355 **PLCThrSel**
0b00	50.3mA	40mV
0b01	70.4mA	56mV
0b10	90.4mA	70mV
0b11	110.5mA	84mV

PLC voltage detection threshold needs to be placed in the middle of the PLC signal for better performance. For each MAX20355 sink current PLCSnkSel, MAX20357 offers one paired voltage detection threshold PLCThrSel. Make sure always use the same code for MAX20355 PLCSnkSel and MAX20357 PLCThrSel (*Table 13*). Similarly, use the same code for MAX20357 PLCSnkSel and MAX20355 PLCThrSel (*Table 14*). It is recommended to start from code '00'. If the external noise is too strong and code '00' does not offer good performance, increase the sink current by moving to larger codes. Note there is a tradeoff between susceptibility to noise and power consumption. Higher sink current offers a better anti-interference ability, but also higher power consumption.

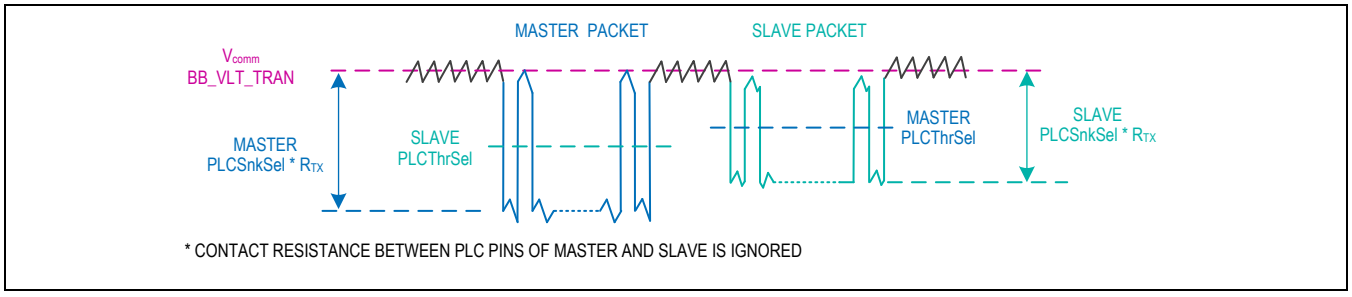


Figure 33. PLC Current Sink and Detection Threshold

Typical Application Circuit

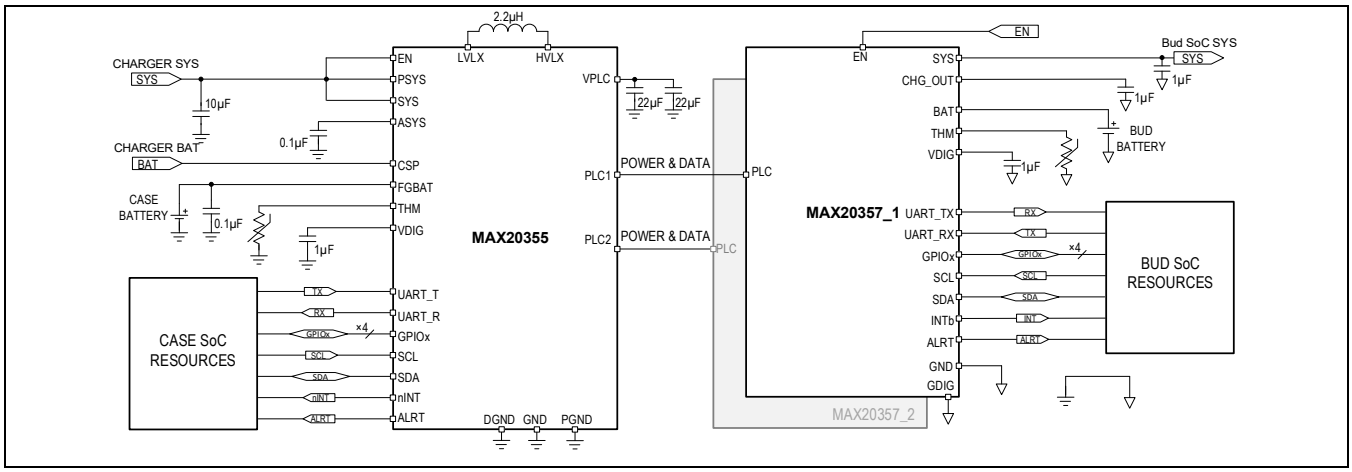


Figure 34. Typical Application Circuit

PCB Layout Guidelines

Careful circuit board layout is critical to achieving low switching power losses and clean, stable operation. *Figure 35* shows a PCB layout example. The following guidelines are references to design the PCB:

1. Place the PSYS capacitor (CPSYS) and VPLC capacitors (CVPLC) close to the VPLC pin and PSYS pin of the IC, respectively. Since the IC operates at a high switching frequency, this placement is critical for minimizing parasitic inductance within the input and output current loops which can cause high voltage spikes and can damage the internal switching MOSFETs.
2. Place the inductor next to the HVLX and LVLX pins and make the traces between the LX pins and the inductor short and wide to minimize PCB trace impedance. Excessive PCB impedance reduces converter efficiency. Furthermore, do not make LX traces take up an excessive amount of area. The voltage on LX nodes switches very quickly and additional area creates more radiated emissions.
3. Place the ASYS capacitor (CASYS) close to the ASYS pin. Proximity to the IC provides a stable supply for the internal circuitry.
4. Place the VDIG capacitor (CVDIG) close to the VDIG pin. Proximity to the IC provides a stable supply for the internal circuitry.
5. Do not neglect ceramic capacitor DC voltage derating. Choose capacitor values and case sizes carefully. Select ceramic capacitors that maintain capacitance over-temperature and DC bias. Refer to the Tutorial 5527 for more information.

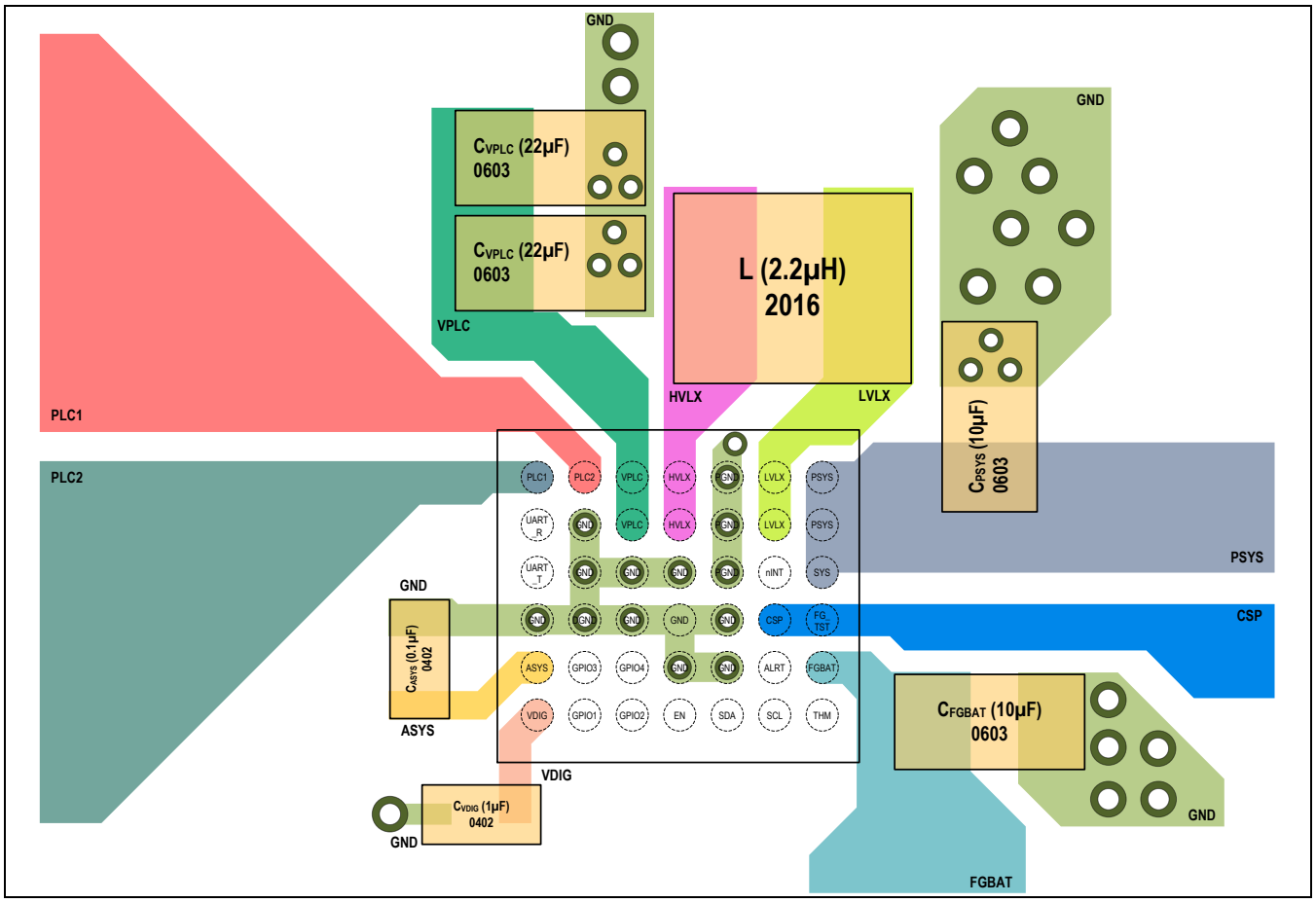


Figure 35. Layout Guideline

Register Map

MAX20355 (Slave Address 0x28)

ADDRESS	NAME	MSB							LSB
USER_INTERRUPT									
0x00	REVISION_ID[7:0]	Revision_id[7:0]							
0x01	Status0[7:0]	itf_rdy_sts	ch1_con_sts	ch2_con_sts	ch1_idl_sts	ch2_idl_sts	–	plc2_moi_det	plc1_moi_det
0x02	Status1[7:0]	plc_status1[7:0]							
0x03	Status2[7:0]	plc_status2[7:0]							
0x04	Status3[7:0]	sys_err_int	bb_fault	thm_flt_sts	–	–	–	–	–

ADDRESS	NAME	MSB							LSB
0x05	Int0[7:0]	itf_rdy_stsInt	ch1_con_int	ch2_con_int	ch1_idl_int	ch2_idl_int	moi_dne_Int	plc2_moi_detInt	plc1_moi_detInt
0x06	Int1[7:0]	sys_err_intlrq	bb_faultIrq	thm_flt_Int	plc_new_dat	plc2_cmd_dne	plc1_cmd_dne	plc2_cmd_err	plc1_cmd_err
0x07	Int2[7:0]	–	–	–	–	moi_irq_det	res_det_abr	res_det_opn	res_det_gnd
0x08	Int3[7:0]	–	–	urt_tmo_flt2	urt_modfail2	urt_moddone2	urt_tmo_flt1	urt_modfail1	urt_moddone1
0x09	IntMask0[7:0]	itf_rdy_stsIntM	ch1_con_intM	ch2_con_intM	ch1_idl_intM	ch2_idl_intM	moi_dne_IntM	plc2_moi_detIntM	plc1_moi_detIntM
0x0A	IntMask1[7:0]	sys_err_intlrqM	bb_faultIrqM	thm_flt_IntM	plc_new_datM	plc2_cmd_dneM	plc1_cmd_dneM	plc2_cmd_errM	plc1_cmd_errM
0x0B	IntMask2[7:0]	–	–	–	0	moi_irq_detM	res_det_abrM	res_det_opnM	res_det_gndM
0x0C	IntMask3[7:0]	–	–	urt_tmo_flt2M	urt_modfail2M	urt_moddone2M	urt_tmo_flt1M	urt_modfail1M	urt_moddone1M
USER_BOT									
0x1A	SYSTEM_REG0[7:0]	off_cmd_inp	soft_reset	–	–	–	–	Stay_ON	enb_otp_ena
0x1B	BOT_CMD[7:0]	fg____reset	fg_ena_byp	fg_ena_val	–	–	–	–	–
USER_UART									
0x21	UART_ctr1[7:0]	pl1_urt_dis	tmo_tmr_ena1	i2c_urt_mod1	i2c_urt_ena1	i2c_urt_abr1	i2c_urt_swc1	i2c_tx__swc1	i2c_rx__swc1
0x22	UART_Ctr2[7:0]	pl2_urt_dis	tmo_tmr_ena2	i2c_urt_mod2	i2c_urt_ena2	i2c_urt_abr2	i2c_urt_swc2	i2c_tx__swc2	i2c_rx__swc2
USER_SYSTEM_CONFIG									
0x30	SYSTEM_CONFIG0[7:0]	–	low_pwr_ena	–	–	–	–	–	–
0x31	PLC_CONFIG0[7:0]	PLCSnkSel[1:0]		–	–	–	–	PLCThrSel[1:0]	
0x32	PLC_CONFIG1[7:0]	–	–	plc2_iprog[2:0]			plc1_iprog[2:0]		

ADDRESS	NAME	MSB							LSB
0x33	PLC_CONFIG2[7:0]	pl2_chn_ena	pl1_chn_ena	pl2_res_req	pl1_res_req	png_rty_num[2:0]			plc_double
0x34	PLC_CONFIG3[7:0]	plc_rrt_tmr[7:0]							
0x35	PLC_CONFIG4[7:0]	plc_config[7:0]							
0x36	PLC_CONFIG5[7:0]	plc_is_full	RAM_is_full	0	0	0	0	–	cont_stream
0x37	PLC_CONFIG6[7:0]	swp_plc_ram	–	no_uart_mde	no_idle_mde	–	DAT_MAX_RTY[2:0]		
0x38	PLC_ARG1[7:0]	plc_cmd_arg1[7:0]							
0x39	PLC_CMD1[7:0]	plc_run_trg1	plc_command1[6:0]						
0x3A	PLC_ARG2[7:0]	plc_cmd_arg2[7:0]							
0x3B	PLC_CMD2[7:0]	plc_run_trg2	plc_command2[6:0]						
0x3C	PLC_RX[7:0]	rx__pl1_pl2	plc_rx_bytes[6:0]						
0x3D	PLC_FIFO[7:0]	–	–	–	–	pl2_fifo_master	pl2_fifo_slave	pl1_fifo_master	pl1_fifo_slave
USER_BB_CONTROL									
0x40	BB_UP_DOWN[7:0]	up_down_val[2:0]			i2c_frc_def	1	1	bb__alg_max	bb__alg_min
0x41	BB_VOLT_DEF[7:0]	bb_vlt__def[7:0]							
0x42	BB_VLT_TRAN[7:0]	bb_vlt_tran[7:0]							
0x46	BB_RMP_CFG3[7:0]	–	–	–	–	–	min_ovl_sel[2:0]		
0x47	BB_ANA_CFG1[7:0]	bb_psv_dchg	bb_act_dchg	0	0	0	0	11[1:0]	
0x48	BB_ANA_CFG2[7:0]	0	bb__frc_ena	–	–	bb__low__bw	bb___l2uh	i2c_zccm_enb	–
0x4A	BB_ALG1[7:0]	avg_t_delta[7:0]							

ADDRESS	NAME	MSB							LSB
0x4B	BB_MULTI[7:0]	–	–	–	–	–	up__dwn_dis	multi_up_en	multi_dw_en
USER_GPIO									
0x58	GPIO1[7:0]	–	–	–	–	GPIOPLCCtr1	GPIOEnRes1	GPIOEnPup1	GPIODout_1
0x59	GPIO2[7:0]	–	–	–	–	GPIOPLCCtr2	GPIOEnRes2	GPIOEnPup2	GPIODout_2
0x5A	GPIO3[7:0]	–	–	–	–	GPIOPLCCtr3	GPIOEnRes3	GPIOEnPup3	GPIODout_3
0x5B	GPIO4[7:0]	–	–	–	–	GPIOPLCCtr4	GPIOEnRes4	GPIOEnPup4	GPIODout_4
0x5C	GPIO_rdb1[7:0]	GPIODAInp_4	GPIODAInp_3	GPIODAInp_2	GPIODAInp_1	–	–	–	GPIOCmosEn
0x5D	GPIO_rdb2[7:0]	PLC2_GPIO4	PLC2_GPIO3	PLC2_GPIO2	PLC2_GPIO1	PLC1_GPIO4	PLC1_GPIO3	PLC1_GPIO2	PLC1_GPIO1
USER_DOP_PORT									
0x60	SOC_byte_1[7:0]	SOC_byte_1[7:0]							
0x61	SOC_byte_0[7:0]	SOC_byte_0[7:0]							
0x62	VCELL_byte_1[7:0]	VCELL_byte_1[7:0]							
0x63	VCELL_byte_0[7:0]	VCELL_byte_0[7:0]							
0x64	TTE_byte_1[7:0]	TTE_byte_1[7:0]							
0x65	TTE_byte_0[7:0]	TTE_byte_0[7:0]							
0x66	AVGVCELL_byte_1[7:0]	AVGVCELL_byte_1[7:0]							
0x67	AVGVCELL_byte_0[7:0]	AVGVCELL_byte_0[7:0]							
0x68	TTF_byte_1[7:0]	TTF_byte_1[7:0]							
0x69	TTF_byte_0[7:0]	TTF_byte_0[7:0]							
0x6A	READY_REG[7:0]	dop_rdy_sig	–	dop_i2c_ena	ttf_reg_rdy	avgvcell_reg_rdy	tte_reg_rdy	vcell_reg_rdy	soc_reg_rdy

ADDRESS	NAME	MSB							LSB
0x6B	FG_RDY_1[7:0]	slv_avg__1[7:0]							
0x6C	FG_RDY_2[7:0]	slv_soc__1[7:0]							
0x6D	FG_RDY_3[7:0]	slv_avg__2[7:0]							
0x6E	FG_RDY_4[7:0]	slv_soc__2[7:0]							
0x6F	FG_RDY_5[7:0]	slv2_chg_done	slv1_chg_dne	–	–	slv_socrdy2	slv_avgrdy2	slv_socrdy1	slv_avgrdy1
USER_MOISTURE_DETECTION									
0x70	ADC_CTRL1[7:0]	–	AdcGndTrh[3:0]				ResDetRty[2:0]		
0x71	ADC_CTRL2[7:0]	–	–	AdcRng[5:0]					
0x72	ADC_CTRL3[7:0]	–	–	–	–	–	AdcAvgNum[2:0]		
0x73	ADC_CTRL4[7:0]	–	–	AdcNoiseCtr[5:0]					
0x74	MOI_DET_REG1[7:0]	–	–	–	–	–	–	RaccDetMIp[1:0]	
0x75	MOI_DET_REG2[7:0]	RaccDetMThr[7:0]							
0x76	MOI_DET_REG3[7:0]	moi_det_aut2	moi_det_aut1	moi_man_pl2	moi_man_pl1	moi_man_rty2	moi_man_rty1	moi_aut_rty2	moi_aut_rty1
0x78	IP RES_REG[7:0]	–	–	–	–	–	–	IP__RES_DET[1:0]	
0x79	ADC_VAL1[7:0]	ADCAvg[7:0]							
0x7A	ADC_VAL2[7:0]	ADCMax[7:0]							
0x7B	ADC_VAL3[7:0]	ADCMin[7:0]							

Register Details

REVISION_ID (0x00)

BIT	7	6	5	4	3	2	1	0
Field	Revision_id[7:0]							
Reset	0x2							

Access Type	Read Only

BITFIELD	BITS	DESCRIPTION
Revision_id	7:0	Revision ID

Status0 (0x01)

BIT	7	6	5	4	3	2	1	0
Field	itf_rdy_sts	ch1_con_sts	ch2_con_sts	ch1_idl_sts	ch2_idl_sts	–	plc2_moi_det	plc1_moi_det
Reset	0x1	0x0	0x0	0x0	0x0	–	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	–	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
itf_rdy_sts	7	OTP Loading Complete	0x0: OTP loading not completed. 0x1: OTP loading completed.
ch1_con_sts	6	PLC1 Connection Status	0x0: Disconnected 0x1: Connected
ch2_con_sts	5	PLC2 Connection Status	0x0: Disconnected 0x1: Connected
ch1_idl_sts	4	PLC1 Idle Status	0x0: Not in idle state. 0x1: Idle state.
ch2_idl_sts	3	PLC2 Idle Status	0x0: Not in idle state. 0x1: Idle state.
plc2_moi_det	1	PLC2 Moisture Detection Status	0x0: No moisture detected. 0x1: Moisture detected.
plc1_moi_det	0	PLC1 Moisture Detection Status	0x0: No moisture detected. 0x1: Moisture detected.

Status1 (0x02)

BIT	7	6	5	4	3	2	1	0
Field	plc_status1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
plc_status1	7:0	PLC1 Communication Status. Valid on plc1_cmd_dne or plc1_cmd_err interrupt.	0x00: NO_PLC_ERROR 0x10: NACK_TO__CMD 0x13: BAD__CMD__ID 0x16: CMD____ERROR 0x17: NAK____LIMIT

Status2 (0x03)

BIT	7	6	5	4	3	2	1	0
Field	plc_status2[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
plc_status2	7:0	PLC2 Communication Status. Valid on plc2_cmd_dne or plc2_cmd_err interrupt.	0x00: NO_PLC_ERROR 0x10: NACK_TO__CMD 0x13: BAD__CMD__ID 0x16: CMD____ERROR 0x17: NAK____LIMIT

Status3 (0x04)

BIT	7	6	5	4	3	2	1	0
Field	sys_err_int	bb_fault	thm_flt_sts	–	–	–	–	–
Reset	0x0	0x0	0x0	–	–	–	–	–
Access Type	Read Only	Read Only	Read Only	–	–	–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
sys_err_int	7	Buck-Boost Output Voltage UVLO Fault	0x0: No fault. 0x1: Fault active.
bb_fault	6	Buck-Boost Fault	0x0: No fault. 0x1: Fault active.
thm_flt_sts	5	Thermal fault status	0x0: No fault. 0x1: Fault active.

Int0 (0x05)

BIT	7	6	5	4	3	2	1	0
Field	itf_rdy_stsInt	ch1_con_int	ch2_con_int	ch1_idl_int	ch2_idl_int	moi_dne_Int	plc2_moi_detInt	plc1_moi_detInt
Reset	0x1	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
itf_rdy_stsInt	7	OTP Loading Complete. Device is ready to communicate through I²C after this bit is asserted. Asserted if a change occurs on itf_rdy_sts.	0x0: Status is not changed. 0x1: Status is changed.

BITFIELD	BITS	DESCRIPTION	DECODE
ch1_con_int	6	PLC1 Connection Interrupt. Asserted if a change occurs on ch1_con_sts.	0x0: Status is not changed. 0x1: Status is changed.
ch2_con_int	5	PLC2 Connection Interrupt. Asserted if a change occurs on ch2_con_sts.	0x0: Status is not changed. 0x1: Status is changed.
ch1_idl_int	4	plc1 Connected Device Idle Interrupt. Asserted if a change occurs on ch1_idl_sts.	0x0: Status is not changed. 0x1: Status is changed.
ch2_idl_int	3	plc2 Connected Device Idle Interrupt. Asserted if a change occurs on ch2_idl_sts.	0x0: Status is not changed. 0x1: Status is changed.
moi_dne_Int	2	Moisture Detection Done Interrupt. It occurs when a moisture measure is completed.	0x0: Moisture measurement is not completed. 0x1: Moisture measurement is completed.
plc2_moi_detInt	1	Moisture Detection Interrupt for PLC2. It occurs when moisture is detected on PLC2. Asserted if a change occurs on plc2_moi_det.	0x0: Status is not changed. 0x1: Status is changed.
plc1_moi_detInt	0	Moisture Detection Interrupt for PLC1. It occurs when moisture is detected on PLC1. Asserted if a change occurs on plc1_moi_det.	0x0: Status is not changed 0x1: Status is changed

Int1 (0x06)

BIT	7	6	5	4	3	2	1	0
Field	sys_err_intIrq	bb_faultIrq	thm_flt_Int	plc_new_dat	plc2_cmd_dne	plc1_cmd_dne	plc2_cmd_err	plc1_cmd_err
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
sys_err_intIrq	7	Buck-Boost Output UVLO Fault Interrupt. Asserted if a change occurs on sys_err_int.	0x0: Status is not changed. 0x1: Status is changed.
bb_faultIrq	6	Buck-Boost Fault Interrupt. Asserted if a change occurs on bb_fault status.	0x0: Status is not changed. 0x1: Status is changed.
thm_flt_Int	5	Thermal Fault Interrupt. Asserted if a change occurs on thm_flt_sts.	0x0: Status is not changed. 0x1: Status is changed.
plc_new_dat	4	PLC New Data Interrupt	0x0: No new data available. 0x1: New data available in RAM.
plc2_cmd_dne	3	PLC2 Command Done Interrupt	0x0: PLC2 command not completed. 0x1: PLC2 command done.
plc1_cmd_dne	2	PLC1 Command Done Interrupt	0x0: PLC1 command not completed. 0x1: PLC1 command done.
plc2_cmd_err	1	PLC2 Command Error Interrupt	0x0: PLC2 command no error. 0x1: PLC2 command error occured.
plc1_cmd_err	0	PLC1 Command Error Interrupt	0x0: PLC1 command no error. 0x1: PLC1 command error occured.

Int2 (0x07)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	moi_irq_det	res_det_abr	res_det_opn	res_det_gnd
Reset	–	–	–	–	0x0	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
moi_irq_det	3	Moisture Detection Measurement Valid Result Interrupt	0x0: Moisture detection valid value not detected. 0x1: Moisture detection valid value detected.
res_det_abr	2	Abort Resistive Measurement Result Interrupt	0x0: Abort resistive value not detected. 0x1: Abort resistive value detected.
res_det_opn	1	Open Resistive Measurement Result Interrupt	0x0: Open resistive value not detected. 0x1: Open resistive value detected.
res_det_gnd	0	Ground Resistive Measurement Result Interrupt	0x0: Ground resistive value not detected. 0x1: Ground resistive value detected.

Int3 (0x08)

BIT	7	6	5	4	3	2	1	0
Field	–	–	urt_tmo_flt2	urt_modfail2	urt_moddone2	urt_tmo_flt1	urt_modfail1	urt_moddone1
Reset	–	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
urt_tmo_flt2	5	PLC2 UART Timeout Interrupt	0x0: UART mode timeout not expired. 0x1: UART mode timeout expired.
urt_modfail2	4	PLC2 UART Mode Fail Interrupt	0x0: UART mode not failed. 0x1: UART mode failed.
urt_moddone2	3	PLC2 UART Mode Done Interrupt. Asserted when device enters/exits UART mode.	0x0: No UART mode done event. 0x1: UART mode done.
urt_tmo_flt1	2	PLC2 UART Timeout Interrupt	0x0: UART mode timeout not expired. 0x1: UART mode timeout expired.
urt_modfail1	1	PLC1 UART Mode Fail Interrupt	0x0: UART mode not failed. 0x1: UART mode failed.
urt_moddone1	0	PLC1 UART Mode Done Interrupt. Asserted when device enters/exits UART mode.	0x0: No UART mode done event. 0x1: UART mode done.

IntMask0 (0x09)

BIT	7	6	5	4	3	2	1	0

Field	itf_rdy_stsIntM	ch1_con_intM	ch2_con_intM	ch1_idl_intM	ch2_idl_intM	moi_dne_IntM	plc2_moi_detIntM	plc1_moi_detIntM
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
itf_rdy_stsIntM	7	itf_rdy_int Interrupt Mask	0x0: Masked 0x1: Unmasked
ch1_con_intM	6	ch1_con_int Interrupt Mask	0x0: Masked 0x1: Unmasked
ch2_con_intM	5	ch2_con_int Interrupt Mask	0x0: Masked 0x1: Unmasked
ch1_idl_intM	4	ch1_idl_int Interrupt Mask	0x0: Masked 0x1: Unmasked
ch2_idl_intM	3	ch2_idl_int Interrupt Mask	0x0: Masked 0x1: Unmasked
moi_dne_IntM	2	moi_dne_int Interrupt Mask	0x0: Masked 0x1: Unmasked
plc2_moi_detIntM	1	plc2_moi_detInt Interrupt Mask	0x0: Masked 0x1: Unmasked
plc1_moi_detIntM	0	plc1_moi_detInt Interrupt Mask	0x0: Masked 0x1: Unmasked

IntMask1 (0x0A)

BIT	7	6	5	4	3	2	1	0
Field	sys_err_intIrqM	bb_faultIrqM	thm_flt_IntM	plc_new_datM	plc2_cmd_dneM	plc1_cmd_dneM	plc2_cmd_errM	plc1_cmd_errM
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
sys_err_intIrqM	7	sys_err_int Interrupt Mask	0x0: Masked 0x1: Unmasked
bb_faultIrqM	6	bb_faultirq Interrupt Mask	0x0: Masked 0x1: Unmasked
thm_flt_IntM	5	thm_flt_int Interrupt Mask	0x0: Masked 0x1: Unmasked
plc_new_datM	4	plc_new_dat Interrupt Mask	0x0: Masked 0x1: Unmasked
plc2_cmd_dneM	3	plc2_cmd_dne Interrupt Mask	0x0: Masked 0x1: Unmasked
plc1_cmd_dneM	2	plc1_cmd_dne Interrupt Mask	0x0: Masked 0x1: Unmasked
plc2_cmd_errM	1	plc2_cmd_err Interrupt Mask	0x0: Masked 0x1: Unmasked

BITFIELD	BITS	DESCRIPTION	DECODE
plc1_cmd_errM	0	plc1_cmd_err Interrupt Mask	0x0: Masked 0x1: Unmasked

IntMask2 (0x0B)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	0	moi_irq_detM	res_det_abrM	res_det_opnM	res_det_gndM
Reset	–	–	–	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
0	4	Reserved.	Keep 0. Do not change.
moi_irq_detM	3	moi_irq_det Interrupt Mask	0x0: Masked 0x1: Unmasked
res_det_abrM	2	res_det_abr Interrupt Mask	0x0: Masked 0x1: Unmasked
res_det_opnM	1	res_det_opn Interrupt Mask	0x0: Masked 0x1: Unmasked
res_det_gndM	0	res_det_gnd Interrupt Mask	0x0: Masked 0x1: Unmasked

IntMask3 (0x0C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	urt_tmo_flt2M	urt_modfail2M	urt_moddone2M	urt_tmo_flt1M	urt_modfail1M	urt_moddone1M
Reset	–	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
urt_tmo_flt2M	5	urt_tmo_flt2 Interrupt Mask	0x0: Masked 0x1: Unmasked
urt_modfail2M	4	urt_modfail2 Interrupt Mask	0x0: Masked 0x1: Unmasked
urt_moddone2M	3	urt_moddone2 Interrupt Mask	0x0: Masked 0x1: Unmasked
urt_tmo_flt1M	2	urt_tmo_flt1 Interrupt Mask	0x0: Masked 0x1: Unmasked
urt_modfail1M	1	urt_modfail1 Interrupt Mask	0x0: Masked 0x1: Unmasked
urt_moddone1M	0	urt_moddone1 Interrupt Mask	0x0: Masked 0x1: Unmasked

SYSTEM_REG0 (0x1A)

BIT	7	6	5	4	3	2	1	0
Field	off_cmd_inp	soft_reset	–	–	–	–	Stay_ON	enb_otp_ena
Reset	0x0	0x0	–	–	–	–	0x1	0x1
Access Type	Write, Read	Write, Read	–	–	–	–	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
off_cmd_inp	7	OFF Mode Request	0x0: No request. 0x1: Request to enter OFF mode.
soft_reset	6	Soft Reset Request	0x0: No request. 0x1: Request soft reset.
Stay_ON	1	Device Stays On After Turn On	0x0: 500ms shutdown timer enabled. 0x1: Device remains ON.
enb_otp_ena	0	EN Pin Functionality	0x0: EN pin does not control OFF mode. 0x1: EN pin controls OFF mode.

BOT_CMD (0x1B)

BIT	7	6	5	4	3	2	1	0
Field	fg____reset	fg_ena_byp	fg_ena_val	–	–	–	–	–
Reset	0x0	0x0	0x0	–	–	–	–	–
Access Type	Write, Read	Write, Read	Write, Read	–	–	–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
fg____reset	7	Fuel Gauge Reset. Autoclear when done.	0x0: No request. 0x1: Request fuel gauge reset.
fg_ena_byp	6	Fuel Gauge Manual Enable Functionality	0x0: Fuel gauge manual enable feature disabled. 0x1: Fuel gauge manual enable feature enabled.
fg_ena_val	5	Fuel Gauge Manual Enable. Valid if fg_ena_byp = 1.	0x0: Fuel gauge off. 0x1: Fuel gauge on.

UART_ctr1 (0x21)

BIT	7	6	5	4	3	2	1	0
Field	pl1_urt_dis	tmo_tmr_ena1	i2c_urt_mod1	i2c_urt_ena1	i2c_urt_abr1	i2c_urt_swc1	i2c_tx__swc1	i2c_rx__swc1
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
pl1_urt_dis	7	PLC1 abort UART mode if entered from UART_REQ PLC command.	0x0: Not Abort 0x1: Abort
tmo_tmr_ena1	6	PLC1 UART Timeout	0x0: Disable 0x1: Enable
i2c_urt_mod1	5	PLC1 UART Enter/Exit Through PLC or I²C Control	0x0: UART enter and exit controlled by PLC. 0x1: UART enter and exit controlled by I²C.
i2c_urt_ena1	4	PLC1 Enter UART Mode (I²C)	0x0: Not enter UART mode. 0x1: Enter UART mode.
i2c_urt_abr1	3	PLC1 Abort UART Mode (I²C)	0x0: Not abort 0x1: Abort
i2c_urt_swc1	2	PLC1 UART Switch PLC or I²C Control	0x0: PLC control 0x1: I²C control
i2c_tx__swc1	1	PLC1 TX Switch I²C Enable	0x0: Open 0x1: Closed
i2c_rx__swc1	0	PLC1 RX Switch I²C Enable	0x0: Open 0x1: Closed

UART_Ctr2 (0x22)

BIT	7	6	5	4	3	2	1	0
Field	pl2_urt_dis	tmo_tmr_ena2	i2c_urt_mod2	i2c_urt_ena2	i2c_urt_abr2	i2c_urt_swc2	i2c_tx__swc2	i2c_rx__swc2
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
pl2_urt_dis	7	PLC2 abort UART mode if entered from UART_REQ PLC command.	0x0: Not abort 0x1: Abort
tmo_tmr_ena2	6	PLC2 UART Timeout	0x0: Disable 0x1: Enable
i2c_urt_mod2	5	PLC1 UART Enter/Exit Through PLC or I²C Control	0x0: UART enter and exit controlled by PLC. 0x1: UART enter and exit controlled by I²C.
i2c_urt_ena2	4	PLC2 Enter UART Mode (I²C)	0x0: Not enter UART mode. 0x1: Enter UART mode.
i2c_urt_abr2	3	PLC2 Abort UART Mode (I²C)	0x0: Not abort 0x1: Abort
i2c_urt_swc2	2	PLC2 UART Switch PLC or I²C Control	0x0: PLC control 0x1: I²C control
i2c_tx__swc2	1	PLC2 TX Switch I²C Enable	0x0: Open 0x1: Closed
i2c_rx__swc2	0	PLC2 RX Switch I²C Enable	0x0: Open 0x1: Closed

SYSTEM_CONFIG0 (0x30)

BIT	7	6	5	4	3	2	1	0

Field	–	low_pwr_ena	–	–	–	–	–	–
Reset	–	0x1	–	–	–	–	–	–
Access Type	–	Write, Read	–	–	–	–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
low_pwr_ena	6	Low-Power Mode Enable	0x0: Disable 0x1: Enable

PLC_CONFIG0 (0x31)

BIT	7	6	5	4	3	2	1	0
Field	PLCSnkSel[1:0]		–	–	–	–	PLCThrSel[1:0]	
Reset	0x1		–	–	–	–	0x1	
Access Type	Write, Read		–	–	–	–	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
PLCSnkSel	7:6	PLC Transmission Sink Current	0x0: 88mA 0x1: 105.4mA 0x2: 123mA 0x3: 140.5mA
PLCThrSel	1:0	PLC Pulse Voltage Threshold	0x0: -40mV 0x1: -56mV 0x2: -70mV 0x3: -84mV

PLC_CONFIG1 (0x32)

BIT	7	6	5	4	3	2	1	0
Field	–	–	plc2_iprog[2:0]			plc1_iprog[2:0]		
Reset	–	–	0x6			0x6		
Access Type	–	–	Write, Read			Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
plc2_iprog	5:3	PLC2 Overcurrent Threshold	0x0: 100mA 0x1: 150mA 0x2: 200mA 0x3: 250mA 0x4: 300mA 0x5: 350mA 0x6: 400mA 0x8: NOT USED
plc1_iprog	2:0	PLC1 Overcurrent Threshold	0x0: 100mA 0x1: 150mA

BITFIELD	BITS	DESCRIPTION	DECODE
			0x2: 200mA 0x3: 250mA 0x4: 300mA 0x5: 350mA 0x6: 400mA 0x7: NOT USED

PLC_CONFIG2 (0x33)

BIT	7	6	5	4	3	2	1	0
Field	pl2_chn_ena	pl1_chn_ena	pl2_res_req	pl1_res_req	png_rty_num[2:0]			plc_double
Reset	0x1	0x1	0x0	0x0	0x2			0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read			Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
pl2_chn_ena	7	PLC2 Channel Enable	0x0: Disabled 0x1: Enabled
pl1_chn_ena	6	PLC1 Channel Enable	0x0: Disabled 0x1: Enabled
pl2_res_req	5	PLC2 resume request from PLC IDLE mode. Autocleared when done.	0x0: No request. 0x1: Request to resume from PLC IDLE.
pl1_res_req	4	PLC1 resume request from PLC idle mode. Autocleared when done.	0x0: No request. 0x1: Request to resume from IDLE.
png_rty_num	3:1	Telemetry PING Retry Number. This is the number of retry master handles in case of error in PING. It applies to all PLC commands other than bulk data transfer (DOUT_REQ). Master turns off buck-boost and disconnects if all the retry attempts are failed.	0x0: 0 0x1: 1 0x2: 2 0x3: 3 0x4: 4 0x5: 5 0x6: 6 0x7: 7
plc_double	0	Toggle between dual-slave and mono-slave mode.	0x0: Dual-slave 0x1: Mono-slave

PLC_CONFIG3 (0x34)

BIT	7	6	5	4	3	2	1	0
Field	plc_rrt_tmr[7:0]							
Reset	0x18							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
plc_rrt_tmr	7:0	Telemetry PING Period. Do not set to be less than 100ms for normal operation.	LSB: 4ms. Telemetry PING period = plc_rrt_tmr x 4ms.

PLC_CONFIG4 (0x35)

BIT	7	6	5	4	3	2	1	0
Field	plc_config[7:0]							
Reset	0xFF							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
plc_config	7:0	Bulk Transfer Auto-Retry of DOUT_REQ. Maximum NACK to DOUT_REQ from slave before master detects an error in DATA transfers. Only works in bulk data transfer mode.	Number of retries = plc_config + 1. When it is set to 0xFF, master retries indefinitely.

PLC_CONFIG5 (0x36)

BIT	7	6	5	4	3	2	1	0
Field	plc_is_full	RAM_is_full	0	0	0	0	–	cont_stream
Reset	0x0	0x0	0x0	0x0	0x0	0x0	–	0x0
Access Type	Read Only	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	–	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
plc_is_full	7	Valid only for bulk data transfer. It indicates the RAM buffer is full. Data received through PLC is not read yet. Device will NACK to the next PLC DOUT_REQ.	0x0: At least one RAM is free. 0x1: Both RAMs are full.
RAM_is_full	6	Valid for both bulk data transfer and mailbox data transfer. When data is received, this bit is set together with plc_new_dat and plc_rx_bytes. RAM_is_full remains at 1 until write 1 from I²C to clear. Make sure to clear RAM_is_full after PLC data is read through I²C. Writing RAM_is_full to one signals to the device that RAM is ready to receive a new data packet. If the device receives a new DOUT_REQ but RAM_is_full remains at 1, the device will NACK to DOUT_REQ.	0x0: No new data received on the PLC line. 0x1: New data received through the PLC line, Ready to be read. Write to 1 to clear.

BITFIELD	BITS	DESCRIPTION	DECODE
0	5	Reserved	Keep 0. Do not change.
0	4	Reserved	Keep 0. Do not change.
0	3	Reserved	Keep 0. Do no change.
0	2	Reserved	Keep 0. Do not change.
cont_stream	0	Valid only for bulk data transfer. If the device is sending data in bulk transfer mode, when this bit is set to 1, only the first packet transfer must be triggered with plc_run_trg1/2. Other packets are sent automatically after plc_cmd_dne interrupt. A new data packet needs to filled in to RAM through I^2C before plc_cmd_dne of previous DOUT_REQ command.	0x0: No continuous stream. 0x1: Continuos stream.

PLC_CONFIG6 (0x37)

BIT	7	6	5	4	3	2	1	0
Field	swp_plc_ram	–	no_uart_mde	no_idle_mde	–	DAT_MAX_RTY[2:0]		
Reset	0x0	–	0x0	0x0	–	0x7		
Access Type	Write, Read	–	Write, Read	Write, Read	–	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
swp_plc_ram	7	Swap PLC RAM Control. Set this bit to 1 to read back data filled to RAM from master SoC in bulk data transfer. Make sure to keep this bit at 0 when triggering data transfer.	0x0: Automatic control. 0x1: RAM readback on I^2C.
no_uart_mde	5	NACK to UART_REQ PLC Command	0x0: Disable. Send ACK to UART_REQ 0x1: Enable. Send NACK to UART_REQ
no_idle_mde	4	NACK to IDLE_REQ System Request	0x0: Disable. Send ACK to IDLE_REQ 0x1: Enable. Send NACK to IDLE_REQ
DAT_MAX_RTY	2:0	Auto-Retry Numbers of Data Packet	0x0: 1 0x1: 2 0x2: 3 0x3: 4 0x4: 5 0x5: 6 0x6: 7 0x7: 8

PLC_ARG1 (0x38)

BIT	7	6	5	4	3	2	1	0
Field	plc_cmd_arg1[7:0]							

Reset	0x0
Access Type	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
plc_cmd_arg1	7:0	PLC1 Command Argument	**SET_GPIO:** {GPIOEnRes4, GPIOEnRes3, GPIOEnRes2, GPIOEnRes1, GPIODout_4, GPIODout_3, GPIODout_2, GPIODout_1} **DOUT_REQ:** Number of bytes = plc_cmd_arg1 + 1 **UART_REQ:** 0x0: Manual configuration of UART direction through I^2C command (All switches are OFF) 0x1: Master receiving, slave transimitting (RX switches are ON) 0x2: Master transmitting, slave receiving (TX switches are ON) 0x3: Local loop back (TX and RX switches are ON) **SYST_REQ:** 0x0: Seal request 0x1: Soft reset 0x2: Hard reset 0x3: Fuel gauge reset 0x4: FIFO request 0x5: Free request 0x6: PLC IDLE mode request 0x7: Hard + soft reset {0x3F, slave_uart_tx, slave_uart_rx}: Hard reset and put slave into UART mode

PLC_CMD1 (0x39)

BIT	7	6	5	4	3	2	1	0
Field	plc_run_trg1	plc_command1[6:0]						
Reset	0x0	0x0						
Access Type	Write, Read	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
plc_run_trg1	7	PLC1 Command Trigger. Write 1 to trigger a PLC command. Autocleared when command is sent out through the PLC line during PING.	0x0: No command in progress. 0x1: Command running.
plc_command1	6:0	PLC1 Command	0x0: SYST_REQ 0x3: SET_GPIO 0x5: DOUT_REQ 0x6: UART_REQ

PLC_ARG2 (0x3A)

BIT	7	6	5	4	3	2	1	0
Field	plc_cmd_arg2[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
plc_cmd_arg2	7:0	PLC2 Command Argument	**SET_GPIO:** {GPIOEnRes4, GPIOEnRes3, GPIOEnRes2, GPIOEnRes1, GPIODout_4, GPIODout_3, GPIODout_2, GPIODout_1} **DOUT_REQ:** Number of bytes = plc_cmd_arg2 + 1 **UART_REQ:** 0x0: Manual configuration of UART direction through I^2C command (All switches are OFF) 0x1: Master receiving, slave transimitting (RX switches are ON) 0x2: Master transmitting, slave receiving (TX switches are ON) 0x3: Local loop back (TX and RX switches are ON) **SYST_REQ:** 0x0: Seal request 0x1: Soft reset 0x2: Hard reset 0x3: Fuel gauge reset 0x4: FIFO request 0x5: Free request 0x6: PLC IDLE mode request 0x7: Hard + soft reset {0x3F, slave_uart_tx, slave_uart_rx}: Hard reset and put slave into UART mode

PLC_CMD2 (0x3B)

BIT	7	6	5	4	3	2	1	0
Field	plc_run_trg2	plc_command2[6:0]						
Reset	0x0	0x0						
Access Type	Write, Read	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
plc_run_trg2	7	PLC2 Command Trigger. Write 1 to trigger a PLC command. Autocleared when command is sent out through PLC line during PING.	0x0: No command in progress. 0x1: Command running
plc_command2	6:0	PLC2 Command	0x0: SYST_REQ 0x3: SET_GPIO 0x5: DOUT_REQ 0x6: UART_REQ

PLC_RX (0x3C)

BIT	7	6	5	4	3	2	1	0
Field	rx__pl1_pl2	plc_rx_bytes[6:0]						
Reset	0x0	0x0						
Access Type	Read Only	Read Only						

BITFIELD	BITS	DESCRIPTION	DECODE
rx__pl1_pl2	7	PLC channel that receives incoming data.	0x0: PLC channel 1 0x1: PLC channel 2
plc_rx_bytes	6:0	Number of bytes received on the PLC line during last DOUT_REQ command. This register is set together with plc_new_dat interrupt and RAM_is_full flag.	Number of bytes = plc_rx_bytes + 1.

PLC_FIFO (0x3D)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	pl2_fifo_master	pl2_fifo_slave	pl1_fifo_master	pl1_fifo_slave
Reset	–	–	–	–	0x0	0x0	0x0	0x0
Access Type	–	–	–	–	Read Only	Read Only	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
pl2_fifo_master	3	PLC2 FIFO Master	0x0: Not in FIFO master mode. 0x1: FIFO master mode.
pl2_fifo_slave	2	PLC1 FIFO Slave	0x0: Not in FIFO slave mode. 0x1: FIFO slave mode.
pl1_fifo_master	1	PLC1 FIFO Master	0x0: Not in FIFO master mode. 0x1: FIFO master mode.
pl1_fifo_slave	0	PLC1 FIFO Slave	0x0: Not in FIFO slave mode. 0x1: FIFO slave mode.

BB_UP_DOWN (0x40)

BIT	7	6	5	4	3	2	1	0
Field	up_down_val[2:0]			i2c_frc_def	1	1	bb__alg_max	bb__alg_min
Reset	0x0			0x0	0x1	0x1	0x0	0x0
Access Type	Write, Read			Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
up_down_val	7:5	Buck-boost output voltage minimum ramp-up/down step with up/down command.	0x0: ±1 0x1: ±2 0x3: ±4 0x4: ±5 0x5: ±6 0x6: ±7 0x7: ±8
i2c_frc_def	4	Force buck-boost output voltage to bb_vlt__def.	0x0: BB voltage tracks slave battery voltage. 0x1: Force to bb_vlt_def
1	3	Reserved	Keep 1. Do not change.
1	2	Reserved	Keep 1. Do not change.
bb__alg_max	1	Maximum Tracking Algorithm. If both bb__alg_max and bb__alg_min are 1, or both are 0, optimized tracking algorithm is selected.	bb__alg_min = 0 0x0: Optimized tracking 0x1: Maximum tracking bb__alg_min = 1 0x0: Minimum tracking 0x1: Optimized tracking
bb__alg_min	0	Minimum Tracking Algorithm. If both bb__alg_max and bb__alg_min are 1, or both are 0, optimized algorithm is selected.	bb__alg_max = 0 0x0: Optimized tracking 0x1: Minimum tracking bb__alg_max = 1 0x0: Maximum tracking 0x1: Optimized tracking

BB_VOLT_DEF (0x41)

BIT	7	6	5	4	3	2	1	0
Field	bb_vlt__def[7:0]							
Reset	0x49							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
bb_vlt__def	7:0	Buck-Boost Default Voltage. Buck-boost converter pulls PLC voltage to the default voltage after master and slave build connection. It is the starting value to track slave battery voltage. This is also the minimum value that PLC voltage can reach.	Buck-boost default voltage = 2.5V + bb_vlt_def x 13.7mV. Capped at 5.5V.

BB_VLT_TRAN (0x42)

BIT	7	6	5	4	3	2	1	0
Field	bb_vlt_tran[7:0]							
Reset	0xFF							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
bb_vlt_tran	7:0	Buck-Boost Communication Voltage. Buck-boost pulls PLC voltage to communication voltage for PLC communication periodically. The period is set by plc_rrt_tmr.	Buck-boost communication voltage = 2.5V + bb_vlt_tran x 13.7mV. Capped at 5.5V.

BB_RMP_CFG3 (0x46)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	min_ovl_sel[2:0]		
Reset	–	–	–	–	–	0x7		
Access Type	–	–	–	–	–	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
min_ovl_sel	2:0	OVLO Threshold for Buck Boost	0x0: 4.8 0x1: 4.9 0x2: 5.0 0x3: 5.1 0x4: 5.2 0x5: 5.3 0x6: 5.4 0x7: 5.5

BB_ANA_CFG1 (0x47)

BIT	7	6	5	4	3	2	1	0

Field	bb_psv_dchg	bb_act_dchg	0	0	0	0	11[1:0]
Reset	0x1	0x1	0x0	0x0	0x0	0x0	0x3
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
bb_psv_dchg	7	Buck-Boost Passive Discharge Control	0x0: Buck-boost passive discharged not performed. 0x1: Buck-boost passively discharged when Enable is low.
bb_act_dchg	6	Buck-Boost Active Discharge Control	0x0: Buck-boost active discharged not performed. 0x1: Buck-boost actively discharged when Enable is low.
0	5	Reserved	Keep 0. Do not change.
0	4	Reserved	Keep 0. Do not change.
0	3	Reserved	Keep 0. Do not change.
0	2	Reserved	Keep 0. Do not change.
11	1:0	Reserved	Keep 0b11. Do not change.

BB_ANA_CFG2 (0x48)

BIT	7	6	5	4	3	2	1	0
Field	0	bb__frc_ena	–	–	bb__low__bw	bb___l2uh	i2c_zccm_enb	–
Reset	0x0	0x0	–	–	0x0	0x1	0x1	–
Access Type	Write, Read	Write, Read	–	–	Write, Read	Write, Read	Write, Read	–

BITFIELD	BITS	DESCRIPTION	DECODE
0	7	Reserved	Keep 0. Do not change.
bb__frc_ena	6	Force the buck-boost to be active.	0x0: Disable 0x1: Enable
bb__low__bw	3	Murata Output Cap.	0x0: Use other cap. 0x1: Use Murata output cap.
bb___l2uh	2	Buck-Boost Inductor Value	0x0: 1µH 0x1: 2.2µH
i2c_zccm_enb	1	Buck-Boost Zero-Crossing Comparator Control	0x0: Enable 0x1: Disable

BB_ALG1 (0x4A)

BIT	7	6	5	4	3	2	1	0
Field	avg_t_delta[7:0]							

Reset	0xA
Access Type	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
avg_t_delta	7:0	Threshold to switch from minimum to maximum tracking if optimized algorithm is choosen.	Min to max threshold = avg_t_delta x 20mV

BB_MULTI (0x4B)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	up__dwn_dis	multi_up_en	multi_dw_en
Reset	–	–	–	–	–	0x0	0x1	0x1
Access Type	–	–	–	–	–	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
up__dwn_dis	2	Disable the up/down function. When up/down is disabled, buck-boost voltage is set to bb_vlt__def.	0x0: Enable 0x1: Disable
multi_up_en	1	Buck-boost output ramp-up step is incremented by 1 LSB if consecutive up command is received. Minimum step is set by up_down_val. Cap to 15 LSBs.	0x0: Disable 0x1: Enable
multi_dw_en	0	Buck-boost output ramp-down step is incremented by 1 LSB if consecutive down command is received. Minimum step is set by up_down_val. Cap to 15 LSBs.	0x0: Disable 0x1: Enable

GPIO1 (0x58)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIOPLCCtr1	GPIOEnRes1	GPIOEnPup1	GPIODout_1
Reset	–	–	–	–	0x1	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
GPIOPLCCtr1	3	GPIO I²C or PLC Control	0x0: I²C 0x1: PLC

BITFIELD	BITS	DESCRIPTION	DECODE
GPIOEnRes1	2	GPIO Input or Output Configuration	0x0: Output 0x1: Input
GPIOEnPup1	1	GPIO Input Pullup or Pulldown Resistor. Active if GPIOEnRes =1.	0x0: Internal pulldown 0x1: Internal pullup
GPIODout_1	0	GPIO Open-Drain Output. Needs external pullup.	0x0: High 0x1: Low

GPIO2 (0x59)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIOPLCCtr2	GPIOEnRes2	GPIOEnPup2	GPIODout_2
Reset	–	–	–	–	0x1	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
GPIOPLCCtr2	3	GPIO I²C or PLC Control	0x0: I²C 0x1: PLC
GPIOEnRes2	2	GPIO Input or Output Configuration	0x0: Output 0x1: Input
GPIOEnPup2	1	GPIO Input Pullup or Pulldown Resistor. Active if GPIOEnRes =1.	0x0: Internal pulldown 0x1: Internal pullup
GPIODout_2	0	GPIO Open-Drain Output. Needs external pullup.	0x0: High 0x1: Low

GPIO3 (0x5A)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIOPLCCtr3	GPIOEnRes3	GPIOEnPup3	GPIODout_3
Reset	–	–	–	–	0x1	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
GPIOPLCCtr3	3	GPIO I²C or PLC control	0x0: I²C 0x1: PLC
GPIOEnRes3	2	GPIO Input or Output Configuration	0x0: Output 0x1: Input
GPIOEnPup3	1	GPIO Input Pullup or Pulldown Resistor. Active if GPIOEnRes =1.	0x0: Internal pulldown 0x1: Internal pullup
GPIODout_3	0	GPIO Open-Drain Output. Needs external pullup.	0x0: High 0x1: Low

GPIO4 (0x5B)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIOPLCCtr4	GPIOEnRes4	GPIOEnPup4	GPIODout_4
Reset	–	–	–	–	0x1	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
GPIOPLCCtr4	3	GPIO I²C or PLC Control	0x0: I²C 0x1: PLC
GPIOEnRes4	2	GPIO Input or Output Configuration	0x0: Output 0x1: Input
GPIOEnPup4	1	GPIO Input Pullup or Pulldown Resistor. Active if GPIOEnRes =1.	0x0: Internal pulldown 0x1: Internal pullup
GPIODout_4	0	GPIO Open-Drain Output. Needs external pullup.	0x0: High 0x1: Low

GPIO_rdb1 (0x5C)

BIT	7	6	5	4	3	2	1	0
Field	GPIODAInp_4	GPIODAInp_3	GPIODAInp_2	GPIODAInp_1	–	–	–	GPIOCmosEn
Reset	0x0	0x0	0x0	0x0	–	–	–	0x0
Access Type	Read Only	Read Only	Read Only	Read Only	–	–	–	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
GPIODAInp_4	7	GPIO Input Status	0x0: Low 0x1: High
GPIODAInp_3	6	GPIO Input Status	0x0: Low 0x1: High
GPIODAInp_2	5	GPIO Input Status	0x0: Low 0x1: High
GPIODAInp_1	4	GPIO Input Status	0x0: Low 0x1: High
GPIOCmosEn	0	GPIO Input Logic Threshold	0x0: 1.8V VDIG 0x1: VCCINT

GPIO_rdb2 (0x5D)

BIT	7	6	5	4	3	2	1	0
Field	PLC2_GPIO4	PLC2_GPIO3	PLC2_GPIO2	PLC2_GPIO1	PLC1_GPIO4	PLC1_GPIO3	PLC1_GPIO2	PLC1_GPIO1
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
PLC2_GPIO4	7	PLC2 Slave GPIO4 Status	0x0: Low 0x1: High
PLC2_GPIO3	6	PLC2 Slave GPIO3 Status	0x0: Low 0x1: High
PLC2_GPIO2	5	PLC2 Slave GPIO2 Status	0x0: Low 0x1: High
PLC2_GPIO1	4	PLC2 Slave GPIO1 Status	0x0: Low 0x1: High
PLC1_GPIO4	3	PLC1 Slave GPIO4 Status	0x0: Low 0x1: High
PLC1_GPIO3	2	PLC1 Slave GPIO3 Status	0x0: Low 0x1: High
PLC1_GPIO2	1	PLC1 Slave GPIO2 Status	0x0: Low 0x1: High
PLC1_GPIO1	0	PLC1 Slave GPIO1 Status	0x0: Low 0x1: High

SOC_byte_1 (0x60)

BIT	7	6	5	4	3	2	1	0
Field	SOC_byte_1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
SOC_byte_1	7:0	SOC byte1 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 0.00390625% of full charge. The full-scale is 256%.

SOC_byte_0 (0x61)

BIT	7	6	5	4	3	2	1	0
Field	SOC_byte_0[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
SOC_byte_0	7:0	SOC byte0 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 0.00390625% of full charge. The full-scale is 256%.

VCELL_byte_1 (0x62)

BIT	7	6	5	4	3	2	1	0
Field	VCELL_byte_1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
VCELL_byte_1	7:0	Cell Voltage byte1 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 78.125µV. The full-scale is 5.12V.

VCELL_byte_0 (0x63)

BIT	7	6	5	4	3	2	1	0
Field	VCELL_byte_0[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
VCELL_byte_0	7:0	Cell Voltage byte0 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 78.125µV. The full-scale is 5.12V.

TTE_byte_1 (0x64)

BIT	7	6	5	4	3	2	1	0
Field	TTE_byte_1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
TTE_byte_1	7:0	Time-to-Empty byte1 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 5.625s. The full-scale is 102.4 hours.

TTE_byte_0 (0x65)

BIT	7	6	5	4	3	2	1	0
Field	TTE_byte_0[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
TTE_byte_0	7:0	Time-to-empty byte0 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 5.625s. The full-scale is 102.4 hours.

AVGVCELL_byte_1 (0x66)

BIT	7	6	5	4	3	2	1	0
Field	AVGVCELL_byte_1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
AVGVCELL_byte_1	7:0	Average Cell Voltage byte1 Readback from Fuel Gauge. It is the moving average of the last 45 seconds.	Concatenate byte1 and byte0: LSB is 78.125µV. The full-scale is 5.12V.

AVGVCELL_byte_0 (0x67)

BIT	7	6	5	4	3	2	1	0
Field	AVGVCELL_byte_0[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
AVGVCELL_byte_0	7:0	Average Cell Voltage byte0 Readback from Fuel Gauge. It is the moving average of the last 45 seconds.	Concatenate byte1 and byte0: LSB is 78.125uV. The full-scale is 5.12V.

TTF_byte_1 (0x68)

BIT	7	6	5	4	3	2	1	0
Field	TTF_byte_1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
TTF_byte_1	7:0	Time-to-Full byte1 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 5.625s. The full-scale is 102.4 hours.

TTF_byte_0 (0x69)

BIT	7	6	5	4	3	2	1	0
Field	TTF_byte_0[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
TTF_byte_0	7:0	Time-to-Full byte0 Readback from Fuel Gauge	Concatenate byte1 and byte0: LSB is 5.625s. The full-scale is 102.4 hours.

READY_REG (0x6A)

BIT	7	6	5	4	3	2	1	0
Field	dop_rdy_sig	–	dop_i2c_ena	ttf_reg_rdy	avgvcell_reg_rdy	tte_reg_rdy	vcell_reg_rdy	soc_reg_rdy
Reset		–	0x1	0x0	0x0	0x0		
Access Type	Read Only	–	Write, Read	Read Only	Read Only	Read Only	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
dop_rdy_sig	7	DOP Port Ready	0x0: Not ready 0x1: Ready
dop_i2c_ena	5	Enable the DOP PORT Sniffer	0x0: Disable 0x1: Enable
ttf_reg_rdy	4	TTF Byte Value is Reliable	0x0: Not reliable 0x1: Reliable
avgvcell_reg_rdy	3	AVGVCELL Byte Value is Reliable	0x0: Not reliable 0x1: Reliable

BITFIELD	BITS	DESCRIPTION	DECODE
tte_reg_rdy	2	TTE Byte Value is Reliable	0x0: Not reliable 0x1: Reliable
vcell_reg_rdy	1	SOC Byte Value is Reliable	0x0: Not reliable 0x1: Reliable
soc_reg_rdy	0	SOC Byte Value is Reliable	0x0: Not reliable 0x1: Reliable

FG_RDY_1 (0x6B)

BIT	7	6	5	4	3	2	1	0
Field	slv_avg__1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
slv_avg__1	7:0	PLC1 Slave Average Battery Voltage byte1 from PLC Channel 1 Fuel Gauge. Valid if slv_avgrdy1 = 1.	LSB is 20mV. The full-scale is 5.12V.

FG_RDY_2 (0x6C)

BIT	7	6	5	4	3	2	1	0
Field	slv_soc__1[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
slv_soc__1	7:0	PLC1 Slave SOC byte1 from PLC Channel 1 Fuel Gauge. Valid if slv_socrdy1 = 1.	LSB is 1% of full charge. The full-scale is 256%.

FG_RDY_3 (0x6D)

BIT	7	6	5	4	3	2	1	0
Field	slv_avg__2[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
slv_avg__2	7:0	PLC2 Slave Average Battery Volatage byte1 from PLC Channel 2 Fuel Gauge. Valid if slv_avgrdy2 = 1.	LSB is 20mV. The full-scale is 5.12V.

FG_RDY_4 (0x6E)

BIT	7	6	5	4	3	2	1	0
Field	slv_soc__2[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
slv_soc__2	7:0	PLC2 Slave SOC byte1 from PLC Channel 2 Fuel Gauge. Valid if slv_socrdy2 = 1.	LSB is 1% of full charge. The full-scale is 256%.

FG_RDY_5 (0x6F)

BIT	7	6	5	4	3	2	1	0
Field	slv2_chg_dne	slv1_chg_dne	–	–	slv_socrdy2	slv_avgrdy2	slv_socrdy1	slv_avgrdy1
Reset	0x0	0x0	–	–				
Access Type	Read Only	Read Only	–	–	Read Only	Read Only	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
slv2_chg_dne	7	PLC2 Slave Charger Done Flag	0x0: Battery not fully charged. 0x1: Battery fully charged.
slv1_chg_dne	6	PLC1 Slave Charger Done Flag	0x0: Battery not fully charged. 0x1: Battery fully charged.
slv_socrdy2	3	PLC2 SOC is Reliable	0x0: Not reliable 0x1: Reliable
slv_avgrdy2	2	PLC2 Average Battery Volatge is Reliable	0x0: Not reliable 0x1: Reliable
slv_socrdy1	1	PLC1 SOC is Reliable	0x0: Not reliable 0x1: Reliable
slv_avgrdy1	0	PLC1 Average Battery Volatge is Reliable	0x0: Not reliable 0x1: Reliable

ADC_CTRL1 (0x70)

BIT	7	6	5	4	3	2	1	0
Field	–	AdcGndTrh[3:0]				ResDetRty[2:0]		
Reset	–	0x4				0x1		
Access Type	–	Write, Read				Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
AdcGndTrh	6:3	ADC Ground Threshold	LSB = 5.882mV
ResDetRty	2:0	Number of Resistive Measurement Retries. The device retries the measurement if one of the following conditions is true: 1. (maximum ADC reading = 0xFF) AND (pullup current = 1µA) AND (average ADC reading < (0xFF - ADCNoiseClampRng[5:0])) 2. (maximum ADC reading = 0xFF) AND (pullup current ≠ 1µA) If the condition is still true after this number of retries, the Abort result is reported.	0x0: No retry. >0x0: Number of retry attempts.

ADC_CTRL2 (0x71)

BIT	7	6	5	4	3	2	1	0
Field	–	–	AdcRng[5:0]					
Reset	–	–	0x9					
Access Type	–	–	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
AdcRng	5:0	ADC Shift Factor. It applies to resistive measurements used in moisture detection. This register must NOT be set lower than the default value.	LSB = 5.882mV

ADC_CTRL3 (0x72)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	AdcAvgNum[2:0]		
Reset	–	–	–	–	–	0x0		

Access Type	–	–	–	–	–	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
AdcAvgNum	2:0	Number of Samples in ADC Reading Averaging. It applies to any resistive measurements used in moisture detection and accessory mode detection.	0x0: 1 sample 0x1: 2 samples 0x2: 4 samples 0x3: 8 samples 0x4: 16 samples 0x5: 32 samples 0x6: 64 samples 0x7: 128 samples

ADC_CTRL4 (0x73)

BIT	7	6	5	4	3	2	1	0
Field	–	–	AdcNoiseCtr[5:0]					
Reset	–	–	0x0					
Access Type	–	–	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
AdcNoiseCtr	5:0	ADC Result Margin to Account for External Noise and Avoid Result Clamping Close to Full-Scale. This register must NOT be changed from the default value.	LSB = 5.882mV

MOI_DET_REG1 (0x74)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	–	RaccDetMIp[1:0]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
RaccDetMIp	1:0	Target current used to specify moisture resistance threshold.	0x0: 1µA 0x1: 4µA 0x2: 16µA 0x3: 64µA

MOI_DET_REG2 (0x75)

BIT	7	6	5	4	3	2	1	0

Field	RaccDetMThr[7:0]
Reset	0x77
Access Type	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RaccDetMThr	7:0	Moisture Detection Voltage Threshold	LSB = 5.882mV.

MOI_DET_REG3 (0x76)

BIT	7	6	5	4	3	2	1	0
Field	moi_det_aut2	moi_det_aut1	moi_man_pl2	moi_man_pl1	moi_man_rty2	moi_man_rty1	moi_aut_rty2	moi_aut_rty1
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x1
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
moi_det_aut2	7	Enable automatic moisture detection on PLC2, performed with 16s rate starting 16s after device turn on.	0x0: Disabled 0x1: Enabled
moi_det_aut1	6	Enable for automatic moisture detection on PLC1, performed with 16s rate starting 8s after device turn on.	0x0: Disabled 0x1: Enabled
moi_man_pl2	5	Manual moisture request on PLC2. Write 1 to launch manual moisture detection. Autoclearing when done.	0x0: No moisture in progress. 0x1: Moisture in progress.
moi_man_pl1	4	Manual moisture request on PLC1. Write 1 to launch manual moisture detection. Autoclearing when done.	0x0: No moisture in progress. 0x1: Moisture in progress.
moi_man_rty2	3	Manual retry for master detection in case of moisture found. Only valid when moisture is previously found. Write 1 to launch moisture retry. This allows the master detection to run moisture detection again. Autoclearing when retry is done.	0x0: No retry request. 0x1: Retry request running.
moi_man_rty1	2	Manual retry for master detection in case of moisture found. Only valid when moisture is previously found. Write 1 to launch moisture retry. This allows the master detection to run moisture detection again. Autoclearing when retry is done.	0x0: No retry request. 0x1: Retry request running.
moi_aut_rty2	1	Enable the automatic re-arm of mositure detection for PLC detection if moisture is found. System re-	0x0: Disable 0x1: Enabled

BITFIELD	BITS	DESCRIPTION	DECODE
		arm moisture automatically when moisture disappears.	
moi_aut_rty1	0	Enable the automatic re-arm of mositure detection for PLC detection if moisture is found. System re-arm moisture automatically when moisture disappears.	0x0: Disable 0x1: Enabled

IP (0x78)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	–	IP__RES_DET[1:0]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Read Only	

BITFIELD	BITS	DESCRIPTION	DECODE
IP__RES_DET	1:0	Final current used when moisture measurement is complete. It is set to 0b00 if the result is Abort or Open. It is set to 0b11 if the result is Short.	0x0: 1µA 0x1: 4µA 0x2: 16µA 0x3: 64µA

ADC_VAL1 (0x79)

BIT	7	6	5	4	3	2	1	0
Field	ADCAvg[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
ADCAvg	7:0	Final Average ADC Reading. It is set to 0x00 if the result is Abort or Ground, and to 0xFF if the result is Open.	LSB = 5.882mV = 0.392% typ.

ADC_VAL2 (0x7A)

BIT	7	6	5	4	3	2	1	0
Field	ADCMax[7:0]							

Reset	0x0
Access Type	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
ADCMax	7:0	Final Maximum ADC Reading. It is set to 0x00 if the result is Abort or Ground, and to 0xFF if the result is Open.	LSB = 5.882mV = 0.392% typ.

ADC_VAL3 (0x7B)

BIT	7	6	5	4	3	2	1	0
Field	ADCMin[7:0]							
Reset	0x0							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION	DECODE
ADCMin	7:0	Final Minimum ADC Reading. It is set to 0x00 if the result is Abort or Ground, and to 0xFF if the result is Open.	LSB = 5.882mV = 0.392% typ.

Ordering Information

PART NUMBER	TEMP RANGE	PIN-PACKAGE
MAX20355EWO+	-40°C to +85°C	42-WLP
MAX20355EWO+T	-40°C to +85°C	42-WLP

+Denotes a lead(Pb)-free/RoHS-compliant package.

T = Tape and reel.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	06/22	Release for Market Intro	—