MAX1720x/MAX1721x Software Implementation Guide

UG6260; Rev 0; 2/16

Description

In this guide are details for implementing the host software to interface with the MAX1720x/MAX1721x fuel gauge ICs.

Table 1. Device Type Versus Cell Count and Communication Interface

Device Type	Cell Count	Communication Interface
MAX17201	Single Cell	I^2C
MAX17211	Single Cell	1-Wire
MAX17205	Multi Cell	I^2C
MAX17215	Multi Cell	1-Wire

To distinguish the type of device, read the DevName register (0x021). The low nibble indicates 0x1 for MAX172x1, and 0x5 for MAX172x5. The 1-Wire®/I^2C distinction is made by a device present on one interface, and not the other. **Table 1** shows which IC type should be used depending on the application requirements.

Initialization

At IC power-up, the fuel gauge retrieves battery model parameters from nonvolatile memory without host software interaction.

Optionally, the host can change some of the auxiliary functions, such as alerts on voltage, current, SOC, or temperature. If the nonvolatile programmed values are acceptable, then the host only needs to take action on the alert outputs. If the host desires different thresholds, the host can write to the VAlrt_Th, TAlrt_Th, SAlrt_Th, IAlrt_Th, ODSCCfg, or ODSCCfg registers as necessary. Please refer to the data sheet for register details.

Useful Registers to Read

The following is a short list of registers the host finds useful in most applications.

Address	Register Name	Purpose/Contents
0x000	Status	Contains alert status and chip status
0x009	VCell	Lowest cell voltage of a pack, or the cell voltage for a single cell
0x006	RepSOC	Reported state of charge
0x005	RepCap	Reported remaining capacity
0x008	Temp	Temperature
0x00A	Current	Battery current
0x011	TTE	Time to empty
0x020	TTF	Time to full

Other applications can have more requirements. Additional registers to read are:

Address	Register Name	Purpose/Contents
0x0D8, 0x0D7, 0x0D6, 0x0D9, 0x0DA	Cell1, Cell2, Cell3, CellX, Batt	Direct cell measurements if channel mode is enabled. Batt register contains the total pack voltage.
0x134, 0x13B, 0x135	Temp1, Temp2, IntTemp	Individual temperature measurements from the thermistors and internal die temperature
0x0B9	AgeForecast	Projected total cycles the battery lasts until full capacity is below the target "dead" battery capacity
0x007	Age	Provides a percentage of full capacity compared to the design capacity
0x017	Cycles	The number of cycles the battery has been used
0x0DD (AtRate=0x004)	AtTTE	After writing AtRate, hypothetical time to empty based on the AtRate value
0x010	FullCap	Full capacity based on present discharge conditions
0x0BE	TimerH	Time since first power up
0x01B, 0x01C, 0x01A	MaxMinVolt, MaxMinCurrent, MaxMinTemp	Maximum and minimum values for each measurement since last life log
0x014	RCell	Calculated battery resistance
0x019, 0x00B, 0x016	AvgVCell, AvgCurr, AvgTA	Average of voltage, current, or temperature
0x0BC	VRipple	Filtered difference of average and instantaneous values of VCell

Alerts Management

The default configuration of the fuel gauge is to automatically clear the alert after the alert condition is no longer present. The host can set the alerts to be "sticky," and make them persist until the host clears the alert. This setting can be changed in nonvolatile memory to permanently alter the behavior to "sticky" mode by changing the nConfig register. This change can be made by the pack maker or the end application if the nonvolatile memory is not locked. Alternatively, it can be changed by the host software in the Config register to only change it until the fuel gauge resets, or the host wants to disable this feature. The Config register details are provided below to make this change.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	SS	TS	VS	ALRTp	AINSH	Ten	Tex	SHDN	COMMSH	0	ETHRM	FTHRM	Aen	Bei	Ber

Figure 1. Config Register (01Dh) format

The SS, TS, and VS bits set the matching alerts to be "sticky." The alerts are cleared by clearing the appropriate alert bit in the Status register.

Example to set low SOC alert to 5%:

1. Set SAlrtTh register to 0xFF05. 0xFF__ indicates the high alert is disabled. 0x__05 sets the low alert to 5%.
2. Set the Config.Aen bit to 1.
3. When the SOC decreases below 0x0500, the alert pin asserts, and the status register indicates the SOC alert has occurred.
4. To disable further alerts on low SOC until the battery is charged, set the SAlrtTh register to 0xFF00 to disable low SOC alerts.

Authentication

This IC has an authentication feature to prevent clone batteries. The host writes a challenge to the IC and reads out a response. The host validates this response by doing a parallel calculation with the same secret, or using predefined challenge and response pairs.

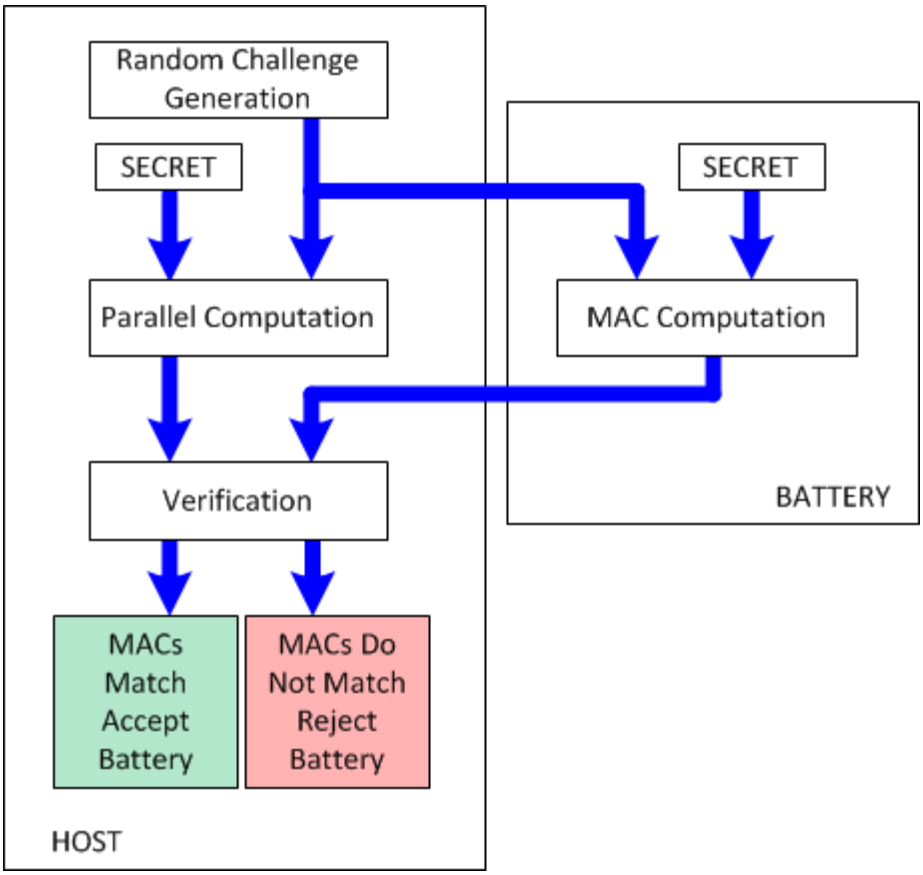

Figure 2. Method 1: Host knows the secret, and calculates a valid output to check against the battery pack response.

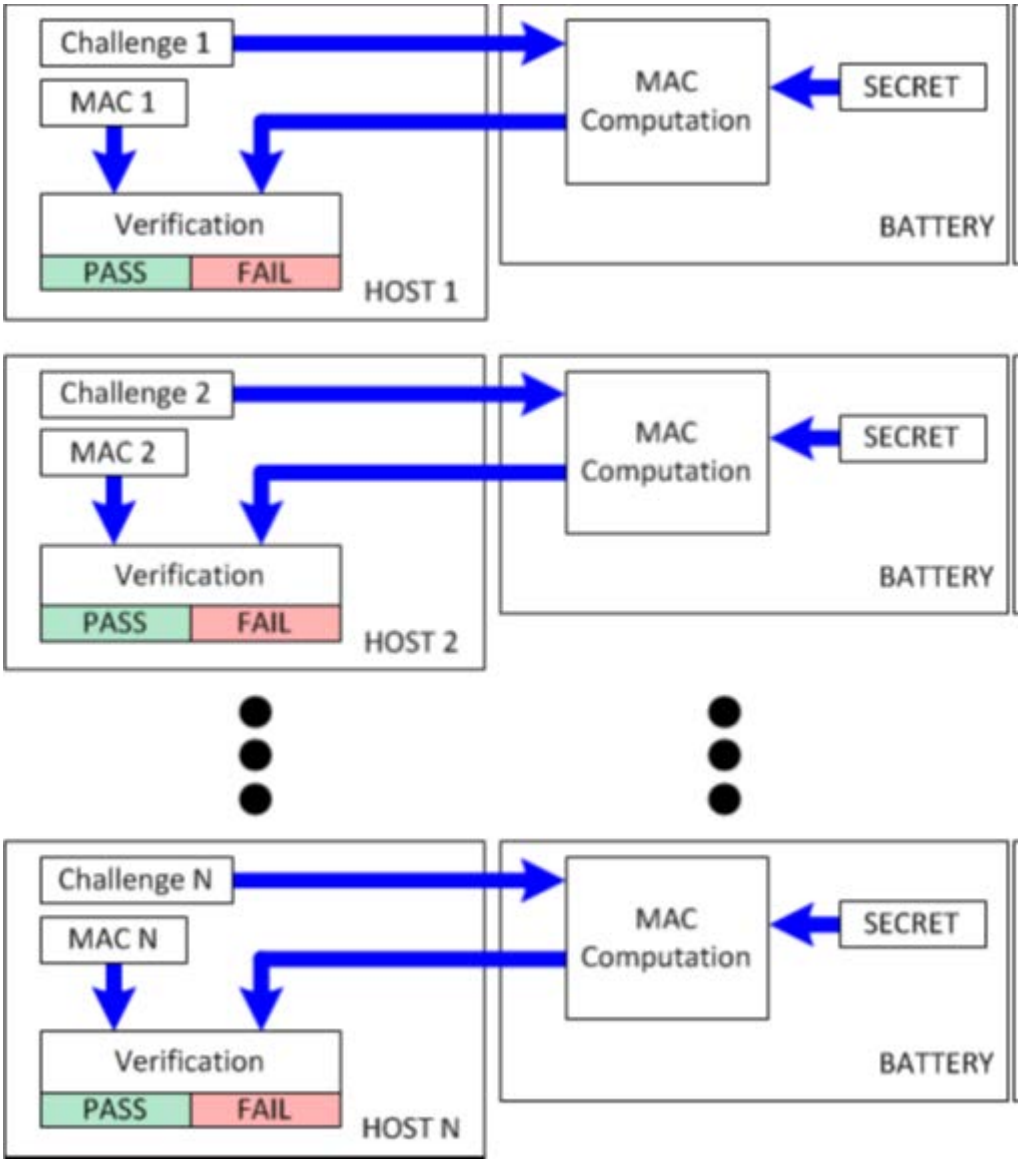

Figure 3. Method 2: The host does not know the secret. The challenge and response pairs are predefined for the host.

To authenticate the battery, the host has to write the challenge to memory addresses 0x0C0 to 0x0C9, and then send the Compute MAC without ROM ID command by writing register 0x060 with value 0x3600. The fuel gauge calculates a response after t_{SHA}, and the host can read it from space 0x0C0 to 0x0CF. The returned MAC computation is compared against the calculated or stored response, and the authenticity is verified.

If the host knows the secret, the Compute MAC with ROM ID command can be used for more security. Write register 0x060 with value 0x3500 to Compute MAC with ROM ID. The returned MAC computation is available after t_{SHA}.

Conclusion

This guide describes the basic operations the host software needs to interface with the MAX17201/MAX17205/MAX17211/MAX17215. There are many additional features not covered here that are listed in the data sheet. For any additional questions, contact Maxim Integrated.