



# MAX1726x ModelGauge m5 EZ User Guide

*UG6597; Rev 3; 11/19*

## Abstract

The MAX1726x series of low-power fuel gauge ICs implements Maxim ModelGauge™ m5 EZ algorithm. ModelGauge m5 EZ algorithm makes fuel gauge implementation easy by eliminating battery characterization requirements and simplifying host software interaction. This user guide provides an extended description of the MAX1726x series of fuel gauge ICs. It includes detailed descriptions of the register set and extended features, as well as application guidance.

## Table of Contents

Overview	8
OCV Estimation and Coulomb Count Mixing	9
Fuel Gauge Learning	10
Empty Compensation	12
Converge-to-Empty	12
Empty Hold and 99% Hold	13
SOCHold Register (D3h)	14
End-of-Charge Detection	14
Predicting Runtime of a Hypothetical Load	16
Lithium Iron Phosphate Support	16
ScOcvLim Register (D1h)	17
ModelGauge m5 Standard Register Formats	18
Analog Measurements	19
Current Measurements	19
CGain Register (2Eh) and COff Register (2Fh) Register	19
Copper Trace Current Sensing	19
Curve Register (B9h)	20
CGTempCo Register (B8h)	20
Temperature Measurements	20
AIN Register (27h)	21
TGain (2Ch) Register and TOff (2Dh) Register	21
Voltage Measurements	21
Dynamic Power	22
Dynamic Power Performance	22
Dynamic Power Output Registers	23
MPPCurrent Register (D9h)	23
SPPCurrent Register (DAh)	23
MaxPeakPower Register (D4h)	23
SusPeakPower Register (D5h)	23
Dynamic Power Configuration Registers	23
PackResistance Register (D6h)	23
SysResistance Register (D7h)	23
MinSysVoltage Register (D8h)	24
RGain Register (43h)	24

Serial Number Feature	25
Determining Fuel Gauge Accuracy	26
Initial Accuracy	26
ModelGauge m5 Registers	27
ModelGauge m5 Algorithm Battery Parameters	28
VEmpty Register (3Ah)	28
DesignCap Register (18h)	29
ModelCfg Register (DBh)	29
IChgTerm Register (1Eh)	29
FullSOCThr Register (13h)	30
OCVTable0 (80h) to OCVTable15 (8Fh) Registers	30
XTable0 (90h) to XTable15 (9Fh) Registers	30
QRTTable00 (12h) to QRTTable30 (42h) Registers	30
RComp0 Register (38h)	30
TempCo Register (39h)	30
ModelGauge m5 Output Registers	31
RepCap Register (05h)	31
AtAvCap Register (DFh)	31
RepSOC Register (06h)	31
AtAvSOC Register (DEh)	31
FullCapRep Register (10h)	31
FullCap Register (35h)	31
FullCapNom Register (23h)	31
TTE Register (11h)	32
AtTTE Register (DDh)	32
TTF Register (20h)	32
Cycles Register (17h)	32
Status Register (00h)	32
Age Register (07h)	33
TimerH (BEh) and Timer (3Eh) Register	33
RCell Register (14h)	34
VRipple Register (BCh)	34
ModelGauge m5 Algorithm Configuration Registers	34
AtRate Register (04h)	34
FilterCfg Register (29h)	34

RelaxCfg Register (2Ah)	35
LearnCfg Register (28h)	36
MiscCfg Register (2Bh)	36
ConvgCfg Register (49h)	37
RippleCfg Register (BDh)	37
ModelGauge m5 Algorithm Additional Registers	38
dQAcc Register (45h)	38
dPAcc Register (46h)	38
QResidual Register (0Ch)	38
AtQResidual Register (DCh)	38
VFSOC Register (FFh)	38
VFOCV Register (FBh)	38
QH Register (4Dh)	38
AvCap (1Fh) and AvSOC (0Eh) Registers	39
MixCap (0Fh) and MixSOC (0Dh) Registers	39
VFRemCap Register (4Ah)	39
FStat Register (3Dh)	39
Status and Configuration Registers	40
Config Register (1Dh) and Config2 Register (BBh)	40
DevName Register (21h)	40
ShdnTimer Register (3Fh)	40
Status2 Register (B0h)	41
HibCfg Register (BAh)	41
Soft-Wakeup (Command Register 60h)	42
Modes of Operation	43
I <sup>2</sup> C Bus System	44
Hardware Configuration	44
I/O Signaling	44
Bit Transfer	44
Bus Idle	45
START and STOP Conditions	45
Acknowledge Bits	45
Data Order	45
Slave Address	45
Read/Write Bit	45

Bus Timing	46
I <sup>2</sup> C Protocols	46
I <sup>2</sup> C Write Data Protocol	46
I <sup>2</sup> C Read Data Protocol	47
Trademarks	48
Revision History	49

## List of Figures

Figure 1. ModelGauge m5 flow diagram.	9
Figure 2. Error is filtered and bounded by a combination of ModelGauge and coulomb counter.	10
Figure 3. Voltage fuel gauge vs. coulomb-counter influence vs. first cycles.	10
Figure 4. ModelGauge m5 learns full capacity during arbitrary cycling.	11
Figure 5. ModelGauge m5 learns full capacity during arbitrary cycling.	12
Figure 6. MAX1726x converge-to-empty performance.	13
Figure 7. SOC hold conceptual drawing.	14
Figure 8. ModelGauge m5 rejects false end-of-charge events.	15
Figure 9. FullCapRep learn at end of charge.	16
Figure 10. Dynamic Power limits voltage spikes when the load is limited by MaxPeakCurrent.	22
Figure 11. ModelGauge m5 algorithm registers.	28
Figure 12. Cell relaxation detection.	36
Figure 13. I <sup>2</sup> C bus interface.	44
Figure 14. I <sup>2</sup> C bus timing diagram.	46

## List of Tables

Table 1. SOCHold (D3h) Format	14
Table 2. ScOcvLim (D1h) Format	17
Table 3. MAX17260/MAX17261/MAX17263 ModelGauge m5 Register Standard Resolutions	18
Table 4. MAX17262 ModelGauge m5 Register Standard Resolutions	18
Table 5. Measurement Range and Resolution vs. Sense Resistor Value for the MAX17260/MAX17261/MAX17263	19
Table 6. Curve (B9h) Format	20
Table 7. Register Settings for Common Thermistor Types	21
Table 8. RGain (43h) Format	24
Table 9. Accessing the Serial Number	25
Table 10. ModelGauge m5 Register Memory Map	27
Table 11. VEmpty (3Ah) Format	28
Table 12. ModelCFG (DBh) Format	29
Table 13. FullSOCThr (13h) Format	30
Table 14. Status (00h) Format	32
Table 15. FilterCfg (29h) Format	34
Table 16. RelaxCfg (2Ah) Format	35
Table 17. LearnCfg (28h) Format	36
Table 18. MiscCfg (2Bh) Format	37
Table 19. RippleCfg (BDh) Format	37
Table 20. FStat (3Dh) Format	39
Table 21. DevName Register Values for Different Variants of the MAX1726x	40
Table 22. ShdnTimer (3Fh) Format	40
Table 23. Status2 (B0h) Format	41
Table 24. HibCfg (BAh) Format	41

## Overview

This user guide provides an extended description of the MAX1726x series of fuel gauge ICs. It includes detailed descriptions of the register set and extended features, as well as application guidance. MAX1726x data sheets describe the basic feature sets and the minimal register set needed to support the plug-and-play ModelGauge™ m5 EZ performance.

MAX1726x series ICs are low-power 5μA operating current fuel gauge ICs that incorporate the Maxim ModelGauge m5 EZ algorithm. The ModelGauge m5 EZ algorithm makes fuel gauge implementation easy by eliminating battery characterization requirements and simplifying host software interaction. The MAX17260 features a high-side current-sensing option in addition to low-side sensing for single-cell applications. The MAX17261 supports multiple-series battery packs with low quiescent current. The MAX17262 has internal current sensing for single-cell applications. The MAX17263 features versatile single/multiple-series cell support with an integrated LED driver and pushbutton.

The MAX1726x measures voltage, current, and temperature to produce fuel gauge results. The MAX1726x uses either an external thermistor or internal die temperature to measure temperature of the battery pack.

The ModelGauge™ m5 EZ robust algorithm provides tolerance against battery diversity. This 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 MAX1726x automatically compensates for aging, temperature, and discharge rate; it provides accurate state of charge (SOC) measurements in percentages (%) and remaining capacity measurements in milliampere-hours (mAh) over a wide range of operating conditions. The MAX1726x ensures that the fuel gauge error always converges to 0% as the cell approaches empty. It also provides accurate estimates of time to empty (TTE) and time to full (TTF) as well as three methods for reporting the age of the battery: reduction in capacity, increase in battery resistance, and cycle odometer.

The MAX1726x provides the following additional features detailed in this user guide:

- Support for special chemistries, such as LiFePO4
- Dynamic Power technology guides in throttling the processor (or other load) optimally to maintain the battery above a minimum voltage while maximizing performance
- TTE estimation calculated either with constant power or constant current



## ModelGauge m5 Algorithm

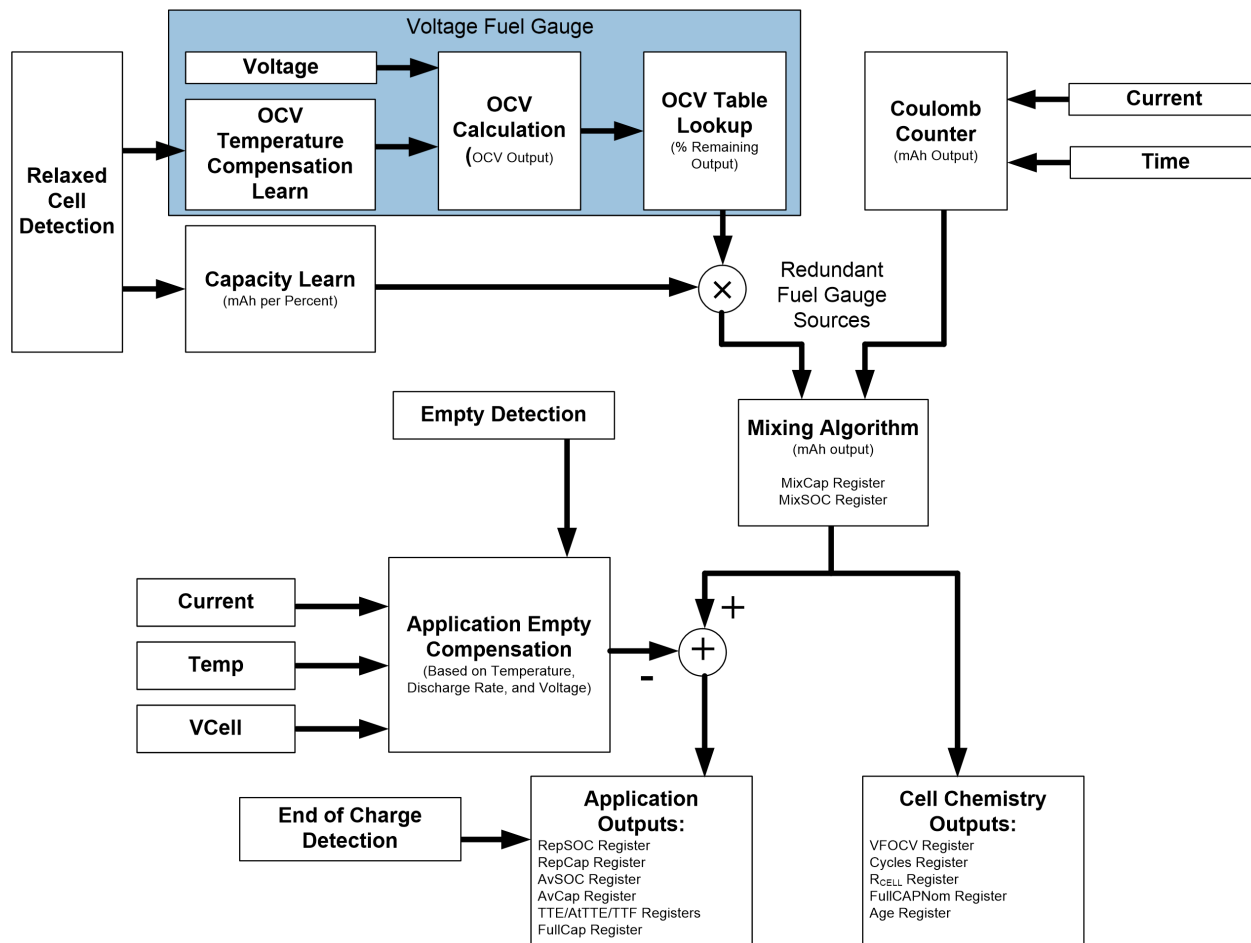


Figure 1. ModelGauge m5 flow diagram.

### OCV Estimation and Coulomb Count Mixing

The core of the ModelGauge m5 algorithm is a mixing algorithm that continuously combines the voltage fuel gauge estimation with the coulomb counter. Unlike traditional coulomb counters, which lose state information after reset, ModelGauge m5 initially uses the voltage fuel gauge. As the cell progresses through cycles in the application, coulomb-counter accuracy improves, and the mixing algorithm alters the weighting so that the coulomb-counter result is dominant. See Figure 2.

The resulting output from the mixing algorithm does not suffer accumulation drift from the current measurement offset error and is more stable than a stand-alone OCV estimation algorithm. Figure 3 shows the gradual reduction of voltage fuel-gauge influence over battery cycles. The initial accuracy depends on the relaxation state of the cell. The highest initial accuracy is achieved with a fully relaxed cell.

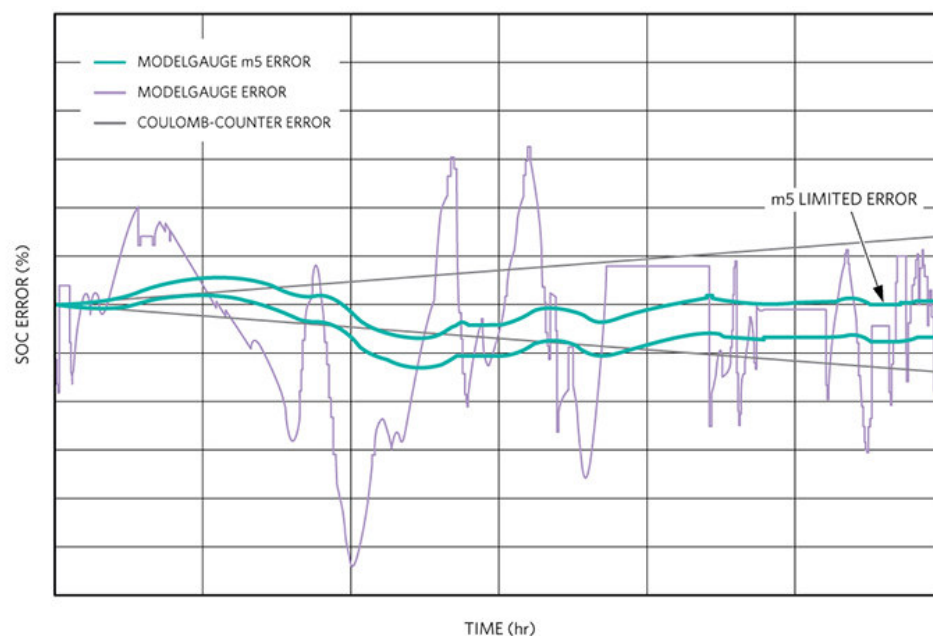


Figure 2. Error is filtered and bounded by a combination of ModelGauge and coulomb counter.

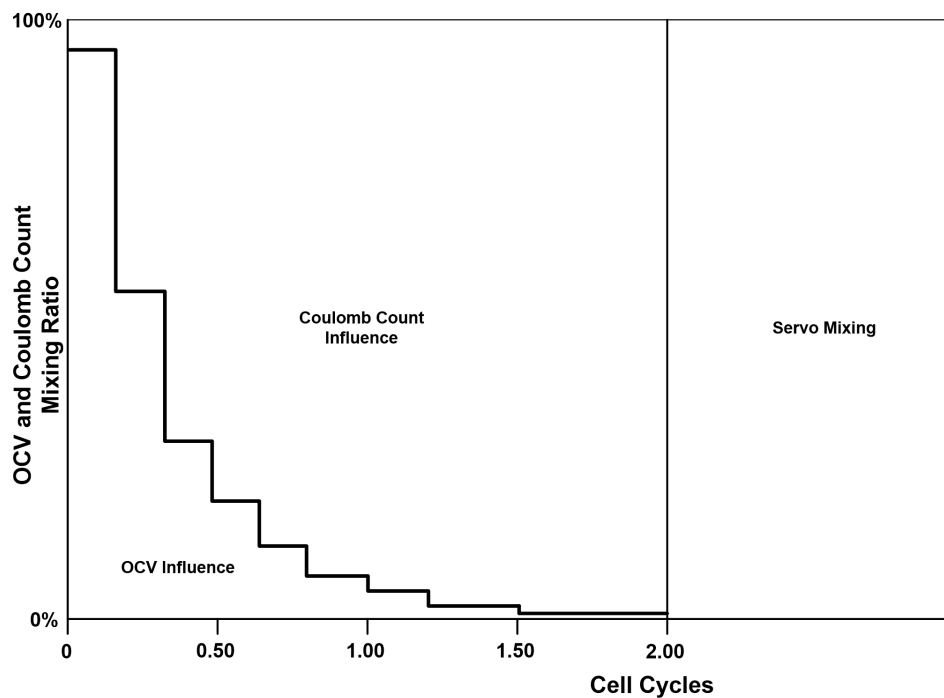


Figure 3. Voltage fuel gauge vs. coulomb-counter influence vs. first cycles.

## Fuel Gauge Learning

The MAX1726x periodically makes internal adjustments to cell characterization and application information in order to remove initial errors and maintain accuracy as the cell ages. These

adjustments always occur as small under-corrections to prevent instability of the learning process as well as noticeable jumps in the fuel gauge outputs. Learning occurs automatically without any input from the host. In addition to estimating the SOC, the IC observes the battery's relaxation response and adjusts the dynamics of the voltage fuel gauge. Registers used by the algorithm include:

- **Application Capacity (FullCapRep Register).** This is the total capacity available to the application at full; it is set through the IChgTerm and FullSOCThr registers as described in the *End-of-Charge Detection* section. See the FullCapRep register description.
- **Cell Capacity (FullCapNom Register).** This is the total cell capacity at full, including some capacity that sometimes is not available to the application due to high loads and/or low temperature. The IC periodically compares percent change based on an open circuit voltage measurement vs. coulomb-count change as the cell charges and discharges, maintaining an accurate estimation of the pack capacity in mAh as the pack ages. See Figure 4.
- **Voltage Fuel-Gauge Adaptation.** The IC observes the battery's relaxation response and adjusts the dynamics of the voltage fuel gauge. This adaptation adjusts the RComp0 register during qualified cell relaxation events. Learning can occur on relaxation events with the charge/discharge cycle. Learning does not require charge-to-full or discharge-to-empty.
- **Empty Compensation Adaptation.** The IC updates internal data whenever cell empty is detected ( $V_{Cell} < V_{Empty}$ ) to account for cell age or other cell deviations from the characterization information.

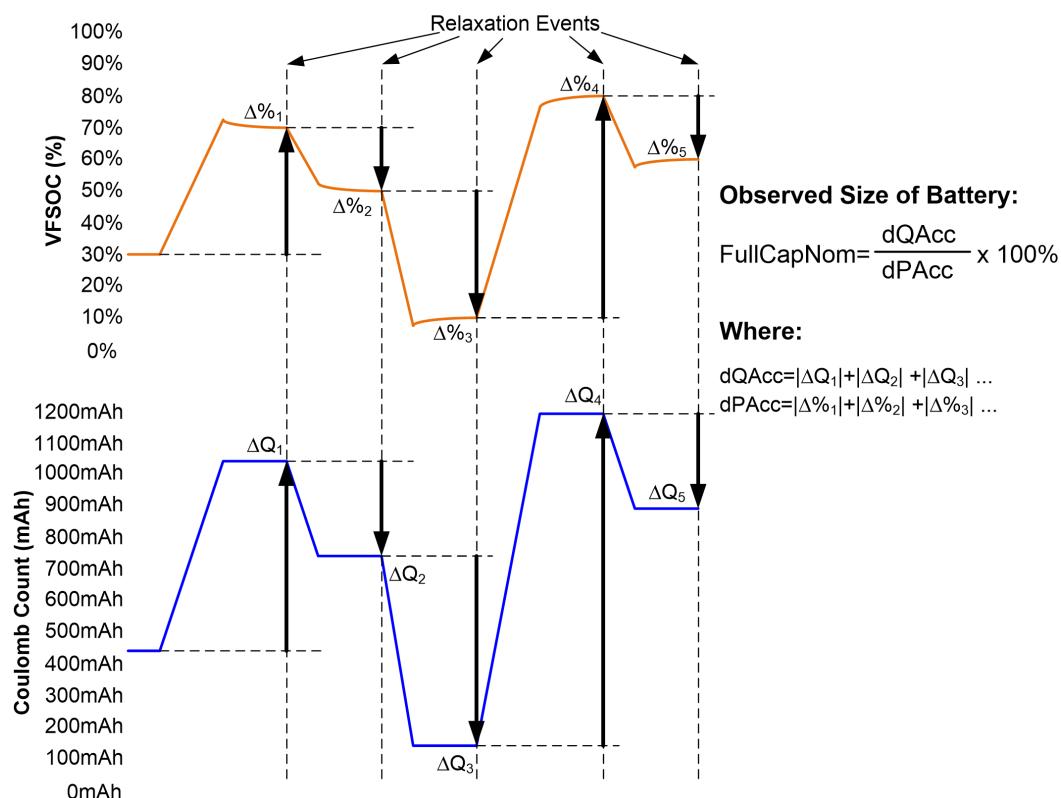


Figure 4. ModelGauge m5 learns full capacity during arbitrary cycling.

## Empty Compensation

As the temperature and discharge rate of an application changes, the amount of charge available to the application also changes. The ModelGauge m5 algorithm distinguishes between the remaining capacity of the cell and the remaining capacity of the application, and it reports both results to the user.

The MixCap output register tracks the charge state of the cell. This is the theoretical mAh of charge that can be removed from the cell under ideal conditions—extremely low discharge current and no concern for cell voltage. This result is not affected by application conditions such as cell impedance or minimum operating voltage of the application. ModelGauge m5 continually tracks the expected empty point of the application in mAh. This is the amount of charge that cannot be removed from the cell by the application because of minimum voltage requirements and internal voltage drops. The IC subtracts the amount of charge not available to the application from the MixCap register and reports the result in the AvCap register.

Because available remaining capacity is highly dependent on the discharge rate, the AvCap register can be subject to large instantaneous changes as the application load current changes. The resulting remaining capacity or percentage can increase, even while discharging, if the load current suddenly drops or temperature increases. This result, although correct, can be very counterintuitive to the host software or end-user. The RepCap output register contains a filtered version of AvCap that removes any abrupt changes in remaining capacity. RepCap converges with AvCap over time to correctly predict either the application empty point while discharging or the application full point while charging. Figure 5 shows the relationship of these registers.

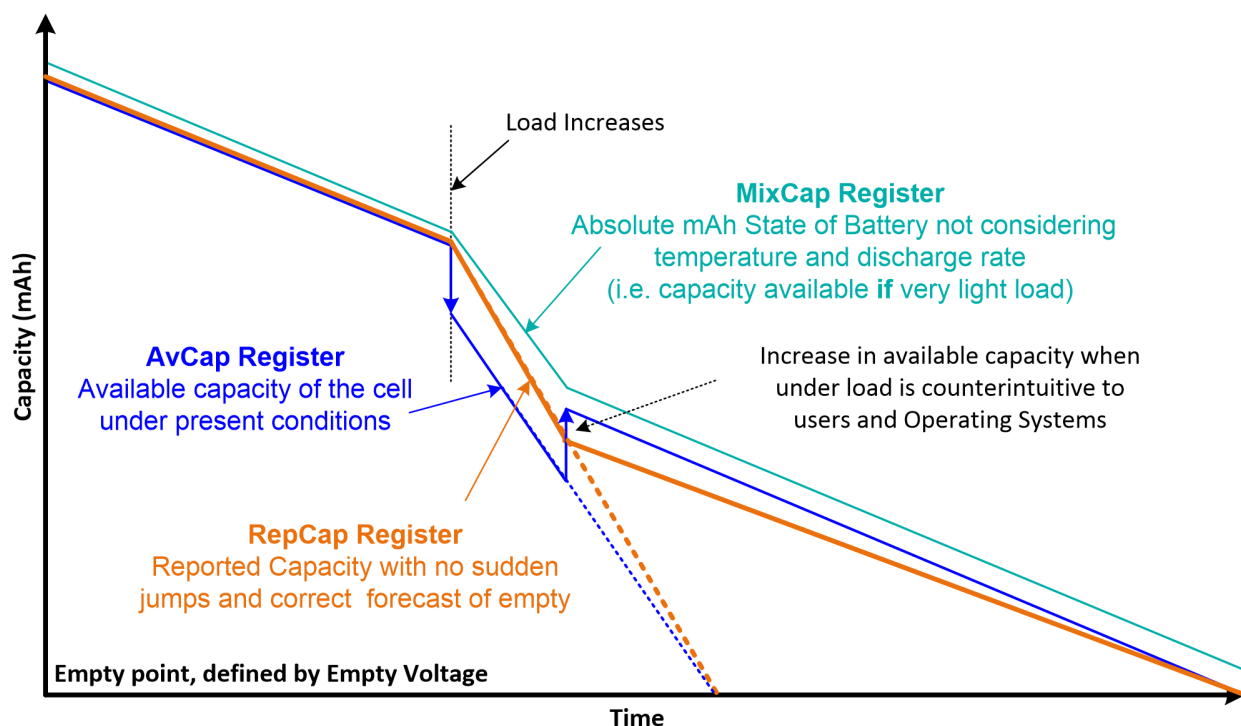


Figure 5. ModelGauge m5 learns full capacity during arbitrary cycling.

## Converge-to-Empty

The MAX1726x includes a feature that guarantees the fuel gauge output smoothly converges to 0% as the cell voltage approaches the empty voltage. As the cell voltage approaches the target

empty voltage (AvgVCell approaches VEmpty) the IC smoothly adjusts the rate of change of RepSOC so that the fuel gauge reports 0% at the same time that the cell voltage reaches empty, as shown in Figure 6. This prevents early or late empty reporting by the fuel gauge, maximizing application runtime.

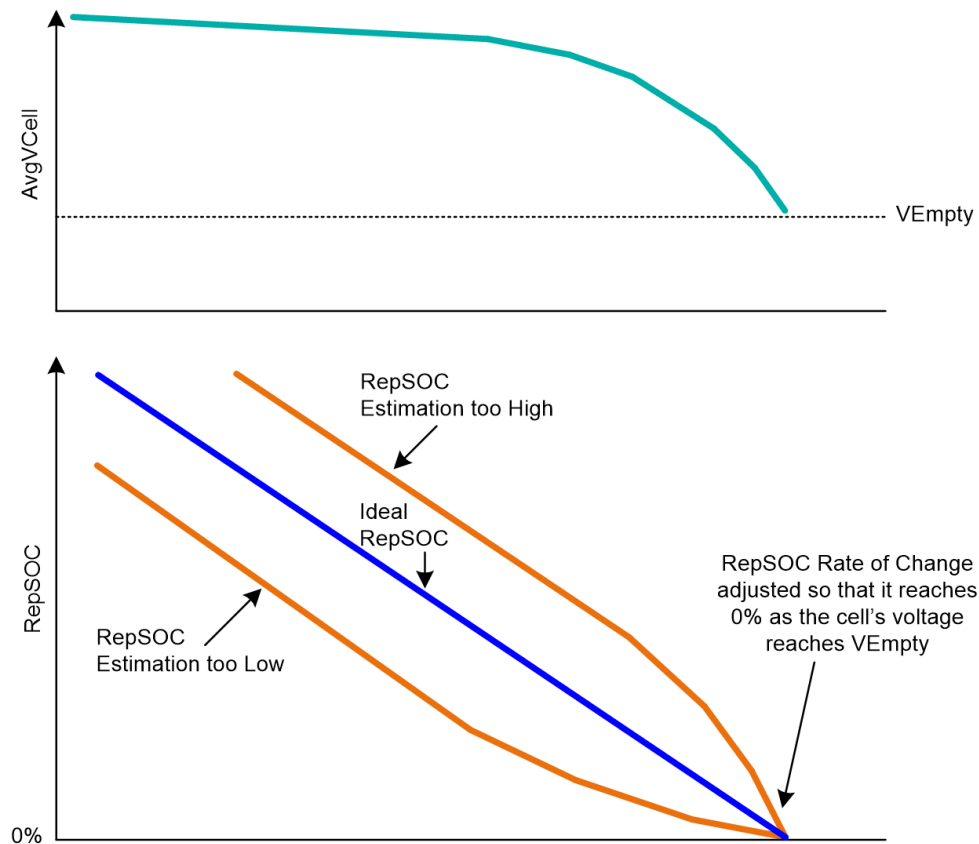


Figure 6. MAX1726x converge-to-empty performance.

### Empty Hold and 99% Hold

The MAX1726x supports two modes that limit the RepSOC% reported until a specific condition is reached.

- Empty Hold.** This feature limits RepSOC to not fall below x% (1% default) until empty voltage is crossed. This can be useful with operating systems that force system shutdown at a particular battery percentage. A Windows computer, for example, may force a system shutdown or hibernate when the fuel gauge crosses 5%. So, setting Empty Hold to 6% can guarantee deeper discharge to a specified voltage level (EmptyVltHold), and thereby often extend runtime.
- 99% Hold.** This feature limits RepSOC to not exceed 99% until a charge termination event is detected. See the *End-of-Charge Detection* section for more details.

See the SOCHold register description for more information.

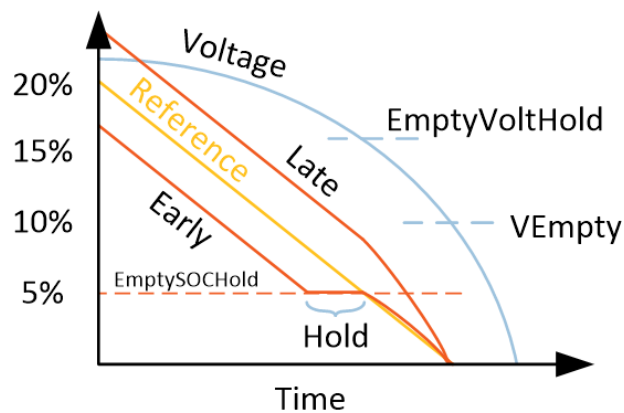


Figure 7. SOC hold conceptual drawing.

### SOCHold Register (D3h)

Register Type: Special

The SOCHold register configures operation of the hold-before-empty feature and also the enable bit for 99% hold during charge. The default value for SOCHold is 0x1002. Table 1 shows the SOCHold register format.

**Table 1. SOCHold (D3h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	99%HoldEn	EmptyVoltHold							EmptySOCHold				

**EmptyVoltHold:** The positive voltage offset that is added to VEmpty. At VCell = VEmpty + EmptyVoltHold point, the empty detection/learning is occurred. EmptyVoltHold has an LSb of 10mV, giving a range of 0 to 1270mV.

**EmptySOCHold:** This is the RepSOC at which RepSOC is held constant until the EmptyVoltHold condition is crossed. After empty detection occurs, RepSOC update continues as expected. EmptySOCHold has an LSb of 0.5%, with a full range of 0 to 15.5%.

**99%HoldEn:** Enable bit for 99% hold feature during charging. When enabled, RepSOC holds a maximum value of 99% until Full Qualified is reached.

### End-of-Charge Detection

The IC detects the end of charge when the application current falls into the band set by the IChgTerm register and the VFSOC register is greater than the FullSOCThr register. By monitoring both the Current and AvgCurrent registers, the device can reject false end-of-charge events, such as application load spikes or early charge-source removal. See Figure 8. When a proper end-of-charge event is detected, the device learns the FullCapRep register based on the RepCap register. If the old FullCapRep value is too high, it is adjusted on a controlled downward slope near the end of charge, as defined by the MiscCfg.FUS setting, until it reaches RepCap. If the old FullCapRep is too low, it is adjusted upward to match RepCap. This prevents the calculated SOC from reporting greater than 100%. See Figure 9.

Charge Termination is detected by the IC when the following conditions are met:

- VFSOC Register > FullSOCThr Register
- AND  $I_{ChgTerm} \times 0.125 < \text{Current Register} < I_{ChgTerm} \times 1.25$
- AND  $I_{ChgTerm} \times 0.125 < \text{AvgCurrent Register} < I_{ChgTerm} \times 1.25$

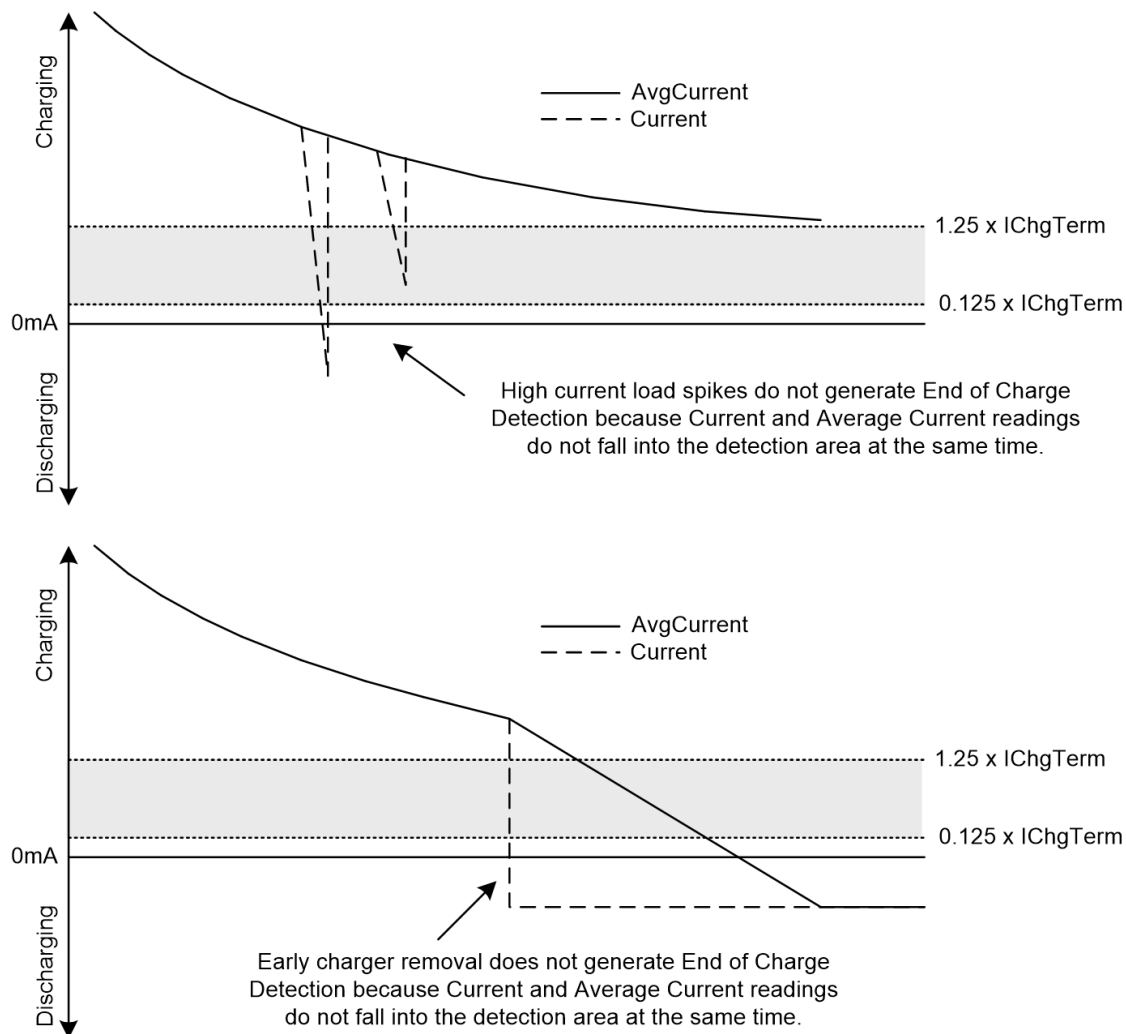


Figure 8. ModelGauge m5 rejects false end-of-charge events.

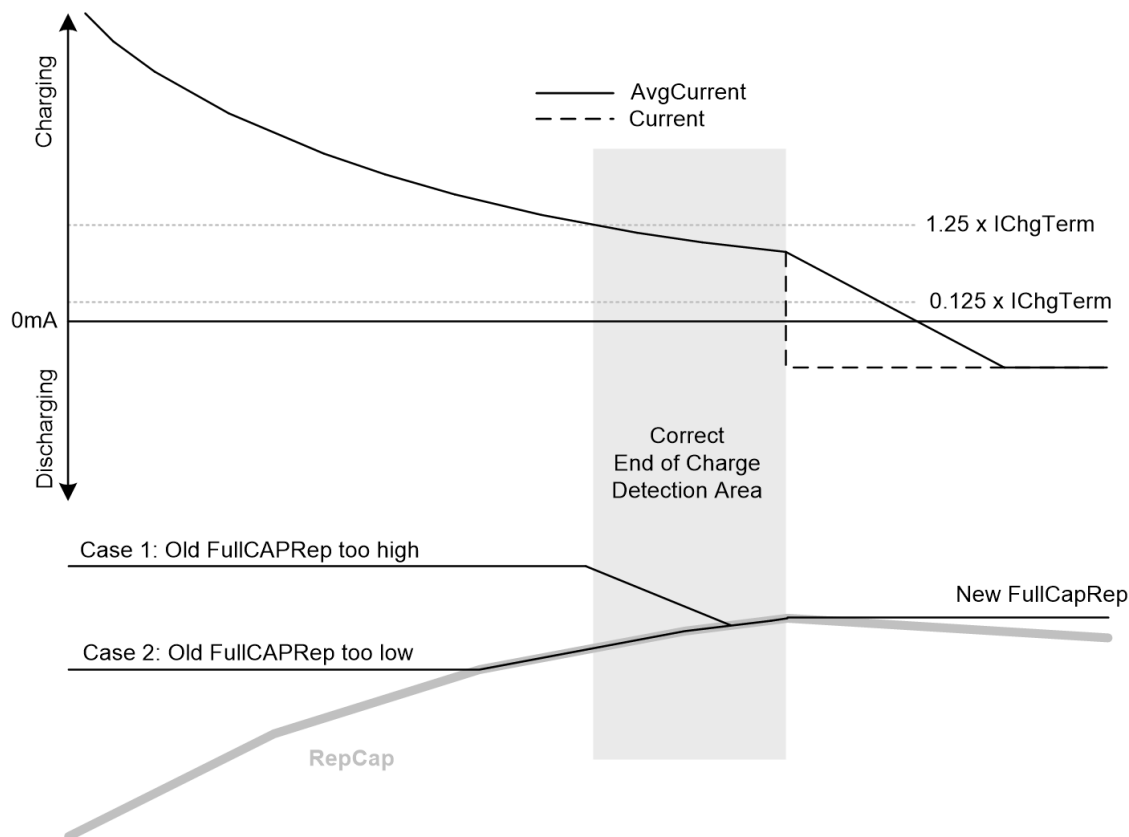


Figure 9. FullCapRep learn at end of charge.

## Predicting Runtime of a Hypothetical Load

The MAX1726x provides an AtRate register that can be used to predict the remaining percentage or time associated with different load conditions. This can be used for system power-management decisions to prevent or limit the load according to the battery capability or runtime requirements.

The AtRate function allows host software to see the theoretical remaining time or capacity for any hypothetical load current. AtRate can be used for power management by limiting system loads depending on present conditions of the battery. To use AtRate, set the AtRateEn bit in the Config2 register. To query the remaining capacity or time associated with a hypothetical load, write the load to AtRate (negative value) and read the result from AtTTE, AtAvSOC, or AtAvCap.

Host software should wait 351ms in active mode or 11.25s in hibernate mode after writing the AtRate register before reading any of the result registers. Exit hibernate with the "Soft-Wakeup" command to provide faster answers (see Command Register (60h)).

## Lithium Iron Phosphate Support

The MAX1726x supports LiFePO<sub>4</sub> batteries with a special model configuration. To produce good SOC (%) accuracy performance, it is necessary to characterize and model the specific LiFePO<sub>4</sub> cells being used. The MAX1726x provides additional algorithm support specifically for the challenges associated with LiFePO<sub>4</sub> and other "flat" OCV chemistries.



The OCV/SOC curve of LiFePO4 is much flatter than conventional lithium cobalt chemistries, which produces a greater sensitivity to the algorithm's interpretation of cell voltage and open-circuit voltage. For the fuel gauge algorithm to achieve accurate full capacity measurement over time, the battery's full capacity must be calculated outside the keep-out window, which has the flattest region in the OCV/SOC curve. The default keep-out window is 3.275V to 3.350V.

To configure the MAX1726x for LiFePO4 support:

1. Characterize the battery. The characterization data should be translated by Maxim into a battery model.
2. Write 0x0060 to ModelCFG register to enable LiFePO4 mode.
3. Load the rest of the battery model (refer to *Option 3: Long Format* in the [MAX1726x Software Implementation Guide](#)).

### ScOcvLim Register (D1h)

Register Type: Special

Initial Value: 0x479E

This register only has usage when ModelCfg.ModelID is selected as 6 (LiFePO4). Table 2 shows the register format.

**Table 2. ScOcvLim (D1h) Format**

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

**OCV\_Low\_Lim:** Defines the lower limit for the OCV keep-out region. A 5mV resolution gives a 2.56V to 5.12V range. The lower limit voltage of the OCV keep-out region is calculated as  $2.56V + (OCV\_Low\_Lim \times 5mV)$ . The default value is 0x8F.

**OCV\_Delta:** Defines the delta between lower and upper limits for the OCV keep-out region. A 2.5mV resolution gives a 0 to 320mV range. The upper limit voltage of the OCV keep-out region is calculated as  $2.56V + (OCV\_Low\_Lim \times 5mV) + (OCV\_Delta \times 2.5mV)$ . The default value is 0x1E.

The default OCV\_low is 3275mV and OCV\_high is 3350mV.

## ModelGauge m5 Standard Register Formats

Unless otherwise stated during a given register's description, all IC registers follow the same format depending on the type of register. See Tables 3 and 4 for the resolution and range of any register described hereafter. Note that current and capacity values are displayed as a voltage and must be divided by the sense resistor to determine amps or amp-hours.

**Table 3. MAX17260/MAX17261/MAX17263 ModelGauge m5 Register Standard Resolutions**

REGISTER TYPE	LSB SIZE	MINIMUM VALUE	MAXIMUM VALUE	NOTES
Capacity	5.0μVh/ R <sub>SENSE</sub>	0.0μVh	327.675mVh/ R <sub>SENSE</sub>	Equivalent to 0.5mA with a 0.010Ω sense resistor.
Percentage	1/256%	0.00%	256.00%	1% LSb when reading only the upper byte.
Voltage	1.25mV/16	0.0V	5.11992V	For MAX17261 voltage is on per-cell basis.
Current	1.5625μV/ R <sub>SENSE</sub>	-51.2mV/ R <sub>SENSE</sub>	51.1984mV/ R <sub>SENSE</sub>	Signed 2's complement format. Equivalent to 156.25μA with a 0.010Ω sense resistor.
Temperature	1/256°C	-128.0°C	+127.996°C	Signed 2'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.

**Table 4. MAX17262 ModelGauge m5 Register Standard Resolutions**

REGISTER TYPE	LSB SIZE	MINIMUM VALUE	MAXIMUM VALUE	NOTES
Capacity	0.5mAh	0.0mAh	32767.5mAh	—
Percentage	1/256%	0.0%	255.9961%	1% LSb when reading only the upper byte.
Voltage	1.25mV/16	0.0V	5.11992V	—
Current	156.25μA	-5.12A	5.12A	Signed 2's complement format.
Temperature	1/256°C	-128.0°C	+127.996°C	Signed 2'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.

## Analog Measurements

To properly fuel gauge a battery, the MAX1726x continually monitors the battery voltage, battery temperature, and current flow into and out of the battery. The following sections detail how these measurements occur.

### Current Measurements

Current flow through the battery is determined by making voltage measurements across the sensing element. The resulting value is reported as a signed value in  $\mu\text{V}$  or  $\mu\text{Vh}$  and must be divided by the sense resistor value in ohms to convert to current.

For the MAX17260/MAX17261/MAX17263, the value of the external sense resistor determines the range and the resolution of current values that can be reported. They have a maximum measurement range of  $\pm 51.2\text{mV}$  and a reporting resolution of  $1.5625\mu\text{V}$ . Table 5 shows the measurement ranges and resolutions of several common-sense resistor values. For the MAX17262, the current measurement is done with an internal sensing element. The measurement range is fixed at  $\pm 5.12\text{A}$ , and the measurement resolution is fixed at  $156.25\mu\text{A}$ .

**Table 5. Measurement Range and Resolution vs. Sense Resistor Value for the MAX17260/MAX17261/MAX17263**

Sense Resistor ( $\text{m}\Omega$ )	Measurement Range (A)	Measurement Resolution ( $\mu\text{A}$ )
20.0	$\pm 2.56$	78.125
10.0	$\pm 5.12$	156.25
5.0	$\pm 10.24$	312.50
3.5 (board trace)	$\pm 14.69$	446.43
2.0	$\pm 25.60$	781.25

### ***CGain Register (2Eh) and COff Register (2Fh) Register***

Register Type: Special

Initial Value: CGain = 0x0400 and COff = 0x0000

The CGain and COff registers adjust the gain and offset of the current measurement result. The current measurement ADC is factory-trimmed to data-sheet accuracy and does not require the user to make further adjustments. The default power-up settings for CGain and COff apply no adjustments to the Current register reading. For specific application requirements, the CGain and COff registers can be used to adjust readings as follows:

Current Register = Current ADC Reading  $\times$  (CGain Register / 0400h) + COff Register

### **Copper Trace Current Sensing**

The MAX17260/MAX17261/MAX17263 can measure current using a copper board trace instead of a traditional sense resistor, the main difference being the ability to adjust to the change in sense resistance over temperature. The MAX17260/MAX17261/MAX17263 evaluation kits (EV kits) include PC board (PCB) traces that demonstrate this functionality.

Board-to-board variations make it challenging to use PCB current sensing to achieve the normally stringent requirements for fuel gauging. However, the MAX17260/MAX17261/MAX17263 can meet this challenge due to the robust ModelGauge m5 EZ algorithm, plus PCB compensations provided.

To configure the MAX17260/MAX17261/MAX17263 for accurate PCB current sensing:

1. Set CGTempCo to 0x20C8, corresponding to copper at 0.4% per °C. Set CGTempCo to 0 to disable copper current sensing when using a normal sense resistor.
2. Set CGain according to the room-temperature resistance associated with the PCB trace.
3. Configure the Curve register to compensate for the self-heating of the trace. The PCB trace temperature mismatches the MAX1726x die temperature (used for compensation), especially at high currents. This additional self-heating is not directly sensed by the IC. Fine-tune the Metal Trace Curve byte of the Curve register value so that at high current the Current register achieves same accuracy as low current.

For 1-ounce copper, a length-to-width ratio of 6:1 creates a 0.0035Ω sense resistor, which is suitable for most applications. The PCB manufacturing process might produce a trace-resistance variation of ±20%. ModelGauge m5 adapts to this variation and reports SOC accurately. The adaptation process is similar to the adaptation for battery-to-battery full-capacity variation.

### **Curve Register (B9h)**

Register Type: Special

The upper half of the Curve register applies curvature correction current measurements made by the IC when using a copper trace as the sense resistor. The lower half of the register is the parameter for temperature measurement using an external NTC resistor. See the *Temperature Measurements* section for detailed information about the Curve.TCurve value.

**Table 6. Curve (B9h) Format**

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

### **CGTempCo Register (B8h)**

Register Type: Special

If CGTempCo is nonzero, then CGTempCo is used to adjust current measurements for temperature. CGTempCo has a range of 0% to 3.1224% per °C with a step size of 3.1224 / 0x10000 percent per °C. If a copper trace is used to measure battery current, CGTempCo should be written to 0x20C8 or 0.4% per °C, which is the approximate temperature coefficient of a copper trace.

### **Temperature Measurements**

The MAX1726x can measure and report its own internal temperature or report an external temperature by using an NTC thermistor divider network connected to the THRM and AIN pins. Only one measurement path can be active at any time.

When measuring temperature externally, the MAX1726x takes a raw percentage reading and converts that value to temperature using gain (TGain register), offset (TOff register), and second-order curve adjustment (Curve register). This allows an accurate temperature to be reported over a variety of different NTC thermistor options. Internal temperature measurement does not use TGain, TOff, or Curve.

### ***AIN Register (27h)***

Register Type: Special

The external temperature measurement on the TH pin is compared to the BATT pin voltage. The MAX1726x stores the result as a ratiometric value from 0% to 100% in the AIN register with an LSB of 0.0122%. The TGain, TOff, and Curve register values are then applied to this ratiometric reading to convert the result to temperature.

### ***TGain (2Ch) Register and TOff (2Dh) Register***

Register Type: Special

Initial Value: TGain = 0xEE56 and TOff = 0x1DA4

The TGain, TOff, and Curve registers are used to calculate temperature from the measurement of the TH pin with an accuracy of  $\pm 3^{\circ}\text{C}$  over a range of  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ . Table 7 lists the recommended TGain, TOff, and Curve register values for common NTC thermistors.

**Table 7. Register Settings for Common Thermistor Types**

THERMISTOR	R25C (k $\Omega$ )	BETA	RECOMMENDED TGain	RECOMMENDED TOff	RECOMMENDED Curve.TCurve
SEMITEC 103AT-2, Murata NCP15XH103F03RC	10	3435	0xEE56	0x1DA4	0x0025
Fenwal® 197-103LAG-A01	10	3974	0xF49A	0x16A1	0x0064
TDK® Type F	10	4550	0xF284	0x18E8	0x0035

### **Voltage Measurements**

For the MAX17260/MAX17262, measurement of the battery voltage is performed by determining the voltage difference between the BATT and GND pins. For the MAX17261, external regulated supply is applied on the BATT pin and 40% of the voltage per cell is measured at the CELLx pin referenced to the GND pin.

For the MAX17263, the voltage at the CELLx pin is measured at startup to determine if it is a single-cell or multi-cell application. In single-cell applications, CELLx is tied to BATT, and the voltage measurement is done on the BATT pin. In multiple-series applications, the external regulated supply is applied on the BATT pin, and 40% of the voltage per cell is measured at the CELLx pin referenced to the GND pin.

## Dynamic Power

To achieve better runtime and CPU performance, the MAX1726x supports the Intel® DBPT standard, which provides the on-demand battery capability used for managing pulse loads such as CPU loads. A CPU requires the battery to deliver short pulses of high power. To support these high pulses without system undervoltage, the MAX1726x indicates the peak power levels that can be taken from the battery. The host can use this information to set its maximum current in accordance with battery power capability.

In many 1-Series applications, the system requires at least 3.3V to operate correctly. By configuring the MAX1726x for Dynamic Power, the system's loads can be controlled or limited to stay within the battery's capability and ensure that a minimum system voltage (MinSysVolt) is not crossed until the battery is a very low state.

The implementation of Intel Dynamic Battery Power Technology v2.0 relies on new functions and corresponding registers. This document defines those new functions. The implementation in the MAX1726x includes all of the same registers as the Intel specification. The register set uses different LSBs and addresses compared to the register set in the data sheet to comply with the standard.

## Dynamic Power Performance

Figure 10 shows the performance of Dynamic Power. In this test, the load was limited according to the feedback from the MaxPeakCurrent register. The MAX1726x continues to update the MaxPeakCurrent output as the SOC, load, or temperature changes. The performance of the MaxPeakCurrent register ensures that the voltage is limited to not fall below the MinSysVolt setting.

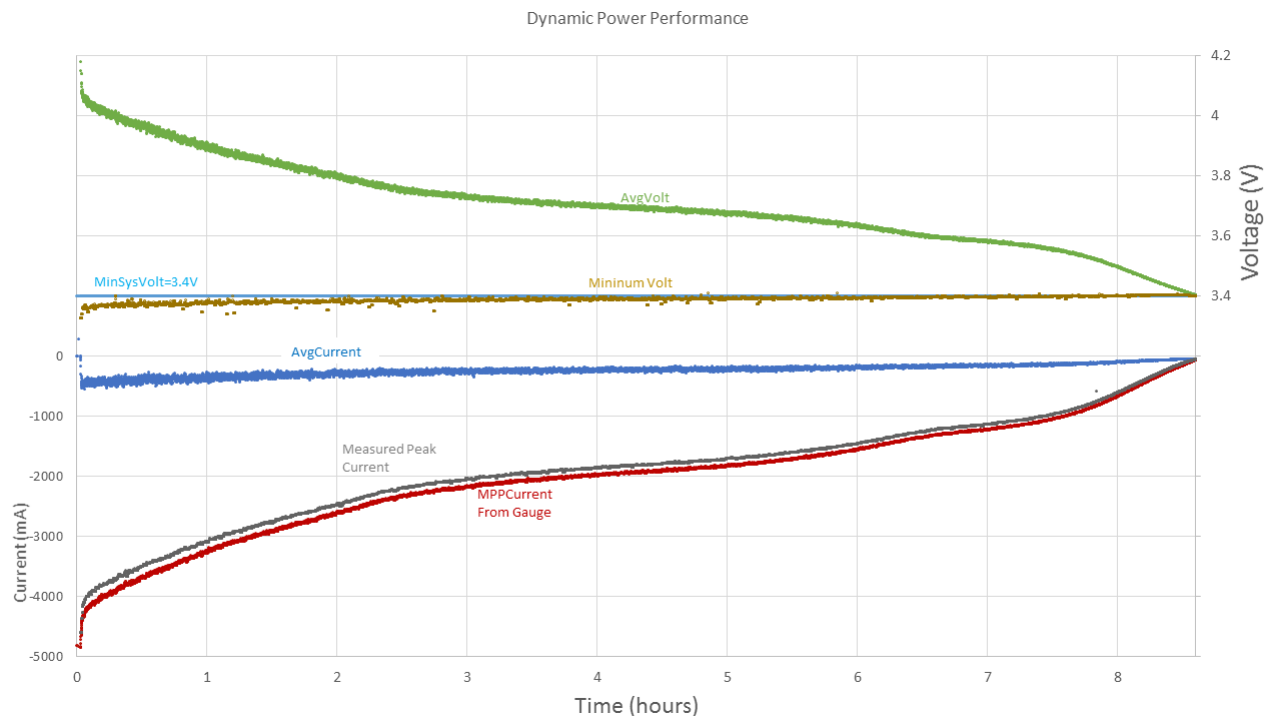


Figure 10. Dynamic Power limits voltage spikes when the load is limited by MaxPeakCurrent.

## Dynamic Power Output Registers

### ***MPPCurrent Register (D9h)***

The MAX1726x estimates the maximum instantaneous peak current of the battery pack in mA, which the battery can support for up to 10ms, given the external resistance and required minimum voltage of the voltage regulator. The MPPCurrent value is negative and updates every 175ms.

### ***SPPCurrent Register (DAh)***

The MAX1726x estimates the sustained peak current of the battery pack in mA, which the battery can support for up to 10s, given the external resistance and required minimum voltage of the voltage regulator. The SPPCurrent value is negative and updates every 175ms.

### ***MaxPeakPower Register (D4h)***

The MAX1726x estimates the maximum instantaneous peak output power of the battery pack in mW, which the battery can support for up to 10ms, given the external resistance and required minimum voltage of the voltage regulator. The MaxPeakPower value is negative (discharge) and updates every 175ms. The LSB is 0.8mW. The calculation for this register value is:

$$\text{MaxPeakPower} = \text{MPPCurrent} \times \text{AvgVCell}$$

### ***SusPeakPower Register (D5h)***

The fuel gauge estimates the sustainable peak output power of the battery pack in mW, which the battery supports for up to 10s, given the external resistance and required minimum voltage of the voltage regulator. The SusPeakPower value is negative and updated each 175ms. The LSB is 0.8mW. The calculation for this register value is:

$$\text{SusPeakPower} = \text{SPPCurrent} \times \text{AvgVCell}$$

## Dynamic Power Configuration Registers

The following registers provide the battery capability estimates from the Dynamic Power calculations.

### ***PackResistance Register (D6h)***

When the MAX1726x is installed host-side, simply set PackResistance to zero, since the MAX1726x can observe the total resistance between it and the battery.

When the MAX1726x is installed pack-side, configure PackResistance according to the total non-cell pack resistance. This should account for all resistances due to cell interconnect, sense resistor, FET, fuse, connector, and other resistance between the cells and output of the battery pack. The cell internal resistance should not be included and is estimated by the MAX1726x.

The LSB of the register is 0.244140625mΩ. (The value of 0x1000 represents 1000mΩ.)

### ***SysResistance Register (D7h)***

Set SysResistance according to the total system resistance. This should include any connector and PCB trace between the MAX1726x and the system at risk for dropout when the voltage falls below MinSysVolt.

SysResistance is initialized to a default value upon removal or insertion of a battery pack. Writes with this function overwrite the default value.

The LSB of the register is 0.244140625mΩ. (The value of 0x1000 represents 1000mΩ.)

### **MinSysVoltage Register (D8h)**

Set MinSysVoltage according to the minimum operating voltage of the system. This is generally associated with a regulator dropout or other system failure/shutdown. The system should still operate normally until this voltage.

MinSysVoltage is initialized to the default value (3.0V). Writing with this function overwrites the default value.

### **RGain Register (43h)**

Register Type: Special

Initial Value: 0x8080

The RGain register sets the value of RGain1 and RGain2 during DBPT register calculation. Table 8 shows the register format.

**Table 8. RGain (43h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RGain1								RGain2				SusToMaxRatio			

RGain provides additional limitations to ensure that the Dynamic Power Outputs (MPPCurrent, MaxPeakPower, SPPCurrent, SusPeakPower) are conservative, thereby preventing any risk of system shutdown.

**RGain1:** Gain resistance used for peak current and power calculation.  $RGain1 = 80\% + 0.15625\% \times RG1$ . The range of RGain1 is between 80~120%.

**RGain2:** Gain resistance used for peak current and power calculation.  $RGain2 = 60\% + 5\% \times RG2$ . The range of RGain2 is between 60~140%.

**SusToMaxRatio:** Used to calculate the maximum ratio between SPPCurrent to MPPCurrent. The maximum value of SPPCurrent =  $MPPCurrent \times (0.75 - SusToPeakRatio \times 0.04)$ .



## Serial Number Feature

Each MAX1726x provides a unique serial number ID. To read this serial number, clear AtRateEn and DPEn in Config2. After the read operation is completed, the IC sets the Status2.SNReady flag to indicate that the serial number read operation is completed.

The 128-bit serial information read from the MAX1726x overwrites 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 9. Accessing the Serial Number**

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

## Determining Fuel Gauge Accuracy

To determine the true accuracy of a fuel gauge, as experienced by end users, the battery should be exercised in a dynamic manner. End-user accuracy cannot be completely understood with only simple cycles. To challenge a correction-based fuel gauge, such as a coulomb counter, test the battery with partial loading sessions. For example, a typical user might operate the device for 10 minutes and then stop use for an hour or more. A robust test method includes these kinds of sessions many times at various loads, temperatures, and durations. Refer to [Application Note 4799: Cell Characterization Procedure for a ModelGauge m3 Fuel Gauge](#).

### Initial Accuracy

The IC uses the first voltage reading after power-up or cell insertion to determine the starting output of the fuel gauge. It is assumed that the cell is fully relaxed prior to this reading; however, this is not always the case. If there is a load or charge current at this time, the initial reading is compensated using a default battery resistance of 40m $\Omega$  to estimate the relaxed cell voltage. If the cell was recently charged or discharged, the voltage measured by the IC might not represent the true SOC of the cell, resulting in initial error in the fuel gauge outputs. In most cases, this error is minor and is quickly removed by the fuel gauge algorithm during the first hour of normal operation.

## ModelGauge m5 Registers

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. Table 10 shows the ModelGauge m5 memory map.

**Table 10. ModelGauge m5 Register Memory Map**

PAGE/ WORD	00h	10h	20h	30h	40h	B0h	D0h
0h	Status	FullCapRep	TTF	Reserved	Reserved	Status2	RSense / UserMem3
1h	VAIrtTh	TTE	DevName	Reserved	Reserved	Power	ScOcvLim
2h	TAIrtTh	QRTale00	QRTale10	QRTale20	QRTale30	ID / UserMem2	VGain
3h	SAIrtTh	FullSocThr	FullCapNom	Reserved	RGain	AvgPower	SOCHold
4h	AtRate	RCell	Reserved	DieTemp	Reserved	IAIrtTh	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

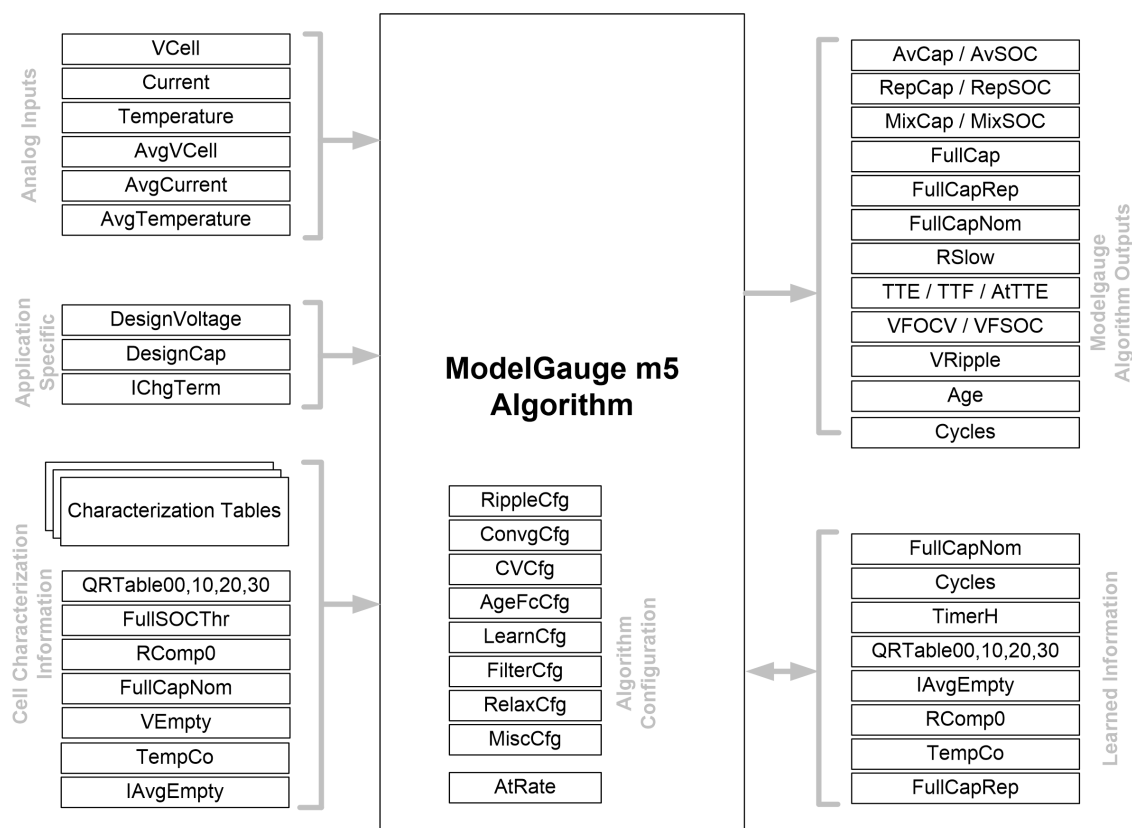


Figure 11. ModelGauge m5 algorithm registers.

## ModelGauge m5 Algorithm Battery Parameters

The following registers are inputs to the ModelGauge m5 algorithm and store characterization information for the application cells as well as important application-specific specifications. They are described only briefly here. Contact Maxim for information regarding cell characterization.

### VEmpty Register (3Ah)

Initial Value: 0xA561 (3.3V / 3.88V)

The VEmpty register sets thresholds related to empty detection during operation. Table 11 shows the register format.

Table 11. VEmpty (3Ah) Format

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

**VE:** Empty Voltage Target, during load. The fuel gauge provides capacity and percentage relative to the empty voltage target, eventually declaring 0% at VE. A 10mV resolution gives a range of 0 to 5.11V. This value is written to 3.3V after reset.

**VR:** Recovery Voltage. Sets the voltage level for clearing empty detection. Once the cell voltage rises above this point, empty voltage detection is re-enabled. A 40mV resolution gives a range or 0 to 5.08V. This value is written to 3.88V, which is recommended for most applications.

### **DesignCap Register (18h)**

Register Type: Capacity

The DesignCap register holds the expected capacity of the cell. This value is used to determine the age and health of the cell by comparing against the measured present cell capacity.

### **ModelCfg Register (DBh)**

Register Type: Special

Initial value: 0x8400

The ModelCFG register controls basic options of the EZ algorithm. Table 12 shows the register format.

**Table 12. ModelCFG (DBh) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Refresh	0	R100	0	0	VChg	0	0	ModelID				Reserved	Reserved	0	0

**Refresh:** Set Refresh to 1 to command the model reload. After execution, the MAX1726x clears Refresh to 0.

**R100:** If using 100kΩ NTC, set R100 = 1; if using 10kΩ NTC, set R100 = 0.

**0:** Bit must be written 0. Do not write 1.

**ModelID:** Choose from one of the following lithium models. For most batteries, use ModelID = 0.

- **ModelID = 0:** Use for most lithium cobalt oxide variants (a large majority of lithium in the marketplace). Supported by EZ without characterization.
- **ModelID = 2:** Use for lithium NCR or NCA cells such as Panasonic®. Supported by EZ without characterization.
- **ModelID = 6:** Use for lithium iron phosphate (LiFePO<sub>4</sub>). For better performance, a custom characterization is recommended in this case, instead of an EZ configuration.

**VChg:** Set VChg to 1 for a charge voltage higher than 4.25V (4.3V–4.4V). Set VChg to 0 for a 4.2V charge voltage.

Reserved: Read-only bit.

### **IChgTerm Register (1Eh)**

Register Type: Current

Initial Value: 0x0640 (250mA on 10mΩ)

The IChgTerm register allows the device to detect when a charge cycle of the cell has completed. IChgTerm should be programmed to the exact charge termination current used in the application. The device detects end of charge if all the following conditions are met:

- VFSOC Register > FullSOCThr Register
- AND IChgTerm x 0.125 < Current Register < IChgTerm x 1.25
- AND IChgTerm x 0.125 < AvgCurrent Register < IChgTerm x 1.25

See the *End-of-Charge Detection* section for more details.

### ***FullSOCThr Register (13h)***

Register Type: Percentage

Initial Value: 95%

The FullSOCThr register gates detection of end-of-charge. VFSOC must be larger than the FullSOCThr value before IChgTerm is compared to the AvgCurrent register value. The recommended FullSOCThr register setting for most custom characterized applications is 95% (default, 0x5F05). For EZ Performance applications the recommendation is 80% (0x5005). See the IChgTerm register description and *End-of-Charge Detection* section for details. Table 13 shows the register format.

**Table 13. FullSOCThr (13h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
FullSOCThr													1	0	1

### ***OCVTable0 (80h) to OCVTable15 (8Fh) Registers***

Register Type: Special

Cell characterization information used by the ModelGauge algorithm to determine capacity versus operating conditions. This table comes from battery characterization data.

### ***XTable0 (90h) to XTable15 (9Fh) Registers***

Register Type: Special

Cell characterization information used by the ModelGauge algorithm to determine capacity versus operating conditions. This table comes from battery characterization data.

### ***QRTTable00 (12h) to QRTTable30 (42h) Registers***

Register Type: Special

The QRTTable00 to QRTTable30 register locations contain characterization information regarding cell capacity under different application conditions.

### ***RComp0 Register (38h)***

Register Type: Special

The RComp0 register holds characterization information critical to computing the open circuit voltage of a cell under loaded conditions.

### ***TempCo Register (39h)***

Register Type: Special

The TempCo register holds temperature compensation information for the RComp0 register value.

## ModelGauge m5 Output Registers

The following registers are outputs from the ModelGauge m5 algorithm.

### ***RepCap Register (05h)***

Register Type: Capacity

RepCap or reported remaining capacity in mAh. This register is protected from making sudden jumps during load changes.

### ***AtAvCap Register (DFh)***

Register Type: Capacity

The AtAvCap register holds the estimated remaining capacity of the cell based on the theoretical load current value of the AtRate register. The value is stored in terms of  $\mu\text{Vh}$  and must be divided by the application sense-resistor value to determine the remaining capacity in mAh.

### ***RepSOC Register (06h)***

Register Type: Percentage

RepSOC is the reported state-of-charge percentage output for use by the application GUI.

### ***AtAvSOC Register (DEh)***

Register Type: Percentage

The AtAvSOC register holds the theoretical SOC of the cell based on the theoretical load of the AtRate register. The register value is stored as a percentage with a resolution of 1/256 % per LSB. The high byte indicates 1% resolution.

### ***FullCapRep Register (10h)***

Register Type: Capacity

This register reports the full capacity that goes with RepCap, generally used for reporting to the GUI. Most applications should only monitor FullCapRep, instead of FullCap or FullCapNom. A new full-capacity value is calculated at the end of every charge cycle in the application.

### ***FullCap Register (35h)***

Register Type: Capacity

FullCap is the full discharge capacity compensated according to the present conditions. A new full-capacity value is calculated continuously as application conditions change (temperature and load).

### ***FullCapNom Register (23h)***

Register Type: Capacity

FullCap is the full discharge capacity compensated according to the present conditions. A new full-capacity value is calculated continuously as application conditions change (temperature and load).

### ***TTE Register (11h)***

Register Type: Time

The TTE register holds the estimated time to empty for the application under present temperature and load conditions. The TTE value is determined by relating AvCap with AvgCurrent.

The corresponding AvgCurrent filtering gives a delay in TTE, but provides more stable results.

### ***AtTTE Register (DDh)***

Register Type: Time

The AtTTE register can be used to estimate time to empty for any theoretical load entered into the AtRate register.

### ***TTF Register (20h)***

Register Type: Time

The TTF register holds the estimated time to full for the application under present conditions. The TTF value is determined by learning the constant current and constant voltage portions of the charge cycle based on experience of prior charge cycles. Time to full is then estimated by comparing present charge current to the charge termination current. Operation of the TTF register assumes all charge profiles are consistent in the application.

### ***Cycles Register (17h)***

Register Type: Special

The Cycles register maintains a total count of the number of charge/discharge cycles of the cell that have occurred. The result is stored as a percentage of a full cycle. For example, a full charge/discharge cycle results in the Cycles register incrementing by 100%.

The Cycles register accumulates fractional or whole cycles. For example, if a battery cycles 10% x 10 times, then it tracks 100% of a cycle.

The Cycles register has a full range of 0 to 655.35 cycles with a 1% LSB.

### ***Status Register (00h)***

Register Type: Special

Initial Value: 0x0002 (change to 0x8082 immediately after POR)

The Status register maintains all flags related to alert thresholds and battery insertion or removal. Table 14 shows the Status register format.

**Table 14. Status (00h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Br	Smx	Tmx	Vmx	Bi	Smn	Tmn	Vmn	dSOCi	Imx	X	X	Bst	Imn	POR	X

**POR (Power-On Reset):** This bit is set to 1 when the device detects that a software or hardware POR event has occurred. This bit must be cleared by system software to detect the next POR event. POR is set to 1 at power-up.



**Imn and Imx (Minimum/Maximum Current Alert Threshold Exceeded):** These bits are set to a 1 whenever a Current register reading is below (Imn) or above (Imx) the IAlrtTh thresholds. These bits may or may not need to be cleared by system software to detect the next event. See the Config.IS bit description. Imn and Imx are cleared to 0 at power-up.

**Vmn and Vmx (Minimum/Maximum Voltage Alert Threshold Exceeded):** These bits are set to a 1 whenever a VCell register reading is below (Vmn) or above (Vmx) the VAlrtTh thresholds. These bits may or may not need to be cleared by system software to detect the next event. See the Config.VS bit description. Vmn and Vmx are cleared to 0 at power-up.

**Tmn and Tmx (Minimum/Maximum Temperature Alert Threshold Exceeded):** These bits are set to a 1 whenever a Temperature register reading is below (Tmn) or above (Tmx) the TAlrtTh thresholds. These bits may or may not need to be cleared by system software to detect the next event. See the Config.TS bit description. Tmn and Tmx are cleared to 0 at power-up.

**Smn and Smx (Minimum/Maximum SOC Alert Threshold Exceeded):** These bits set to 1 when the SOC is below (Smn) or above (Smx) the SAlrtTh thresholds. These bits might or might not need to be cleared by system software to detect the next event. See the Config.SS description. Smn and Smx are cleared to 0 at power-up.

**Bst (Battery Status):** This bit is useful when the IC is used in a host-side application. This bit is set to 0 when a battery is present in the system and set to 1 when the battery is absent. Bst is set to 0 at power-up.

**dSOCi (State of Charge 1% Change Alert):** This bit is set to 1 when the RepSOC register crosses an integer percentage boundary such as 50.0%, 51.0%, etc. The bit must be cleared by host software. dSOCi is set to 1 at power-up.

**Bi (Battery Insertion):** This bit is useful when the IC is used in a host-side application. This bit is set to 1 when the device detects that a battery has been inserted into the system by monitoring the TH pin. This bit must be cleared by system software to detect the next insertion event. Bi is set to 0 at power-up.

**Br (Battery Removal):** This bit is useful when the IC is used in a host-side application. Br is set to 1 when the system detects that a battery has been removed from the system. This bit must be cleared by system software to detect the next removal event. Br is set to 1 at power-up.

**X (Don't Care):** This bit is undefined and can be logic 0 or 1.

### ***Age Register (07h)***

Register Type: Percentage

The Age register contains a calculated percentage value of the application's present cell capacity compared to its original design capacity. The result can be used by the host to gauge the battery pack health as compared to a new pack of the same type. The equation for the register output is:

$$\text{Age Register (\%)} = 100\% \times (\text{FullCapRep Register} / \text{DesignCap Register})$$

For example, if DesignCap = 2000mAh and FullCapRep = 1800mAh, then Age = 90% (or 0x5A00)

### ***TimerH (BEh) and Timer (3Eh) Register***

Register Type: Special

Initial Value: 0x0000

TimerH and Timer provide a long-duration time count since the last POR. A 3.2-hour LSb gives a full-scale range for the register of up to 23.94 years. The Timer register LSb is 175.8ms, giving a full-scale range of 0 to 3.2 hours. TimerH and Timer can be interpreted together as a 32-bit timer.

### ***RCell Register (14h)***

Register Type: Resistance

Initial Value: 0x0290 (160mΩ)

The RCell register provides the calculated internal resistance of the cell. RCell is determined by comparing open-circuit voltage (VFOCV) against measured voltage (VCell) over a long time period while under load or charge current.

### ***VRipple Register (BCh)***

Register Type: Special

Initial Value: 0x0000

The VRipple register holds the slow average RMS ripple value of VCell register reading variation compared to the AvgVCell register. The default filter time is 22.5 seconds. See the RippleCfg register description. VRipple has an LSb weight of 1.25mV/128.

## **ModelGauge m5 Algorithm Configuration Registers**

The following registers allow operation of the ModelGauge m5 algorithm to be adjusted for the application. It is recommended that the default values for these registers be used.

### ***AtRate Register (04h)***

Register Type: Current

Host software should write the AtRate register with a negative two's complement 16-bit value of a theoretical load current prior to reading any of the at-rate output registers (AtTTE, AtAvSOC, AtAvCap).

### ***FilterCfg Register (29h)***

Register Type: Special

Initial Value: 0xCEA4

The FilterCfg register sets the average time period for all ADC readings, for mixing OCV results and coulomb-count results. It is recommended that these values are not changed unless absolutely required by the application. Table 15 shows the FilterCfg register format.

**Table 15. FilterCfg (29h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1	1	TEMP			MIX			VOLT			CURR				

**CURR:** Sets the time constant for the AvgCurrent register. The default POR value of 0100b gives a time constant of 5.625s. The equation setting the period is:

$$\text{AvgCurrent time constant} = 45\text{s} \times 2^{\text{(CURR-7)}}$$

**VOLT:** Sets the time constant for the AvgVCell register. The default POR value of 010b gives a time constant of 45.0s. The equation setting the period is:

$$\text{AvgVCell time constant} = 45\text{s} \times 2^{\text{(VOLT-2)}}$$

**MIX:** Sets the time constant for the mixing algorithm. The default POR value of 1101b gives a time constant of 12.8 hours. The equation setting the period is:

$$\text{Mixing Period} = 45\text{s} \times 2^{\text{(MIX-3)}}$$

**TEMP:** Sets the time constant for the AvgTA register. The default POR value of 0001b gives a time constant of 1.5min. The equation setting the period is:

$$\text{AvgTA time constant} = 45\text{s} \times 2^{\text{TEMP}}$$

**1:** Write these bits to 1. Do not write 0.

### ***RelaxCfg Register (2Ah)***

Register Type: Special

Initial Value: 0x2039

The RelaxCfg register defines how the IC detects whether the cell is in a relaxed state with a low dV/dt. Figure 12 describes relaxation detection. If AvgCurrent remains below the LOAD threshold while AvgVCell changes less than the dV threshold over two consecutive periods of dt, the cell is considered relaxed. Table 16 shows the RelaxCfg register format.

**Table 16. RelaxCfg (2Ah) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
LOAD							dV				dt				

**LOAD:** Sets the threshold, which the AvgCurrent and Current registers are compared against. The AvgCurrent and Current registers must remain below this threshold value for the cell to be considered unloaded. Load is an unsigned 7-bit value, where 1 LSb = 50μV (5mA on 10mΩ). The default value is 800μV (80mA on 10mΩ).

**dV:** Sets the change threshold, which AvgVCell is compared against. If the cell voltage changes by less than dV over two consecutive periods set by dt, the cell is considered relaxed; dV has a range of 0 to 40mV where 1 LSb = 1.25mV. The default value is 3.75mV.

**dt:** Sets the time period over which the change in AvgVCell is compared against dV. If the cell voltage changes by less than dV over two consecutive periods set by dt, the cell is considered relaxed. The default value is 1.5 minutes. The comparison period is calculated as:

$$\text{Relaxation Period} = 2(\text{dt} - 8) \times 45\text{s}$$

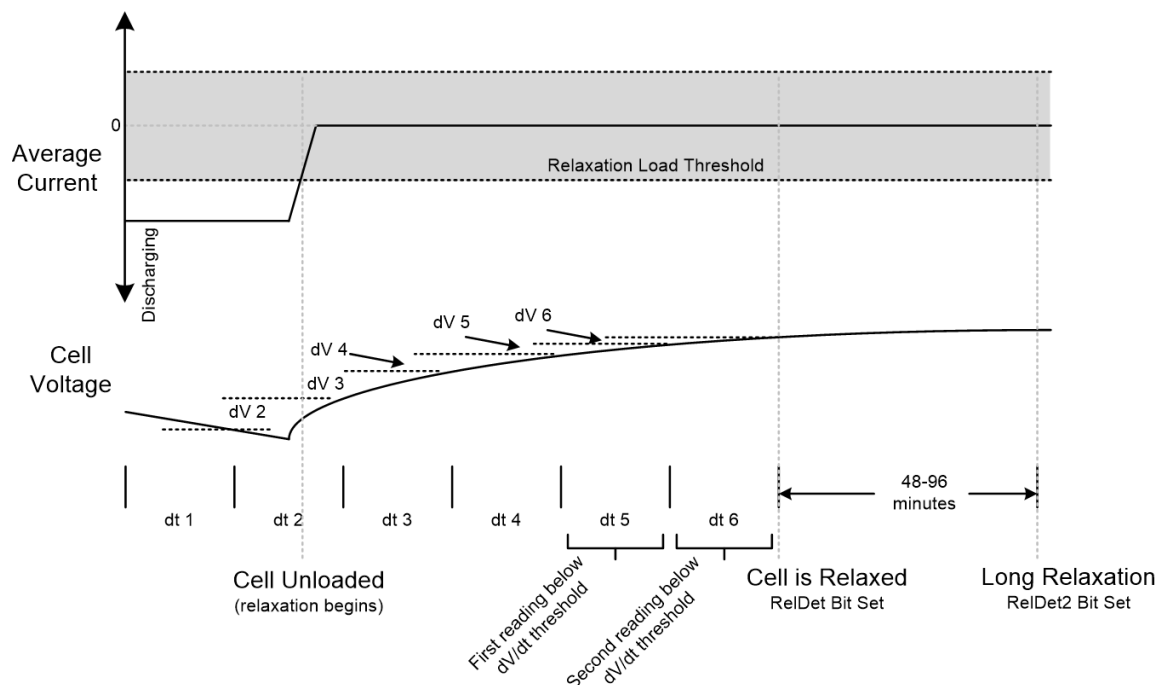


Figure 12. Cell relaxation detection.

### LearnCfg Register (28h)

Register Type: Special

Initial Value: 0x4486

The LearnCfg register controls all functions relating to adaptation during operation. Table 17 shows the register format.

Table 17. LearnCfg (28h) Format

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	0	0	1	0	0	1	LS			0	1	1	0

**0:** Bit must be written 0. Do not write 1.

**1:** Bit must be written 1. Do not write 0.

**LS:** Learn Stage. The Learn Stage value controls the influence of the voltage fuel gauge on the mixing algorithm. Learn Stage defaults to 0h, making the voltage fuel gauge dominate. Learn Stage then advances to 7h over the course of two full cell cycles to make the coulomb counter dominate. Host software can write the Learn Stage value to 7h to advance to the final stage at any time. Writing any value between 1h and 6h is ignored.

### MiscCfg Register (2Bh)

Register Type: Special

Initial Value: 0x3870

The MiscCfg control register enables various other functions of the device. The MiscCfg register default values should not be changed unless specifically required by the application. Table 18 shows the register format.

**Table 18. MiscCfg (2Bh) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
FUS				1	0	MR					1	0	0	SACFG	

**0:** Bit must be written 0. Do not write 1.

**1:** Bit must be written 1. Do not write 0.

**SACFG:** SOC Alert Config. SOC Alerts can be generated by monitoring any of the SOC registers as follows. SACFG defaults to 0b00 at power-up:

- 00: SOC Alerts are generated based on the RepSOC register.
- 01: SOC Alerts are generated based on the AvSOC register.
- 10: SOC Alerts are generated based on the MixSOC register.
- 11: SOC Alerts are generated based on the VFSOC register.

**MR:** Mixing Rate. This value sets the strength of the servo mixing rate after the final mixing state has been reached (> 2.08 complete cycles). The units are MR0 = 6.25μV, giving a range up to 19.375mA with a standard 10mΩ sense resistor. Setting this value to 00000b disables servo mixing and the MAX1726x continues with time-constant mixing indefinitely. The default setting is 18.75μV or 1.875mA with a standard sense resistor.

**FUS:** Full Update Slope. This value prevents jumps in the RepSOC and FullCapRep registers by setting the rate of adjustment of FullCapRep near the end of a charge cycle. The update slope adjustment range is from 2% per 15 minutes (0000b) to a maximum of 32% per 15 minutes (1111b).

### **ConvCfgr Register (49h)**

Register Type: Special

The ConvCfgr register configures operation of the converge-to-empty feature. The default and recommended value for ConvCfgr is 0x2241.

### **RippleCfgr Register (BDh)**

Register Type: Special

The RippleCfgr register configures ripple measurement and ripple compensation. The default and recommended value for this register is 0x0204. Table 19 shows the register format.

**Table 19. RippleCfgr (BDh) Format**

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

**NR:** Ripple Measurement Filter. Sets the filter magnitude for ripple observation as defined by the following equation giving a range of 1.4 seconds to 180 seconds.

Ripple Time Range = 1.4 seconds × 2NR

**kDV:** Ripple Empty Compensation Coefficient. Configures MAX1726x to compensate the fuel gauge % according to the ripple.

## ModelGauge m5 Algorithm Additional Registers

The following registers contain intermediate ModelGauge m5 data that might be useful for debugging or performance analysis. The values in these registers initially update within 710ms after the IC is reset.

### ***dQAcc Register (45h)***

Register Type: Capacity

This register tracks change in battery charge between relaxation points. It is available to the user for debug purposes.

### ***dPAcc Register (46h)***

Register Type: Percentage (1/16% per LSB)

Initial Value: 0x0190 (25%)

This register tracks change in battery SOC between relaxation points. It is available to the user for debug purposes.

### ***QResidual Register (0Ch)***

Register Type: Capacity

The QResidual register provides the calculated amount of charge in mAh that is currently inside of, but cannot be removed from, the cell under present application conditions (load and temperature). This value is subtracted from the MixCap value to determine capacity available to the user under present conditions (AvCap).

### ***AtQResidual Register (DCh)***

Register Type: Capacity

The AtQResidual register provides the calculated amount of charge in mAh that is currently inside of, but cannot be removed from, the cell under present temperature and hypothetical load (AtRate).

This value is subtracted from the MixCap value to determine capacity available to the user (AtAvCap).

### ***VFSOC Register (FFh)***

Register Type: Percentage

The VFSOC register holds the calculated present SOC of the battery according to the voltage fuel gauge.

### ***VFOCV Register (FBh)***

Register Type: Voltage

The VFOCV register contains the calculated open-circuit voltage of the cell as determined by the voltage fuel gauge. This value is used in other internal calculations.

### ***QH Register (4Dh)***

Register Type: Capacity

Initial Value: 0x0000

The QH register displays the raw coulomb count generated by the device. This register is used internally as an input to the mixing algorithm. Monitoring changes in QH over time can be useful for debugging device operation.

### **AvCap (1Fh) and AvSOC (0Eh) Registers**

Register Type: Capacity (AvCap), Percentage (AvSOC)

The AvCap and AvSOC registers hold the calculated available capacity and percentage of the battery based on all inputs from the ModelGauge m5 algorithm, including empty compensation. These registers provide unfiltered results. Jumps in the reported values can be caused by abrupt changes in load current or temperature. See the *Empty Compensation* section for details.

### **MixCap (0Fh) and MixSOC (0Dh) Registers**

Register Type: Capacity (MixCap) and Percentage (MixSOC)

The MixCap and MixSOC registers hold the calculated remaining capacity and percentage of the cell before any empty compensation adjustments are performed.

See the *Empty Compensation* section for details.

### **VFRemCap Register (4Ah)**

Register Type: Capacity

The VFRemCap register holds the remaining capacity of the cell as determined by the voltage fuel gauge before any empty compensation adjustments are performed.

See the *Empty Compensation* section for details.

### **FStat Register (3Dh)**

Register Type: Special

The FStat register is a read-only register that monitors the status of the ModelGauge m5 algorithm. Table 20 is the FStat register format.

**Table 20. FStat (3Dh) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	ReIDt	EDet	FQ	ReIDt2	X	X	X	X	X	DNR

**DNR:** Data Not Ready. This bit is set to 1 at cell insertion and remains set until the output registers have been updated. Afterward, the IC clears this bit, indicating the fuel gauge calculations are up to date. This takes 710ms from power-up.

**FQ:** Full Qualified. This bit is set when all charge termination conditions have been met. See the *End-of-Charge Detection* section for details.

**EDet:** Empty Detection. This bit is set to 1 when the IC detects that the cell empty point has been reached. This bit is reset to 0 when the cell voltage rises above the recovery threshold. See the VEmpty register for details.

**ReIDt:** Relaxed Cell Detection. This bit is set to 1 when the ModelGauge m5 algorithm detects that the cell is in a fully relaxed state. This bit is cleared to 0 when a current greater than the Load threshold is detected. See Figure 12.

**RelDt2:** Long Relaxation. This bit is set to 1 when the ModelGauge m5 algorithm detects that the cell has been relaxed for a period of 48 to 96 minutes or longer. This bit is cleared to 0 when the cell is no longer in a relaxed state. See Figure 12.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

## Status and Configuration Registers

The following registers control IC operation not related to the fuel gauge such as power-saving modes and ALRT pin functionality.

### **Config Register (1Dh) and Config2 Register (BBh)**

Register Type: Special

See individual data sheet for details.

### **DevName Register (21h)**

Register Type: Special

The DevName register holds device type and firmware revision information. This allows host software to easily identify the type of IC being communicated to.

**Table 21. DevName Register Values for Different Variants of the MAX1726x**

PART	ADDRESS	FEATURE
MAX17260	0x4031	Single-cell optional high-side sensing
MAX17261	0x4033	Multi-cell fuel gauge
MAX17262	0x4039	Internal current sensing
MAX17263	0x4037	Single/multi-cell with integrated LED driver

### **ShdnTimer Register (3Fh)**

Register Type: Special

Initial Value: 0x0000

The ShdnTimer register sets the time-out period from when a shutdown event is detected until the device disables the regulators and enters low-power mode. Table 22 shows the ShdnTimer register format.

**Table 22. ShdnTimer (3Fh) Format**

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

**CTR:** Shutdown Counter. This register counts the total amount of elapsed time since the shutdown trigger event. This counter value stops and resets to 0 when the shutdown time-out completes. The counter LSB is 1.4s.

**THR:** Sets the shutdown time-out period from a minimum of 45s to a maximum of 1.6h. The default POR value of 0h gives a shutdown delay of 45s. The equation setting the period is:

$$\text{Shutdown Time-Out Period} = 175.8\text{ms} \times 2^{(8+\text{THR})}$$



### Status2 Register (B0h)

Register Type: Special

Initial Value: 0x0000

The Status2 register maintains status of various firmware functions. Table 23 shows the Status2 register format.

**Table 23. Status2 (B0h) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	AtRateReady	DPRReady	X	X	X	SNReady	X	X	FullDet	X	X	X	Hib	X

**Hib:** Hibernate Status. This bit is set to a 1 when the device is in hibernate mode or 0 when the device is in active mode. Hib is set to 0 at power-up.

**FullDet:** Full Detected.

For the following 3 bits, see also the *Serial Number Feature* section for more details.

**SNReady:** If SNReady = 1, the unique serial number is available over the I<sup>2</sup>C. This bit is set to 1 by firmware after the serial number is read internally and placed into RAM. Serial number overwrites Dynamic Power and AtRate output registers as described in the *Serial Number Feature* section.

**AtRateReady:** If AtRateReady = 1, AtRate output registers are filled by the firmware and ready to be read by the host.

**DPRReady:** If DPRReady = 1, Dynamic Power output registers are filled by the firmware and ready to be read by the host.

**X:** Don't Care. This bit is undefined and can be logic 0 or 1.

### HibCfg Register (BAh)

Register Type: Special

Initial Value: 0x870C

The HibCfg register controls hibernate mode functionality. The MAX1726x enters and exits hibernate when the battery current is less than approximately C/100. While in hibernate mode, the MAX1726x reduces its operating current to 5μA by reducing ADC sampling to once every 5.625s. Table 24 shows the register format.

**Table 24. HibCfg (BAh) Format**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
EnHib	HibEnterTime			HibThreshold				0	0	0	HibExitTime		HibScalar		

**0:** Bit must be written 0. Do not write 1.

**HibScalar:** Sets the task period while in hibernate mode based on the following equation:

$$\text{Hibernate Mode Task Period (s)} = 351\text{ms} \times 2^{(\text{HibScalar})}$$

**HibExitTime:** Sets the required time period of consecutive current readings above the HibThreshold value before the IC exits hibernate and returns to active mode of operation.

$$\text{Hibernate Mode Exit Time (s)} = (\text{HibExitTime} + 1) \times 702\text{ms} \times 2^{(\text{HibScalar})}$$

**HibThreshold:** Sets the threshold level for entering or exiting hibernate mode. The threshold is calculated as a fraction of the full capacity of the cell using the following equation:

$$\text{Hibernate Mode Threshold (mA)} = \frac{\text{Full Cap (mAh)} / 0.8\text{hrs}}{2^{(\text{HibThreshold})}}$$

**HibEnterTime:** Sets the time period that consecutive current readings must remain below the HibThreshold value before the IC enters hibernate mode, as defined by the following equation. The default HibEnterTime value of 000b causes the IC to enter hibernate mode if all current readings are below the HibThreshold for a period of 5.625 seconds, but the IC could enter hibernate mode as quickly as 2.812 seconds.

$$2.812\text{s} \times 2^{(\text{HibEnterTime})} < \text{Hibernate Mode Entry Time} < 2.812\text{s} \times 2^{(\text{HibEnterTime}+1)}$$

**EnHib:** Enable Hibernate Mode. When set to 1, the IC will enter hibernate mode if conditions are met. When set to 0, the IC always remains in the active mode of operation.

### ***Soft-Wakeup (Command Register 60h)***

Register Type: Special

The Command register accepts commands to perform functions listed in the following table.

Command	Mnemonic	Description
0000h	Clear	Clears all commands.
0090h	Soft wakeup	Wakes up the fuel gauge from hibernate mode to reduce the response time of the IC to configuration changes. This command must be manually cleared (0000h) afterward to keep proper fuel gauge timing.

To wake and exit hibernate:

1. Write HibCfg = 0x0000.
2. Soft-Wakeup Command. Write Command Register (60h) to 0x0090.
3. Clear Command. Write Command Register (60h) to 0x0000.
4. Eventually restore HibCfg to again allow automatic hibernate decisions.

## Modes of Operation

The IC operates in one of three power modes: shutdown, hibernate, and active. While in active mode, the IC operates as a high-precision fuel gauge with temperature, voltage, current, and accumulated current measurements acquired continuously, and the resulting values updated in the measurement registers. Hibernate mode is a fully functional reduced power (5 $\mu$ A) consumption version of active mode. In hibernate mode, all registers are updated every 5.625s instead of the normal 175.8ms. In shutdown mode, the internal LDO regulator is disabled, all activity stops, ADC register and fuel-gauge output values are lost.

**Entering Hibernate Mode.** Pack Idle: Hibernate mode must be enabled by setting HibCfg.EnHib = 1. The IC then enters hibernate mode if the current persists below about C/100.

**Exiting Hibernate Mode.** Pack Active: The IC returns to active mode if Current exceeds approximately C/100.

**Entering Shutdown Mode (from Active Mode or Hibernate Mode).** Because the MAX1726x supports a 1 $\mu$ A shutdown as well as a 5 $\mu$ A hibernate mode (fully functional and fuel gauging), it is generally best to keep the fuel gauge enabled to continue to track the battery state, even when the system is shut down. However, for small batteries and long storage periods, it might be necessary to save the additional 4 $\mu$ A.

- **Software Shutdown:** Software shutdown can be forced by setting Config.SHDN = 1. To command shutdown within 22.5 seconds, write ShdnTimer = 0x001E.
- **Pack Disconnect:** The IC enters shutdown if Config.COMMSH = 1 and communication lines are open (logic-low) for longer than the ShdnTimer period.

These shutdown entry modes are all programmable according to the application. Shutdown events are gated by the ShdnTimer register, which allows a delay between the shutdown event/command and entering the mode to prevent final I<sup>2</sup>C traffic from inadvertently re-awakening the IC.

**Exiting Shutdown Mode (IC Always Exits into Active Mode).**

- **Pack Connect.** The IC returns to active mode on any edge of any communication line.
- **IC Reset.** If the IC is power-cycled or the software RESET command is sent the IC returns to active mode of operation.

See the detailed descriptions of the ShdnTimer, HibCfg, and Config registers.

## I<sup>2</sup>C Bus System

The MAX1726x uses the standard I<sup>2</sup>C protocol. See the *I<sup>2</sup>C Protocols* section for specific protocol details.

### Hardware Configuration

The I<sup>2</sup>C bus system supports operation as a slave-only device in a single or multislave, and single or multimaster system. Up to 128 slave devices might share the bus using 7-bit slave addresses. The I<sup>2</sup>C interface consists of a serial data line (SDA) and serial clock line (SCL). SDA and SCL provide bidirectional communication between the IC and a master device at speeds up to 400kHz. The IC's SDA pin operates bidirectionally. When the IC receives data, SDA operates as an input. When the IC returns data, SDA operates as an open-drain output with the host system providing a resistive pull-up. See Figure 13. The IC always operates as a slave device, receiving and transmitting data under the control of a master device. The master initiates all transactions on the bus and generates the SCL signal, as well as the START and STOP bits which begin and end each transaction.

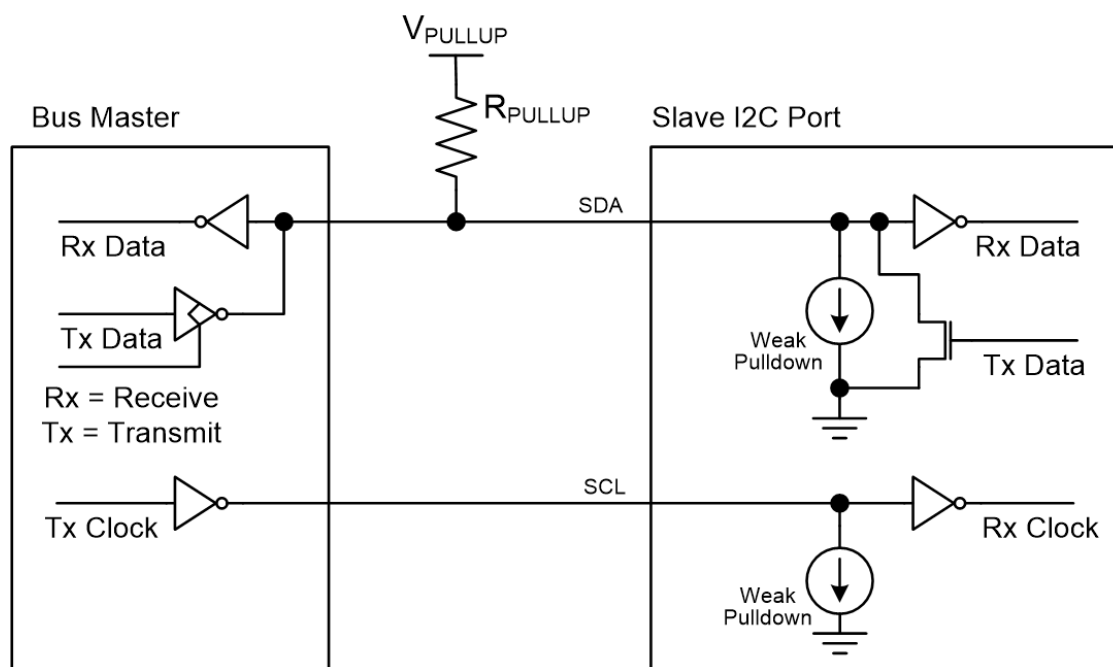


Figure 13. I<sup>2</sup>C bus interface.

### I/O Signaling

The following individual signals are used to build byte-level I<sup>2</sup>C communication sequences.

#### Bit Transfer

One data bit is transferred during each SCL clock cycle, with the cycle defined by SCL transitioning low to high and then high to low. The SDA logic level must remain stable during the high period of the SCL clock pulse. Any change in SDA when SCL is high is interpreted as a START or STOP control signal.

### ***Bus Idle***

The bus is defined to be idle, or not busy, when no master device has control. Both SDA and SCL remain high when the bus is idle. The STOP condition is the proper method to return the bus to the idle state.

### ***START and STOP Conditions***

The master initiates transactions with a START condition by forcing a high-to-low transition on SDA while SCL is high. The master terminates a transaction with a STOP condition by a low-to-high transition on SDA while SCL is high. A Repeated START condition can be used in place of a STOP then START sequence to terminate one transaction and begin another without returning the bus to the idle state. In multimaster systems, a Repeated START allows the master to retain control of the bus. The START and STOP conditions are the only bus activities in which the SDA transitions when SCL is high.

### ***Acknowledge Bits***

Each byte of a data transfer is acknowledged with an Acknowledge bit (ACK) or a No Acknowledge bit (NACK). Both the master and the IC slave generate acknowledge bits. To generate an Acknowledge, the receiving device must pull SDA low before the rising edge of the acknowledge-related clock pulse (ninth pulse) and keep it low until SCL returns low. To generate a No Acknowledge, the receiver releases SDA before the rising edge of the acknowledge-related clock pulse and leaves SDA high until SCL returns low. Monitoring the acknowledge bits allows for detection of unsuccessful data transfers. An unsuccessful data transfer can occur 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 attempt communication. If a transaction is aborted mid-byte, the master should send additional clock pulses to force the slave IC to free the bus prior to restarting communication.

### ***Data Order***

With I<sup>2</sup>C communication, a byte of data consists of 8 bits ordered most significant bit (MSb) first. The least significant bit (LSb) of each byte is followed by the Acknowledge bit. IC registers comprising multibyte values are ordered least significant byte (LSB) first.

### ***Slave Address***

A bus master initiates communication with a slave device by issuing a START condition followed by a Slave Address and the read/write (R/W) bit. When the bus is idle, the IC continuously monitors for a START condition followed by its slave address. When the IC receives a slave address that matches its Slave Address, it responds with an Acknowledge bit during the clock period following the R/W bit. The MAX1726x supports the slave address 0x6C (or 0x36 for 7 MSb address), MAX17260 has a variant with slave address of 0x1A available for order.

### ***Read/Write Bit***

The R/W bit following the slave address determines the data direction of subsequent bytes in the transfer. R/W = 0 selects a write transaction, with the following bytes being written by the master to the slave. R/W = 1 selects a read transaction, with the following bytes being read from the slave by the master.

## Bus Timing

The IC is compatible with any bus timing up to 400kHz. Refer to the MAX17260/MAX17261/MAX17262/MAX17263 data sheet Electrical Characteristics table for timing details. No special configuration is required to operate at any speed. Figure 14 shows an example of standard I<sup>2</sup>C bus timing.

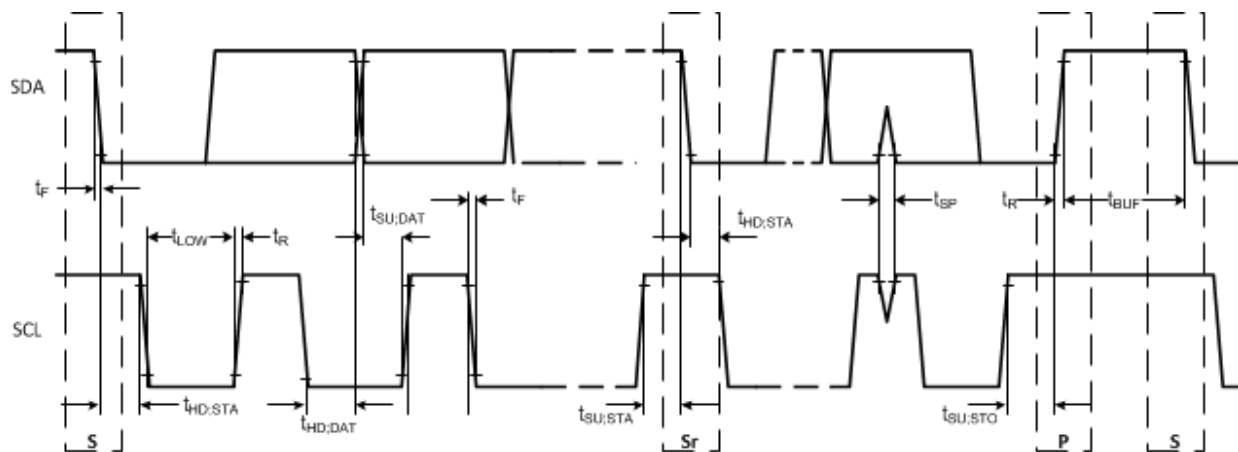


Figure 14. I<sup>2</sup>C bus timing diagram.

## I<sup>2</sup>C Protocols

The following I<sup>2</sup>C communication protocols must be used by the bus master to access MAX1726x memory locations 00h to FFh. These protocols follow the standard I<sup>2</sup>C specification for communication.

### I<sup>2</sup>C Write Data Protocol

The Write Data protocol is used to transmit data to the IC 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 15 for an example Write Data communication sequence.

### Write Data Communication Protocol



### Example Word Write to I2C Command Register Address 060h

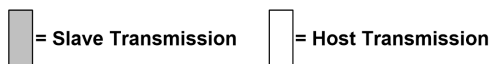
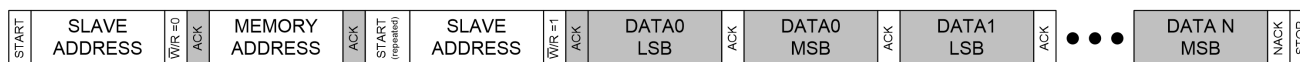


Figure 15. Example I<sup>2</sup>C Write Data communication sequence.

### I<sup>2</sup>C Read Data Protocol

The Read Data protocol is used to transmit data from IC memory locations 00h to FFh. The memory address is sent by the bus master as a single byte value immediately after the slave address. Immediately following the memory address, the bus master issues a REPEATED START followed by the slave address. The MAX1726x sends an ACK to acknowledge the address and begins transmitting data. A word of data is read as two separate bytes that the master must acknowledge. Because the address is automatically incremented after the final bit of each 16-bit word received by the IC, the LSB of the data at the next memory address can be read immediately after the acknowledgment of the MSB of data at the previous address. The master indicates the end of a read transaction by sending a NACK followed by a STOP. If the bus master continues an auto-incremented read transaction beyond memory address FFh, the IC transmits all 1s until a NACK or STOP is received. Data from reserved address locations is undefined. See Figure 16 for an example Read Data communication sequence.

### I2C Read Data Communication Protocol



### Example Read Data of Current and AvgCurrent Registers Addresses 00Ah-00Bh

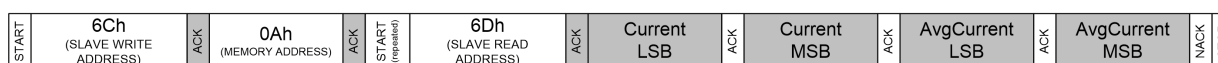


Figure 16. Example I<sup>2</sup>C Read Data communication sequence.

## Trademarks

Fenwal is a registered trademark of Fenwal, Inc.

Intel is a registered trademark of Intel Corporation.

ModelGauge is a trademark of Maxim Integrated Products, Inc.

Panasonic is a registered trademark of Panasonic Corporation.

TDK is a registered trademark of TDK Corporation.



## Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	3/18	Initial release	—
1	5/18	Updated the Overview, Current Measurements, Copper Trace Current Sensing, Voltage Measurements, and Bus Timing sections, Table 3, Table 4, Table 21, and the CNFG_CGH_A Register Table	8, 18–21, 40, 46
2	6/18	Updated Abstract	1
3	11/19	Added features for MAX17262, MAX17263; added table; updated table numbering, register number, and references	8, 17–21, 26

©2019 by Maxim Integrated Products, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. MAXIM INTEGRATED PRODUCTS, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. MAXIM ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering or registered trademarks of Maxim Integrated Products, Inc. All other product or service names are the property of their respective owners.