



MPF42790

2 to 16 Stacked Cells Battery Pack Fuel Gauge with Level LEDs

DESCRIPTION

The MPF42790 is a drop-in solution to provide comprehensive status information on lithium-ion battery strings up to 16 series cells. The MPF42790 estimates the internal state-of-charge (SOC) and state-of-health (SOH) for each cell, as well as the full pack. The MPF42790 also determines remaining runtime, charge time, and instantaneous available power. On-board memory enables the lifetime logging of key parameters.

The MPF42790 is simple to use and supports a variety of lithium cell chemistries and cell sizes. A few basic configurations allow for quick set-up, and advanced configuration capabilities can fine-tune the device for specific applications.

When paired with an MP279x battery monitor, the MPF42790 can achieve SOC accuracy within 3%. The MPF42790 can also be paired with other analog front-end (AFE) or battery stack monitors. The 400kHz I²C interface provides standard, robust communication. The 5-level LED drivers provide a simple, cost-effective charge level indicator for the device.

The MPF42790 is available in a TQFN-32 (4mmx4mm) package.

MINIMUM SYSTEM REQUIREMENTS

- Battery Monitor Providing Individual Cell Voltages for 2 to 16 Series Cells, Pack Current, and Temperature
- M0 System MCU with I²C and Interrupt, 48kB of Flash and 4kB of RAM

FEATURES

- Compatible with Commonly used Battery Monitors for up to 16 Series Cells
 - ±3% State-of-Charge (SOC) Accuracy when Paired with MP279x Battery Monitors
- Provides Critical Battery Information:
 - Pack and Cell SOC
 - Pack and Cell State-of-Health (SOH)
 - Remaining Runtime and Charge Time
 - Instantaneous Available Power
- Supports a Wide Variety of Lithium Cells
- Adaptive Learning can be Enabled to:
 - Refine Initial Charge Settings
 - Refine Initial Discharge Settings
 - Update Individual Cells' SOH to Track Degradation
- Provides Lifetime Logging
- Supports 5-Level LED SOC Indicator with Push-Button Trigger
- 2.5V Minimum Supply Voltage
- Low Current Consumption: 6μA in Disabled Mode and 135μA (Average) in Operating Mode during Rest
- Supports Up to 400kHz I²C with CRC for Robust Communication
- Available in a Compact TQFN-32 (4mmx4mm) Package
- Available in Turn-Key MPS BMS Module:
 - MBM1xS-P50 and MBM1xS-P100

APPLICATIONS

- Light EV: Scooters, Bikes, and Golf Carts
- Energy Storage: UPS and Renewable Energy
- Industrial Robots, Floor Cleaners, and Forklifts
- Cordless Tools and Gardening Equipment

All MPS parts are lead-free, halogen-free, and adhere to the RoHS directive. For MPS green status, please visit the MPS website under Quality Assurance. "MPS", the MPS logo, and "Simple, Easy Solutions" are trademarks of Monolithic Power Systems, Inc. or its subsidiaries.

TYPICAL APPLICATION

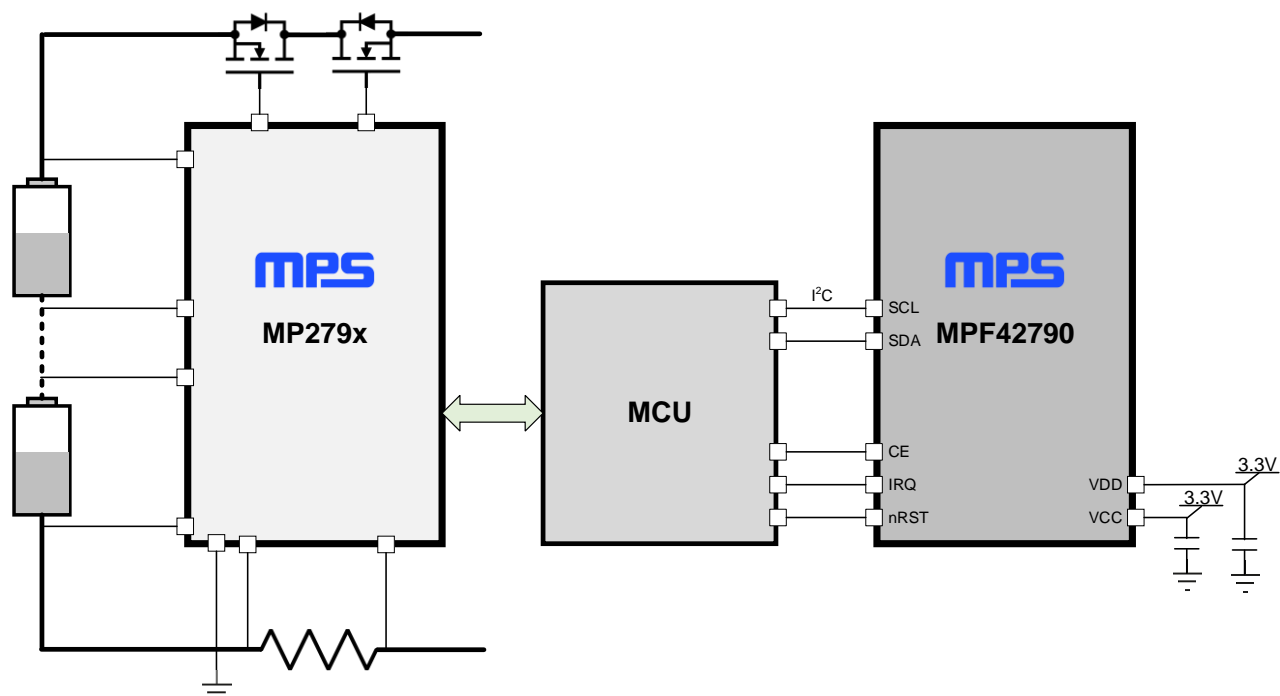


Figure 1: MPF42790 with MP279x AFE Typical Electrical Diagram

ORDERING INFORMATION

| Part Number* | Package | Top Marking | MSL Rating |
|-----------------------|-------------------|-------------|------------|
| MPF42790DRT-0B-yyyy** | TQFN-32 (4mmx4mm) | See Below | 3 |
| EVKT-MPF42790 | Evaluation kit | - | - |
| EVKT-MBM1xS-P50 | Evaluation kit | - | - |

* For Tape & Reel, add suffix -Z (e.g. MPF42790DRT-0B-yyyy-Z).

** "-yyyy" refers to the default configuration identifier ("-0000" by default), where each "y" is a hexadecimal value between 0 and F. Work with an MPS FAE to obtain a customized default configuration.

TOP MARKING

MPSYWW
M4279X
LLLLLL

MPF4279x Family Generic Version

MPS : MPS prefix
 Y : Year code
 WW : Week code
 M4279X : Family part number*
 LLLLLL : Lot number

* The specific part number is in the IC_VER register.

EVALUATION KIT EVKT-MBM1XS-P50 (MP279x BMS)

EVKT-MBM1xS-P50 kit contents (items below can be ordered separately):

| # | Part Number | Item | Quantity |
|---|-----------------------|--|----------|
| 1 | MBM1xS-P50 | MPF42790DRT-0B-yyyy and MP279x AFE reference design evaluation board | 1 |
| 2 | EVKT-USB_RS232/I2C-01 | USB to RS232/I ² C adapter | 1 |
| 3 | Online resources | Include datasheet, user guide, product brief, and GUI | 1 |

Order directly from MonolithicPower.com or our distributors.

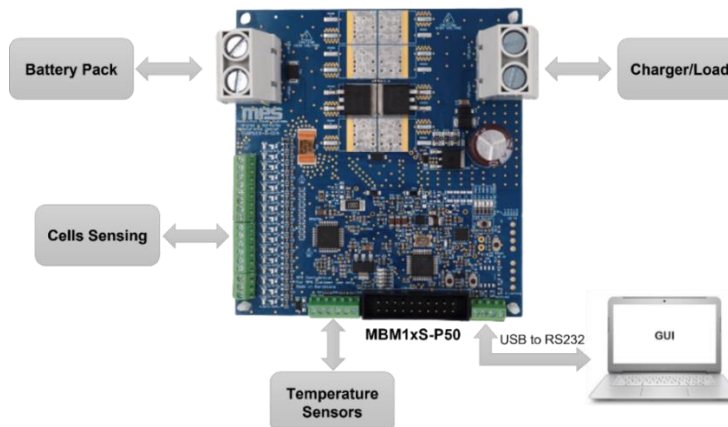
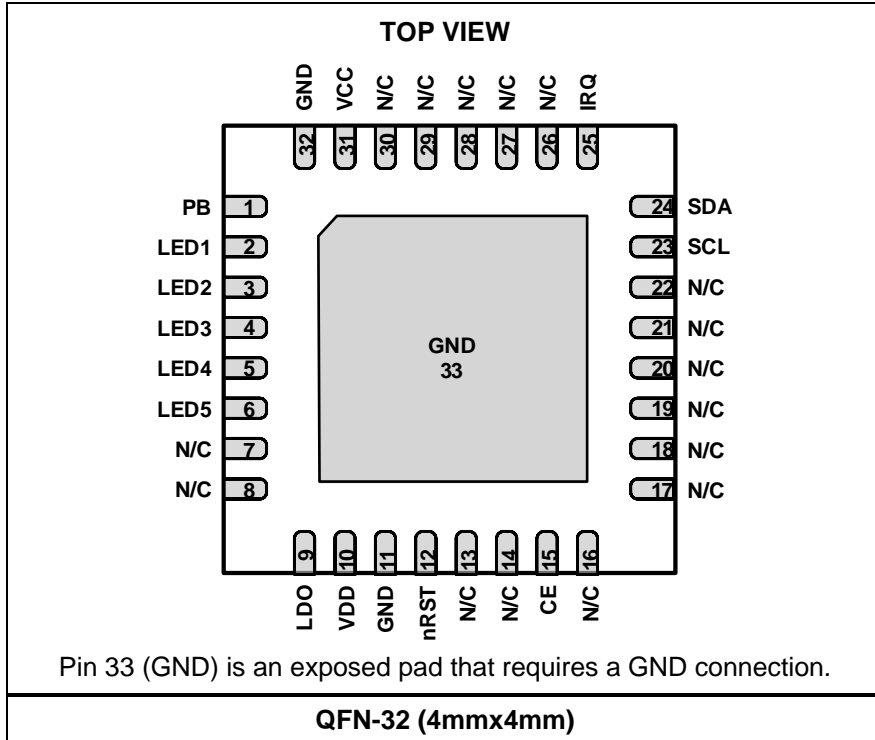


Figure 2: MBM1xS-P50 Evaluation Kit Set-Up

PACKAGE REFERENCE



PIN FUNCTIONS

| Pin # | Name | I/O | Description |
|-------|------|-------|---|
| 1 | PB | I | Push-button. Triggers state-of-charge (SOC) LED indicator. |
| 2 | LED1 | O | LED 1 driver. The LED1 pin reports the SOC with a 330Ω resistor in series. |
| 3 | LED2 | O | LED 2 driver. The LED2 pin reports the SOC with a 330Ω resistor in series. |
| 4 | LED3 | O | LED 3 driver. The LED3 pin reports the SOC with a 330Ω resistor in series. |
| 5 | LED4 | O | LED 4 driver. The LED4 pin reports the SOC with a 330Ω resistor in series. |
| 6 | LED5 | O | LED 5 driver. The LED5 pin reports the SOC with a 330Ω resistor in series. |
| 7 | NC | - | No connection. |
| 8 | NC | - | No connection. |
| 9 | LDO | Power | Internal LDO. Bypass the LDO pin with a 2.2μF + 100nF ceramic capacitor connected to ground. |
| 10 | VDD | Power | Power supply input. Bypass VDD with a 2.2μF ceramic capacitor connected to ground. |
| 11 | GND | Power | Ground pin. |
| 12 | nRST | I | IC reset control. |
| 13 | NC | - | No connection. |
| 14 | NC | - | No connection. |
| 15 | CE | I | Chip enabled. Set the CE pin to stop fuel gauge updates and disable the communication interface to minimize current consumption. |
| 16 | NC | - | No connection. |
| 17 | NC | - | No connection. |
| 18 | NC | - | No connection. |
| 19 | NC | - | No connection. |
| 20 | NC | - | No connection. |
| 21 | NC | - | No connection. |
| 22 | NC | - | No connection. |
| 23 | SCL | I/O | I²C interface clock. Connect the SCL pin to the logic rail through a 10kΩ resistor. |
| 24 | SDA | I/O | I²C interface data. Connect the SDA pin to the logic rail through a 10kΩ resistor. |
| 25 | IRQ | O | Interrupt request pin. The IRQ pin is the interrupt going to the host system(s). |
| 26 | NC | - | No connection. |
| 27 | NC | - | No connection. |
| 28 | NC | - | No connection. |
| 29 | NC | - | No connection. |
| 30 | NC | - | No connection. |
| 31 | VCC | Power | 3V to 3.3V power supply input. Bypass the VCC pin with a 2.2μF ceramic capacitor connected to ground. |
| 32 | GND | Power | Ground pin. |
| 33 | GND | Power | Ground pin. |

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾

| | |
|---|--------------------------|
| VCC supply voltage (V_{CC}) | 3.6V |
| VDD supply voltage (V_{DD}) | 3.6V |
| nRST pin..... | -0.3V to +3.6V |
| All other pins | -0.3V to $V_{CC} + 0.3V$ |
| Total power dissipation ($T_A = 25^\circ C$) ⁽²⁾ ... | 500mW |
| Storage temperature | -55°C to +150°C |
| Junction temperature | -55°C to +150°C |

ESD Ratings

Human body model (HBM) $\pm 4000V$

Recommended Operating Conditions ⁽³⁾

| | |
|---|--------------------------|
| Supply voltage (V_{DD}) | 3V to 3.6V |
| All other pins | -0.3V to $V_{CC} + 0.3V$ |
| Operating junction temp (T_J) | -40°C to +85°C |

| | | |
|--|---------------|---------------|
| Thermal Resistance ⁽⁴⁾ | θ_{JA} | θ_{JC} |
| TQFN-32 (4mmx4mm)..... | 47..... | 4.5....°C/W |

Notes:

- 1) Exceeding these ratings may damage the device.
- 2) The maximum allowable power dissipation is a function of the maximum junction temperature, T_J (MAX), the junction-to-ambient thermal resistance, θ_{JA} , and the ambient temperature, T_A . The maximum allowable continuous power dissipation at any ambient temperature is calculated by P_D (MAX) = $(T_J$ (MAX) - T_A) / θ_{JA} . Exceeding the maximum allowable power dissipation can produce an excessive die temperature, and the regulator may go into thermal shutdown. Internal thermal shutdown circuitry protects the device from permanent damage.
- 3) The device is not guaranteed to function outside of its operating conditions.
- 4) Measured on JESD51-7, 4-layer PCB.

ELECTRICAL CHARACTERISTICS

$V_{CC} = 3.3V$, $T_A = 25^{\circ}C$, unless otherwise noted.

| Parameters | Symbol | Condition | Min | Typ | Max | Units |
|--|---------------------|---|-----------------------|------|----------------------|---------|
| State-of-Charge Performance | | | | | | |
| Pack state-of-charge (SOC) accuracy ⁽⁵⁾ | θ_{ERR} | MPS MP279x AFE family, $15^{\circ}C \leq T_A \leq 35^{\circ}C$, $I_{CHG} \leq 0.5C$, $I_{DIS} \leq 0.5C$ | -2 | 0 | +2 | % |
| | | MPS MP279x AFE family, $10^{\circ}C \leq T_A \leq 50^{\circ}C$, $I_{CHG} \leq 1C$, $I_{DIS} \leq 1C$ | -3 | 0 | +3 | % |
| | | MPS MP279x AFE family, $10^{\circ}C \leq T_A \leq 50^{\circ}C$, $I_{CHG} \leq 2C$, $I_{DIS} \leq 2C$ | -4 | 0 | +4 | % |
| Power Supply | | | | | | |
| VDD operating voltage range | V_{DD} | | 2.5 | 3.3 | 3.6 | V |
| VCC operating voltage range | V_{CC} | | 2.5 | 3.3 | 3.6 | V |
| Total active current | I_{DD_ACTIVE} | $V_{DD} = 3.3V$, LEDs off, fuel gauge updating | | 7.5 | | mA |
| Total standby current | $I_{DD_STANDBY}$ | $V_{DD} = 3.3V$, LEDs off, fuel gauge idle | | 50 | | μA |
| Average operating current during CHG or DSG | $I_{DD_CHG/DSG}$ | $V_{DD} = 3.3V$, LEDs off, NCELLS_SER = 10, EXE_TIME = 4s, with I ² C traffic | | 270 | | μA |
| Average operating current during rest | I_{DD_REST} | $V_{DD} = 3.3V$, LEDs off, NCELLS_SER = 10, EXE_TIME = 4s, WEXE_TIME_REST = 4, with I ² C traffic | | 135 | | μA |
| Total disabled current | I_{DD_DIS} | $V_{DD} = 3.3V$ | | 6 | | μA |
| Power-On Reset (POR) | | | | | | |
| Release threshold of POR | V_{ROT} | V_{DD} rising | 1.66 | 1.79 | 1.9 | V |
| nRST PIN | | | | | | |
| nRST pin threshold voltage | V_{RST} | | 0.2 x V_{DD} | | 0.9 x V_{DD} | V |
| Minimum pulse width on the nRST pin | t_{RST} | $V_{DD} = 3.3V$ | | 700 | | ns |
| Time-out after reset | t_{TOUT} | | | 64 | 128 | ms |
| CE | | | | | | |
| Low input voltage | V_{IL_CE} | $I_{OL} = 5mA$ | -0.3 | | $V_{DD} \times 0.35$ | V |
| High input voltage | V_{IH_CE} | | $V_{DD} \times 0.65V$ | | $V_{DD} + 0.3V$ | V |
| Low leakage current | $I_{CE_LKG_LOW}$ | $V_{CE} = 3.3V$ | | | 3 | μA |
| High leakage current | $I_{CE_LKG_HIGH}$ | $V_{CE} = 3.3V$ | | | 3 | μA |
| IRQ | | | | | | |
| Low output voltage | V_{IRQL} | Sink = 4mA | | | 0.4 | V |
| High output voltage | V_{IRQH} | Source = 4mA | $V_{DD} - 0.4$ | | | V |
| PB | | | | | | |
| Low input voltage | V_{PBL} | | -0.3 | | $V_{DD} \times 0.35$ | V |
| High input voltage | V_{PBH} | | $V_{DD} \times 0.65V$ | | $V_{DD} + 0.3V$ | V |

ELECTRICAL CHARACTERISTICS (continued)
 $V_{DD} = 3.3V$, $T_A = 25^{\circ}C$, unless otherwise noted.

| Parameters | Symbol | Condition | Min | Typ | Max | Units |
|--|---------------|----------------------------|---------------------|-----|---------------------|---------|
| LEDs | | | | | | |
| High-level output voltage | V_{OH_LED} | Output current = 4mA | $V_{DD} - 0.4V$ | | | V |
| High-level output current | I_{OH_LED} | $V_{LEDx} = V_{DD} - 0.4V$ | 4 | | | mA |
| I²C DC Characteristics | | | | | | |
| High input voltage | V_{IH} | SCL, SDA | $0.7 \times V_{DD}$ | | $V_{DD} + 0.3$ | V |
| Low input voltage | V_{IL} | SCL, SDA | -0.5 | | $0.3 \times V_{DD}$ | V |
| Low output voltage | V_{OL} | SDA 3mA sink current | | | 0.4 | V |
| I²C Timing Characteristics | | | | | | |
| Spikes suppressed by input filter | t_{SP} | | 0 | | 50 | ns |
| Operating frequency range | f_{SCL} | | 10 | | 400 | kHz |
| SCL clock low period | t_{LOW} | | 1.125 | | | μs |
| SCL clock high period | t_{HIGH} | | 1.125 | | | μs |
| SCL and SDA falling time | t_{FALL} | | | | 0.34 | μs |
| SCL and SDA rising time | t_{RISE} | | | | 0.34 | μs |
| Data hold time | t_{HD_DAT} | | 0 | | | ns |
| Data set-up time | t_{SU_DAT} | | 125 | | | ns |
| Data valid time | t_{V_DAT} | | 475 | | | ns |
| Set-up time for a repeated start condition | t_{SU_STA} | | 125 | | | ns |
| Hold time for a repeated start condition | t_{HD_STA} | | 0 | | | ns |
| Set-up time for a stop condition | t_{SU_STO} | | 125 | | | ns |

Note:

- 5) Validated on a 10S1P Samsung INR18650 25R pack (see the Typical Performance Characteristics section on page 8 for more details). Similar results can be achieved with other cell/pack types after characterization.

TYPICAL PERFORMANCE CHARACTERISTICS

The fuel gauge performance depends on multiple factors, such as the accuracy of the measurements, the correctness of the configuration, and the fidelity of the cell mathematical model. This means that fuel gauge performance may vary depending on the battery operating conditions.

This section shows examples of the MPS MPF42790 fuel gauge pack’s SOC performance when paired with MPS’s MP279x AFE family.

Constant-Current/Constant-Voltage (CC/CV) Charge and Dynamic Discharge Cycle

The next scenarios consist of charging a 10S1P ⁽⁶⁾ battery using the typical CC/CV method, followed by a highly dynamic discharge at different ambient temperatures. The charge constant current rate is 1C, while the charge termination current in this example is 0.1C. The highly dynamic discharge corresponds to a typical e-bike’s current profile, with an average current of 1C and maximum peak currents up to 2.8C. Figure 3 shows the current profile of the complete cycle at 25°C.

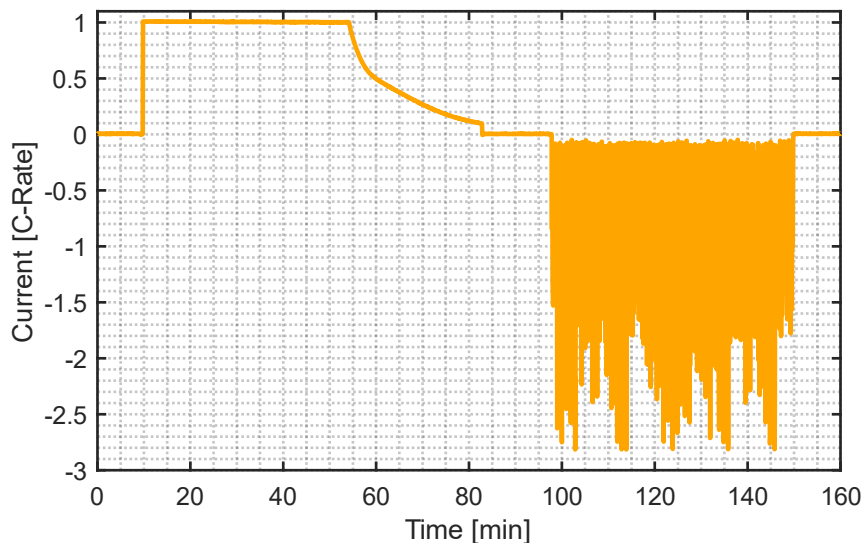


Figure 3: CC/CV Charge and Dynamic Discharge Current Profile

Figure 4 on page 10 shows the MPF42790’s performance for the CC/CV charge and dynamic discharge cycle at an ambient temperature of 25°C. During charge, the root-mean-square ⁽⁷⁾ and maximum pack SOC error are 0.70% and 1.20%, respectively. During discharge, the root-mean-squared and pack SOC error are 1.18% and 2.93%, respectively.

Note:

6) 10S1P refers to the battery configuration. There are 10 groups of 1 parallel cell connected in series.

7) The RMS error is equal to $\sqrt{\frac{\sum_{n=1}^N (\theta_n - \hat{\theta}_n)^2}{N}}$, where θ is the actual SOC, $\hat{\theta}$ is the estimated SOC, and N is the number of samples.

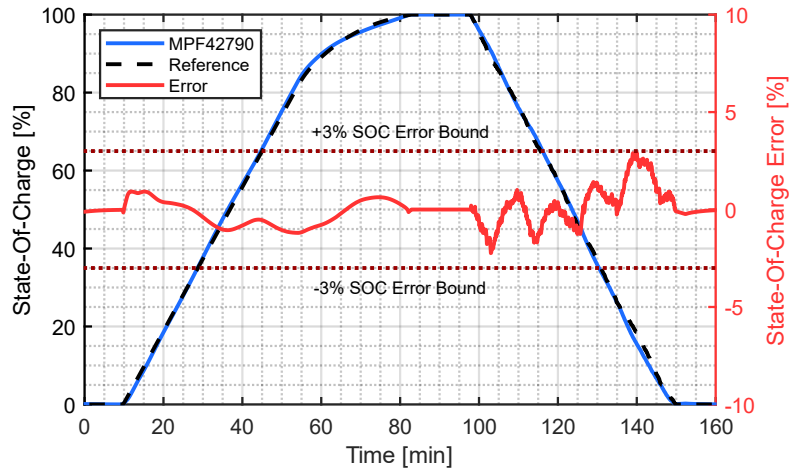


Figure 4: MPF42790 Performance for a CC/CV Charge and Dynamic Discharge (Ambient Temperature = 25°C)

Figure 5 shows the MPF42790’s performance for the CC/CV charge and dynamic discharge cycle at 40°C ambient temperature. During charge, the root-mean-squared and maximum pack SOC error are 1.31% and 2.49%, respectively. During discharge, the root-mean-squared and pack SOC error are 1.36% and 2.96%, respectively.

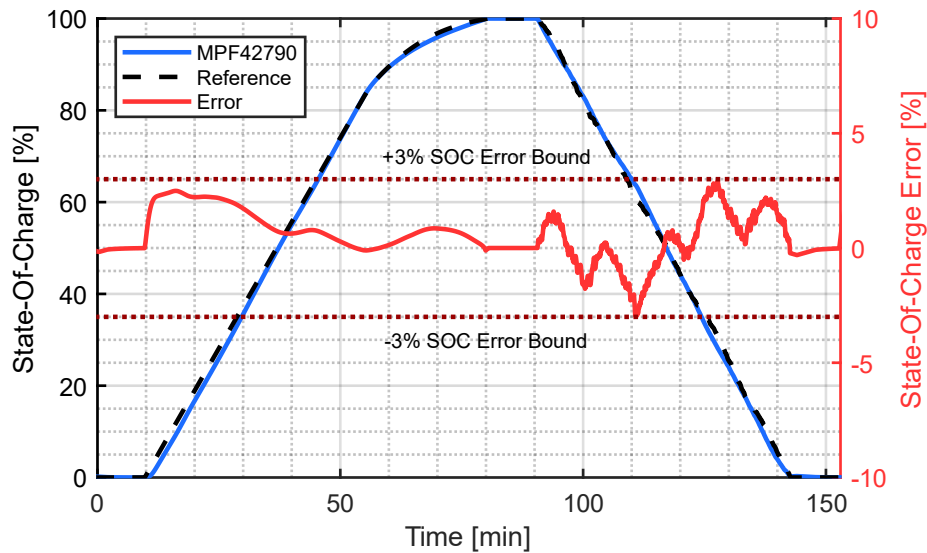


Figure 6: MPF42790 Performance for a CC/CV Charge and Dynamic Discharge (Ambient Temperature = 40°C)

Performance Summary

This section provides a summary of the MPF42790’s real-world performance. **Error! Reference source not found.** shows a summary of the pack SOC performance metrics for a 10S1P battery.

Table 2: MPF42790 SOC Root-Mean-Squared (and Maximum) Error

| Test Case | 25°C | 40°C |
|-------------------|---------------|---------------|
| CC/CV charge | 0.70% (1.20%) | 1.31% (2.49%) |
| Dynamic discharge | 1.18% (2.93%) | 1.36% (2.96%) |

OPERATION

The MPF42790 relies on a host microcontroller (MCU) to periodically update the fuel gauge's required inputs. This includes the cell voltages (via the VRDG_CELLxx registers), current (via the IRDG_CELLxx registers), and temperature (via the TRDG_TSx registers).

The host MCU is responsible for synchronizing the input data to the fuel gauge's execution time (EXE_TIME). The host MCU sends the EXE_CMD command to trigger the fuel gauge execution and waits until the iteration is completed (FG_EXE_FLAG = 0). Then the host MCU can read back the fuel gauge's output registers, such as the pack state-of-charge (SOC). The interrupt request (IRQ) pin can be configured to notify several events, such as a completed iteration or other conditions.

Operating Modes

Active

In active mode, the fuel gauge is either updating the battery internal states or communicating via the I²C bus.

Standby

In standby mode, the fuel gauge is idle, which means that all triggered updates have been completed and the fuel gauge is waiting for activity on the SDA line to transition to active mode. See the I²C Communication Interface section on page 12 for more details.

Disabled

In disabled mode, the chip enable (CE) pin is low. I²C communication is not available, and the fuel gauge achieves minimal current consumption but still retains all internal state variables in its memory.

Configuration and Data Exchange

Configuration Mode

In the configuration mode, the MPF42790 is set to receive configuration parameters. However,

the device will stop operating. This mode is enabled by sending the CONFIG_MODE_CMD and can be confirmed by reading the CONFIG_MODE_FLAG register. After the fuel gauge configurations are successfully updated, the host system must send the CONFIG_EXIT_CMD command to save the configuration into non-volatile memory (NVM). This ensures that the configuration remains after a hardware reset (cycling the device's power or cycling power on the nRST pin).

Editing Mode

As an alternative to the configuration mode, the edit mode allows for partial configuration changes. In this mode, the fuel gauge settings can be updated, but not changes to the fuel gauge cell model. Configuration mode is required to change the fuel gauge cell model. In this mode, the fuel gauge operation does not stop, which allows for on-the-fly configurations. This mode is enabled by sending the EDIT_CONFIG_CMD command and can be confirmed by reading the EDIT_SETTINGS_FLAG. The host system can exit this mode by sending the END_EDIT_CONFIG_CMD or CONFIG_EXIT_CMD command, which store the updated configuration in the MPF42790's NVM.

LED Control

The MPF42790 can drive five external LEDs that report the pack SOC. There are 2 methods for controlling the LEDs, direct control or manual control, which is configured using the LEDS_ON_MAN register.

Note that the LED settings can be written at any time during active or standby operating modes without having to enter configuration or edit modes.

Direct LED Control

With direct LED control, the fuel gauge directly controls the LEDs based on the pack SOC (see

Table 1 on page 12). When the PB pin is pulled low (if the LEDS_ON_BTN register is enabled), or when the battery pack is charging (if LEDS_ON_CHG is enabled), the LEDs display pack SOC. The deglitch time between charge and discharge can be configured via the

LEDS_TRANS register. The LED turn-on time can be configured via the LEDS_ON_TIME register.

Table 1: LED State-of-Charge Indicator Map

| Pack SOC | LED Status | | | | |
|-----------------|------------|-----|-----|-----|-----|
| | 1 | 2 | 4 | 4 | 5 |
| SOC > 90% | On | On | On | On | On |
| 90% ≥ SOC > 70% | On | On | On | On | Off |
| 70% ≥ SOC > 50% | On | On | On | Off | Off |
| 50% ≥ SOC > 30% | On | On | Off | Off | Off |
| 30% ≥ SOC > 10% | On | Off | Off | Off | Off |
| 10% ≥ SOC | Off | Off | Off | Off | Off |

Manual LED Control

With manual LED control, the host manually controls each LED via the corresponding LEDx_ON register.

I²C Communication Interface

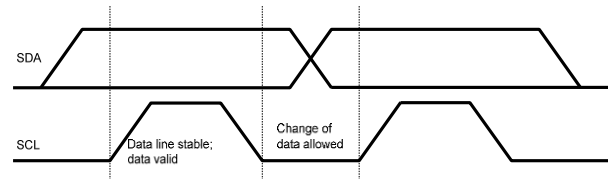
The MPF42790 has an I²C channel that communicates with the host MCU which extracts the data from the AFE.

The I²C is a bidirectional, two-wire serial interface. Only two bus lines are required: a serial data line (SDA) and a serial clock line (SCL). The device can be considered a master or a slave when performing data transfers. A device that initiates a data transfer on the bus and generates the clock signals to permit the transfer is considered a master. Any device that the master addresses is considered a slave.

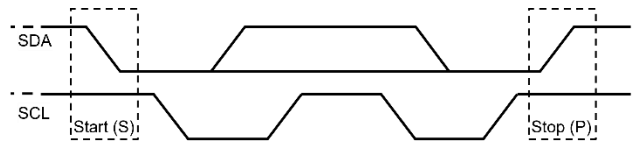
Both MPF42790 I²C interfaces operate as slave devices with a configurable address (0x08 by default). They receive control inputs from the master device and ignore general call addresses.

The I²C interface supports both standard mode (up to 100kbits), and fast mode (up to 400kbits). The SDA and SCL lines are bidirectional with open drain outputs that should be connected to the positive supply voltage via a current source or pull-up resistor. When the bus is free, both lines are high.

The data on the SDA line must be stable during the high period of the clock. the high or low state of the data line can only change when the clock signal on the SCL line is low. Note that a single clock pulse is generated for each data bit transferred (see Figure 7).


Figure 7: Bit Transfer on the I²C Bus

All transactions must begin with a start (S) command and can be terminated with a stop (P) command. Start and stop commands are always generated by the master. A start command is defined by a high-to-low transition on the SDA line while SCL is high (see Figure 8). A stop command is defined by a low-to-high transition on the SDA line while the SCL is high (see Figure 8). The bus is considered busy after a start command and free after a stop command.


Figure 8: I²C Start and Stop Commands

Every byte on the SDA line must be 8 bits long and must be followed by an acknowledge (ACK) bit. Note that data is transferred with the most significant bit (MSB) first.

A slave cannot receive or transmit a complete byte of data while performing other tasks, but it can hold the SCL line low to force the master into a wait state (clock stretching). Then, when the slave is ready, data transfer continues, and the clock line (SCL) is released.

The acknowledgement takes place after every byte and allows the receiver to signal to the transmitter that the byte was successfully received. All clock pulses are generated by the master, including the 9th clock pulse (ACK).

The transmitter releases the SDA line during the ACK clock pulse so that the receiver pulls the SDA line low. If the SDA line remains high during the acknowledge clock pulse, it is not acknowledged (NACK). Then the master can generate a stop command to abort the transfer, or it can generate a repeated start (Sr) command to start a new transfer.

After the start command, a 7-bit slave address is sent, followed by the read/write (R/W) bit. A 0

represents a write transmission (W), while a 1 indicates a read request (R). Figure 9 shows a complete data transfer.

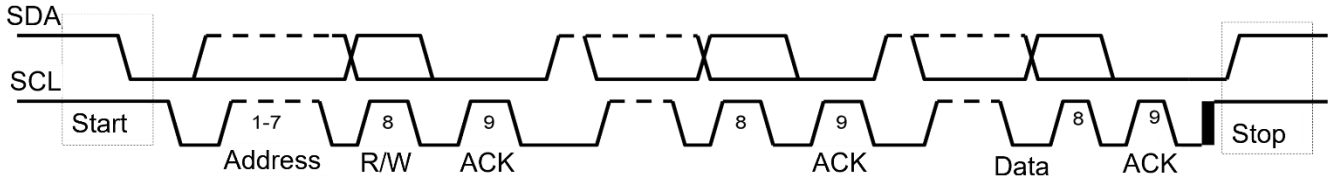


Figure 9: Complete Data Transfer

Active Mode

The MPF42790 must be in active mode to communicate (see Operating Modes for additional detail). To transition to active mode, the SDA line must be pulled low. There is a 20ms timeout before starting operation, or the device transitions back to standby mode. After the operations are completed (e.g. I²C communication or fuel gauge updates) there is a 5ms timeout.

A low 5ms pulse on the SDA line is recommended to transition to active mode. To verify that the MPF42790 is ready for communication, an I²C header with the start command, the device address, and the R/W bit can be sent. The device responds with an ACK signal if it is ready for communication.

Protocol Layer

The MPF42790 uses a protocol where 2 bytes are used for the register and command addresses. A length field is also provided to declare the number of bytes in each read or write transaction. The maximum allowed transaction length is 82 bytes of data (not including the register address, length byte, or CRC bytes).

Cyclic redundancy check (CRC) can be used to ensure the transaction’s integrity. When enabled, the last 4 bytes of the transaction correspond to the CRC code. Figure 10 and Figure 11 show an example of read and write transactions with CRC, respectively.

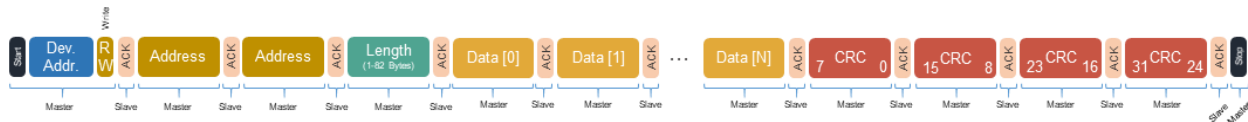


Figure 10: I²C Read Transaction with CRC

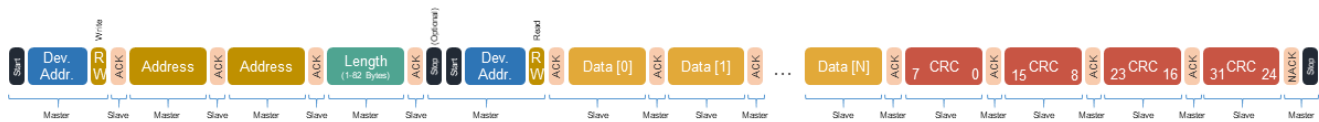


Figure 11: I²C Write Transaction with CRC

If CRC is disabled, a write transaction can optionally include the correct CRC bytes to be accepted. During a read, the MPF42790 does not include CRC, so the NACK signal and stop command from the master come after DATA[x].

I²C operations using any address outside of the register map are invalid, even if they are partially valid addresses. An invalid read operation returns a 0 as the data and correct CRC (if

enabled), and an invalid write operation is ignored.

Cyclic Redundancy Check (CRC)

CRC spans the register address, length, and data payload. It is generated in blocks of 4 bytes. If the number of bytes is not a multiple of 4, the block is padded with 0x00. The algorithm to generate the CRC code is listed on page 14.

```

unsigned long crc32 (unsigned short Reg_Address, unsigned
char len, unsigned char *data){
    short i;
    unsigned long crc = 0xffffffff;
    unsigned char dataTemp[4];

    for (i=-1; i<len; i++) {
        if(i==-1) {
            dataTemp[0]=len;
            dataTemp[1]=Reg_Address&0x00FF;
            dataTemp[2]=(Reg_Address&0xFF00)>>8;
            dataTemp[3]=0;
        }
        else dataTemp[i%4]=data[i];
        if((i%4)==3 || i == len-1 || i == -1) {
            for (char j=0; j<4; j++) {
                crc ^= dataTemp[3-j] << 24;
                for (char k = 0; k < 8; ++k) {
                    if ((crc & 0x80000000) != 0)
                        crc = (crc << 1) ^ 0x04C11DB7;
                    else
                        crc <<= 1;
                }
            }
            dataTemp[0]=0;
            dataTemp[1]=0;
            dataTemp[2]=0;
            dataTemp[3]=0;
        }
    }
    return crc;
}

```

To disable CRC and configure the slave address as 0x08, send the following message: {0x00, 0x41, 0x01, 0x08, 0x94, 0xA0, 0xDE, 0xDD} (Address = 0x4100, Length = 0x01, Data = 0x08, CRC = 0xDDDE A094).

I²C REGISTER MAP

Fuel Gauge Input

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|---------------------|---|----------------------|------|--|-----------------------|------------------------|
| VRDG_CELL1, 0x0000 | Inputs cell 1's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL2, 0x0002 | Inputs cell 2's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL3, 0x0004 | Inputs cell 3's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL4, 0x0006 | Inputs cell 4's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL5, 0x0008 | Inputs cell 5's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL6, 0x000A | Inputs cell 6's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL7, 0x000C | Inputs cell 7's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL8, 0x000E | Inputs cell 8's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL9, 0x0010 | Inputs cell 9's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL10, 0x0012 | Inputs cell 10's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL11, 0x0014 | Inputs cell 11's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL12, 0x0016 | Inputs cell 12's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL13, 0x0018 | Inputs cell 13's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL14, 0x001A | Inputs cell 14's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL15, 0x001C | Inputs cell 15's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_CELL16, 0x001E | Inputs cell 16's voltage reading. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_PACK, 0x0020 | Inputs the pack's voltage reading if VRDG_PACK_EN is enabled. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mV | 0mV | 0mV to 131070mV |
| IRDG_CELL1, 0x0022 | Inputs cell 1's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |

| | | | | | | |
|------------------------|---|------------|-----|---|-----|--|
| IRDG_CELL2, 0x0026 | Inputs cell 2's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL3, 0x002A | Inputs cell 3's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL4, 0x002E | Inputs cell 4's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL5, 0x0032 | Inputs cell 5's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL6, 0x0036 | Inputs cell 6's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL7, 0x003A | Inputs cell 7's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL8, 0x003E | Inputs cell 8's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL9, 0x0042 | Inputs cell 9's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL10, 0x0046 | Inputs cell 10's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL11, 0x004A | Inputs cell 11's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL12, 0x004E | Inputs cell 12's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL13, 0x0052 | Inputs cell 13's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL14, 0x0056 | Inputs cell 14's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL15, 0x005A | Inputs cell 15's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_CELL16, 0x005E | Inputs cell 16's current reading if IRDG_CELLS_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| IRDG_PACK, 0x0062 | Inputs the pack's current reading if IRDG_PACK_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 1mA | 0mA | -524288mA to +524287mA |
| CCRDG, 0x0066 | Inputs the Coulomb count reading if CCRDG_EN is enabled. | 32, [31:0] | R/W | 32-bit signed, 2-comp integer LSB = 0.0005mA | 0mA | -1073741.832mA to +1073741.832mA |
| TRDG_TS1, 0x006A | Inputs temperature sensor 1's reading. | 16, [15:0] | R/W | 16-bit signed, 2-comp integer LSB = 0.01°C | 0°C | -273.15°C to +327.67°C |
| TRDG_TS2, 0x006C | Inputs temperature sensor 2's reading. | 16, [15:0] | R/W | 16-bit signed, 2-comp integer LSB = 0.01°C | 0°C | -273.15°C to +327.67°C |
| TRDG_TS3, 0x006E | Inputs temperature sensor 3's reading. | 16, [15:0] | R/W | 16-bit signed, 2-comp integer LSB = 0.01°C | 0°C | -273.15°C to +327.67°C |

| | | | | | | |
|-----------------------|---|------------|-----|---|-----|---------------------------|
| TRDG_TS4, 0x0070 | Inputs temperature sensor 4's reading. | 16, [15:0] | R/W | 16-bit signed, 2-comp integer LSB = 0.01°C | 0°C | -273.15°C to +327.67°C |
| TRDG_DIE, 0x0072 | Inputs the die temperature reading. | 16, [15:0] | R/W | 16-bit signed, 2-comp integer LSB = 0.01°C | 0°C | -273.15°C to +327.67°C |
| BAL_CELL1, 0x0074 | Indicates cell 1's balancing status. 1: Active 0: Inactive | 1, [0] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL2, 0x0074 | Indicates cell 2's balancing status. 1: Active 0: Inactive | 1, [1] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL3, 0x0074 | Indicates cell 3's balancing status. 1: Active 0: Inactive | 1, [2] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL4, 0x0074 | Indicates cell 4's balancing status. 1: Active 0: Inactive | 1, [3] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL5, 0x0074 | Indicates cell 5's balancing status. 1: Active 0: Inactive | 1, [4] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL6, 0x0074 | Indicates cell 6's balancing status. 1: Active 0: Inactive | 1, [5] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL7, 0x0074 | Indicates cell 7's balancing status. 1: Active 0: Inactive | 1, [6] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL8, 0x0074 | Indicates cell 8's balancing status. 1: Active 0: Inactive | 1, [7] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL9, 0x0074 | Indicates cell 9's balancing status. 1: Active 0: Inactive | 1, [8] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL10, 0x0074 | Indicates cell 10's balancing status. 1: Active 0: Inactive | 1, [9] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL11, 0x0074 | Indicates cell 11's balancing status. 1: Active 0: Inactive | 1, [10] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL12, 0x0074 | Indicates cell 12's balancing status. 1: Active 0: Inactive | 1, [11] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL13, 0x0074 | Indicates cell 13's balancing status. 1: Active 0: Inactive | 1, [12] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL14, 0x0074 | Indicates cell 14's balancing status. 1: Active 0: Inactive | 1, [13] | R/W | Boolean true/false | 0 | 0 to 1 |
| BAL_CELL15, 0x0074 | Indicates cell 15's balancing status. 1: Active 0: Inactive | 1, [14] | R/W | Boolean true/false | 0 | 0 to 1 |

| | | | | | | |
|-------------------------|---|---------|-----|--------------------|---|--------|
| BAL_CELL16, 0x0074 | Indicates cell 16's balancing status. 1: Active 0: Inactive | 1, [15] | R/W | Boolean true/false | 0 | 0 to 1 |
| HARD_RST, 0x0076 | Resets the fuel gauge and lifetime log if set to 1 (i.e. FG_RST = LOG_RST = 1). This is a self-clearing register that returns to 0 automatically. | 1, [0] | R/W | Boolean true/false | 0 | 0 to 1 |
| FG_RST, 0x0076 | Resets the fuel gauge if set to 1 (i.e. STATUS_RST = SOH_RST = SOC_RST = FULL_RST = EMTY_RST = IR_RST = ETAC_RST = IDIS_AVG_RST = IDIS_END_RST = ICHG_END_RST = ICHG_CC_RST = VCHG_CV_RST = 1). This is a self-clearing register that returns to 0 automatically. | 1, [1] | R/W | Boolean true/false | 0 | 0 to 1 |
| LOG_RST, 0x0076 | Resets the lifetime log. 1: Reset (self-clearing) 0: No action | 1, [2] | R/W | Boolean true/false | 0 | 0 to 1 |
| WKUP, 0x0076 | Wakes up the fuel gauge. 1: Wake up (self-clearing) 0: No action | 1, [3] | R/W | Boolean true/false | 0 | 0 to 1 |
| STATUS_RST, 0x0076 | Resets the pack status. 1: Reset (self-clearing) 0: No action | 1, [4] | R/W | Boolean true/false | 0 | 0 to 1 |
| SOH_RST, 0x0076 | Resets the SOH. 1: Reset (self-clearing) 0: No action | 1, [5] | R/W | Boolean true/false | 0 | 0 to 1 |
| SOC_RST, 0x0076 | Resets the SOC. 1: Reset (self-clearing) 0: No action | 1, [7] | R/W | Boolean true/false | 0 | 0 to 1 |
| FULL_RST, 0x0076 | Resets the pack full. 1: Reset (self-clearing) 0: No action | 1, [8] | R/W | Boolean true/false | 0 | 0 to 1 |
| EMTY_RST, 0x0076 | Resets the pack empty. 1: Reset (self-clearing) 0: No action | 1, [9] | R/W | Boolean true/false | 0 | 0 to 1 |
| IDIS_AVG_RST, 0x0076 | Resets the average discharge current. 1: Reset (self-clearing) 0: No action | 1, [10] | R/W | Boolean true/false | 0 | 0 to 1 |
| IDIS_END_RST, 0x0076 | Resets the discharge termination current. 1: Reset (self-clearing) 0: No action | 1, [11] | R/W | Boolean true/false | 0 | 0 to 1 |
| ICHG_END_RST, 0x0076 | Resets the charge termination current. 1: Reset (self-clearing) 0: No action | 1, [12] | R/W | Boolean true/false | 0 | 0 to 1 |
| ICHG_CC_RST, 0x0076 | Resets the charge CC. 1: Reset (self-clearing) 0: No action | 1, [13] | R/W | Boolean true/false | 0 | 0 to 1 |
| VCHG_CV_RST, 0x0076 | Resets the charge CV. 1: Reset (self-clearing) 0: No action | 1, [14] | R/W | Boolean true/false | 0 | 0 to 1 |

| | | | | | | |
|---------------------|---|--------|-----|--|-------------|--------|
| PCHG_SHW, 0x0078 | Shows the maximum charge power in PCHG, the limiting cell ID in PCHG_ID, and the limiting factor in PCHG_LIM (PCHG_SHW_EN by default). 1: Show 0: Do not show | 1, [0] | R/W | Control Enable = 1 Disable = 0 | PCHG_SHW_EN | 0 to 1 |
| PDIS_SHW, 0x0078 | Shows the maximum discharge power (PDIS_SHW_EN by default). 1: Show 0: Do not show | 1, [1] | R/W | Control Enable = 1 Disable = 0 | PDIS_SHW_EN | 0 to 1 |

Fuel Gauge Backup

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|---------------------------|---|----------------------|------|---|-----------------------|------------|
| SOH_CELL1_BKUP, 0x0200 | <p>These bits are the backup register for cell 1's SOH. This address is a R/W alias for the SOH_CELL1 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | 32, [31:0] | R/W | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL2_BKUP, 0x0204 | <p>These bits are the backup register for cell 2's SOH. This address is a R/W alias for the SOH_CELL2 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | 32, [31:0] | R/W | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL3_BKUP, 0x0208 | <p>These bits are the backup register for cell 3's SOH. This address is a R/W alias for the SOH_CELL3 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | 32, [31:0] | R/W | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL4_BKUP, 0x020C | <p>These bits are the backup register for cell 4's SOH. This address is a R/W alias for the SOH_CELL4 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | 32, [31:0] | R/W | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |

| | | | | | | |
|-----------------------------------|---|-------------------|------------|---|-----------|-------------------|
| <p>SOH_CELL5_BKUP, 0x0210</p> | <p>These bits are the backup register for cell 5's SOH. This address is a R/W alias for the SOH_CELL5 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL6_BKUP, 0x0214</p> | <p>These bits are the backup register for cell 6's SOH. This address is a R/W alias for the SOH_CELL6 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL7_BKUP, 0x0218</p> | <p>These bits are the backup register for cell 7's SOH. This address is a R/W alias for the SOH_CELL7 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL8_BKUP, 0x021C</p> | <p>These bits are the backup register for cell 8's SOH. This address is a R/W alias for the SOH_CELL8 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL9_BKUP, 0x0220</p> | <p>These bits are the backup register for cell 9's SOH. This address is a R/W alias for the SOH_CELL9 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |

| | | | | | | |
|---------------------------------|--|-------------------|------------|---|-----------|-------------------|
| <p>SOH_CELL10_ BKUP, 0x0224</p> | <p>These bits are the backup register for cell 10's SOH. This address is a R/W alias for the SOH_CELL10 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL11_ BKUP, 0x0228</p> | <p>These bits are the backup register for cell 11's SOH. This address is a R/W alias for the SOH_CELL11 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35..</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL12_ BKUP, 0x022C</p> | <p>These bits are the backup register for cell 12's SOH. This address is a R/W alias for the SOH_CELL12 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL13_ BKUP, 0x0230</p> | <p>These bits are the backup register for cell 13's SOH. This address is a R/W alias for the SOH_CELL13 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL14_ BKUP, 0x0234</p> | <p>These bits are the backup register for cell 14's SOH. This address is a R/W alias for the SOH_CELL14 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |

| | | | | | | |
|------------------------------------|---|-------------------|------------|---|-----------|------------------------|
| <p>SOH_CELL15_BKUP, 0x0238</p> | <p>These bits are the backup register for cell 15's SOH. This address is a R/W alias for the SOH_CELL15 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>SOH_CELL16_BKUP, 0x023C</p> | <p>These bits are the backup register for cell 16's SOH. This address is a R/W alias for the SOH_CELL16 register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this or any other of the cell SOH backup registers prepares the fuel gauge to use the values written in these registers as a starting point for the SOH learning.</p> | <p>32, [31:0]</p> | <p>R/W</p> | <p>32-bit unsigned integer LSB = 0.0000001%</p> | <p>0%</p> | <p>0% to 255%</p> |
| <p>IDIS_AVG_BKUP, 0x0280</p> | <p>These bits are the backup register for the average discharge current. This address is a R/W alias for the IDIS_AVG register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this backup register prepares the fuel gauge to use the values written in this register as a starting point for this learning.</p> | <p>16, [15:0]</p> | <p>R/W</p> | <p>16-bit unsigned integer LSB = 2mA</p> | <p>-</p> | <p>0mA to 131070mA</p> |
| <p>IDIS_END_BKUP, 0x0290</p> | <p>These bits are the backup register for the discharge end current. This address is a R/W alias for the IDIS_END register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this backup register prepares the fuel gauge to use the values written in this register as a starting point for this learning.</p> | <p>16, [15:0]</p> | <p>R/W</p> | <p>16-bit unsigned integer LSB = 2mA</p> | <p>-</p> | <p>0mA to 131070mA</p> |
| <p>ICHG_CC_BKUP, 0x02A0</p> | <p>These bits are the backup register for the charge CC current. This address is a R/W alias for the ICHG_CC register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this backup register prepares the fuel gauge to use the values written in this register as a starting point for this learning.</p> | <p>16, [15:0]</p> | <p>R/W</p> | <p>16-bit unsigned integer LSB = 2mA</p> | <p>-</p> | <p>0mA to 131070mA</p> |

| | | | | | | |
|-------------------------------|---|------------|------------|--------------------------------------|---|-----------------|
| <p>ICHG_END_BKUP, 0x02B0</p> | <p>These bits are the backup register for the charge termination current. This address is a R/W alias for the ICHG_END register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this backup register prepares the fuel gauge to use the values written in this register as a starting point for this learning.</p> | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070mA |
| <p>VCHG_CV_BKUP, 0x02C0</p> | <p>These bits are the backup register for the charge CV voltage. This address is a R/W alias for the VCHG_CV register.</p> <p>The fuel gauge must be in learning backup mode for this register to be written. For more details, see the FG_BKUP_ENABLE register on page 23 and the FG_BKUP_FLAG register on page 35.</p> <p>Writing to this backup register prepares the fuel gauge to use the values written in this register as a starting point for this learning.</p> | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070mA |
| <p>FG_BKUP_ENABLE, 0x0300</p> | <p>Enables learning backup mode. See the FG_BKUP_FLAG register on page 35 for more details.</p> | 1, [0] | Write-only | Control Enable = 1 Disable = 0 | 0 | 0 to 1 |

Fuel Gauge Output

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|--------------------------|------------------------------|----------------------|-----------|--------------------------------------|-----------------------|------------|
| EMPTY_SOC_CELL1, 0x0564 | Returns cell 1's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL2, 0x0565 | Returns cell 2's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL3, 0x0566 | Returns cell 3's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL4, 0x0567 | Returns cell 4's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL5, 0x0568 | Returns cell 5's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL6, 0x0569 | Returns cell 6's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL7, 0x056A | Returns cell 7's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL8, 0x056B | Returns cell 8's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL9, 0x056C | Returns cell 9's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL10, 0x056D | Returns cell 10's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL11, 0x056E | Returns cell 11's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |

| | | | | | | |
|--------------------------|---|------------|-----------|---|-------------|-------------------|
| EMPTY_SOC_CELL12, 0x056F | Returns cell 12's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL13, 0x0570 | Returns cell 13's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL14, 0x0571 | Returns cell 14's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL15, 0x0572 | Returns cell 15's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL16, 0x0573 | Returns cell 16's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_ID, 0x0574 | Returns the cell ID for the cell that is empty limiting the pack. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 0 (cell ID) | 1 to 16 (cell ID) |
| EMPTY_RUNTIME, 0x0575 | Returns the remaining time for the pack to become empty. Smoother changes for the remaining time estimate can be achieved by increasing EMPTY_RUNTIME_FILTER with the tradeoff of an increased phase delay. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1s | 0s | 0s to 65535s |
| EMPTY_LIM, 0x0577 | Outputs the pack's empty limiting factor. 0: Cell 1: Pack | 8, [7:0] | Read-only | Non-standard LSB = 1 | 0 | 0 to 1 |
| FULL_SOC_CELL1, 0x057C | Returns cell's 1 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL2, 0x057D | Returns cell's 2 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL3, 0x057E | Returns cell's 3 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL4, 0x057F | Returns cell's 4 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL5, 0x0580 | Returns cell's 5 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL6, 0x0581 | Returns cell's 6 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL7, 0x0582 | Returns cell's 7 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL8, 0x0583 | Returns cell's 8 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL9, 0x0584 | Returns cell's 9 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL10, 0x0585 | Returns cell's 10 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |

| | | | | | | |
|-------------------------|--|------------|-----------|---|------|-------------------|
| FULL_SOC_CELL11, 0x0586 | Returns cell's 11 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL12, 0x0587 | Returns cell's 12 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL13, 0x0588 | Returns cell's 13 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL14, 0x0589 | Returns cell's 14 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL15, 0x058A | Returns cell's 15 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL16, 0x058B | Returns cell's 16 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_ID, 0x058C | Returns the cell ID for the cell that is full limiting the pack. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 0 | 1 to 16 (cell ID) |
| FULL_RUNTIME, 0x058D | Returns the remaining time to reach a full pack. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1s | 0 | 0s to 65535s |
| FULL_LIM, 0x058F | Returns the pack's full limiting factor. 0: Cell 1: Pack | 1, [0] | Read-only | Non-standard | 0 | 0 to 1 |

Fuel Gauge Output

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|-------------------------|-----------------------------|----------------------|-----------|--------------------------------------|-----------------------|------------|
| EMPTY_SOC_CELL1, 0x0564 | Returns cell 1's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL2, 0x0565 | Returns cell 2's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL3, 0x0566 | Returns cell 3's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL4, 0x0567 | Returns cell 4's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL5, 0x0568 | Returns cell 5's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL6, 0x0569 | Returns cell 6's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL7, 0x056A | Returns cell 7's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL8, 0x056B | Returns cell 8's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL9, 0x056C | Returns cell 9's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |

| | | | | | | |
|--------------------------|---|------------|-----------|---|-------------|-------------------|
| EMPTY_SOC_CELL10, 0x056D | Returns cell 10's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL11, 0x056E | Returns cell 11's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL12, 0x056F | Returns cell 12's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL13, 0x0570 | Returns cell 13's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL14, 0x0571 | Returns cell 14's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL15, 0x0572 | Returns cell 15's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_SOC_CELL16, 0x0573 | Returns cell 16's empty SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| EMPTY_ID, 0x0574 | Returns the cell ID for the cell that is empty limiting the pack. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 0 (cell ID) | 1 to 16 (cell ID) |
| EMPTY_RUNTIME, 0x0575 | Returns the remaining time for the pack to become empty. Smoother changes for the remaining time estimate can be achieved by increasing EMPTY_RUNTIME_FILTER with the tradeoff of an increased phase delay. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1s | 0s | 0s to 65535s |
| EMPTY_LIM, 0x0577 | Outputs the pack's empty limiting factor. 0: Cell 1: Pack | 8, [7:0] | Read-only | Non-standard LSB = 1 | 0 | 0 to 1 |
| FULL_SOC_CELL1, 0x057C | Returns cell's 1 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL2, 0x057D | Returns cell's 2 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL3, 0x057E | Returns cell's 3 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL4, 0x057F | Returns cell's 4 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL5, 0x0580 | Returns cell's 5 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL6, 0x0581 | Returns cell's 6 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL7, 0x0582 | Returns cell's 7 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL8, 0x0583 | Returns cell's 8 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4 % | 100% | 0% to 100% |
| FULL_SOC_CELL9, 0x0584 | Returns cell's 9 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |

| | | | | | | |
|-------------------------|--|------------|-----------|---|-------------|-------------------|
| FULL_SOC_CELL10, 0x0585 | Returns cell's 10 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL11, 0x0586 | Returns cell's 11 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL12, 0x0587 | Returns cell's 12 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL13, 0x0588 | Returns cell's 13 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL14, 0x0589 | Returns cell's 14 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL15, 0x058A | Returns cell's 15 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_SOC_CELL16, 0x058B | Returns cell's 16 full SOC. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 100% | 0% to 100% |
| FULL_ID, 0x058C | Returns the cell ID for the cell that is full limiting the pack. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 0 (cell ID) | 1 to 16 (cell ID) |
| FULL_RUNTIME, 0x058D | Returns the remaining time to reach a full pack. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1s | 0s | 0s to 65535s |
| FULL_LIM, 0x058F | Returns the pack's full limiting factor. 0: Cell 1: Pack | 1, [0] | Read-only | Non-standard LSB = 1 | 0 | 0 to 1 |
| IDIS_AVG, 0x0590 | Returns the learned value for the typical load current, which is drained from the battery during its use (the average discharge current). | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070 mA |
| IDIS_END, 0x0593 | Returns the learned value for the typical discharge termination current. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070 mA |
| ICHG_CC, 0x0597 | Returns the learned charge current of the charger in the CC region, as performed by the algorithm. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070 mA |
| ICHG_END, 0x059B | Returns the learned value for the typical end of the discharge current | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | - | 0mA to 131070 mA |
| VCHG_CV, 0x059F | Returns the learned value for the typical charge CV voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mV | 0mV | 0mV to 131070mV |
| SOC_PACK_UNAVBL, 0x05A3 | Indicates whether the pack has an unavailable SOC due to empty or full conditions, or a mismatch between cells. Reflects the quantity of pack's unusable capacity. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.002 % | 0% | 0% to 100% |
| SOC_PACK, 0x05A5 | Returns the pack's SOC. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.002 % | 0% | 0% to 100% |

| | | | | | | |
|------------------------|--|------------|-----------|---|----|------------|
| SOC_ABS_CELL1, 0x05A8 | Returns cell 1's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL2, 0x05A9 | Returns cell 2's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL3, 0x05AA | Returns cell 3's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL4, 0x05AB | Returns cell 4's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL5, 0x05AC | Returns cell 5's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL6, 0x05AD | Returns cell 6's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL7, 0x05AE | Returns cell 7's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL8, 0x05AF | Returns cell 8's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL9, 0x05B0 | Returns cell 9's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL10, 0x05B1 | Returns cell 10's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL11, 0x05B2 | Returns cell 11's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL12, 0x05B3 | Returns cell 12's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL13, 0x05B4 | Returns cell 13's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL14, 0x05B5 | Returns cell 14's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL15, 0x05B6 | Returns cell 15's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOC_ABS_CELL16, 0x05B7 | Returns cell 16's absolute SOC. Shows as reserved. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.4% | 0% | 0% to 100% |
| SOH_PACK, 0x0608 | Returns the pack's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL1, 0x060C | Returns cell 1's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL2, 0x0610 | Returns cell 2's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL3, 0x0614 | Returns cell 3's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |

| | | | | | | |
|-----------------------|---|------------|-----------|---|-------------|-----------------------|
| SOH_CELL4, 0x0618 | Returns cell 4's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL5, 0x061C | Returns cell 5's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL6, 0x0620 | Returns cell 6's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL7, 0x0624 | Returns cell 7's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL8, 0x0628 | Returns cell 8's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL9, 0x062C | Returns cell 9's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL10, 0x0630 | Returns cell 10's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL11, 0x0634 | Returns cell 11's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL12, 0x0638 | Returns cell 12's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL13, 0x063C | Returns cell 13's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL14, 0x0640 | Returns cell 14's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL15, 0x0644 | Returns cell 15's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| SOH_CELL16, 0x0648 | Returns cell 16's SOH. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.0000001% | 0% | 0% to 255% |
| PRDG, 0x070C | Returns the power state of the battery pack, which is equal to the pack current multiplied by the battery pack voltage. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.1mW | 0mW | 0mW to 429496729.5 mW |
| PDIS, 0x0710 | Returns the maximum possible applicable discharge power. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.1mW | 0mW | 0mW to 429496729.5 mW |
| PDIS_ID, 0x0714 | Returns the cell ID that is most significantly limiting the maximum discharge power estimate. | 5, [4:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 1 (cell ID) | 1 to 16 (cell ID) |
| PDIS_LIM, 0x0715 | Returns the limiting factor in the maximum discharge power estimate. 0: Cell 1: Pack | 1, [0] | Read-only | 8-bit unsigned integer | 0 | 0 to 1 |
| PCHG, 0x0716 | Returns the maximum possible applicable charge power. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 0.1mW | 0mW | 0mW to 429496729.5 mW |
| PCHG_ID, 0x071A | Returns the cell ID that is most significantly limiting the maximum charge power estimate. | 5, [4:0] | Read-only | 8-bit unsigned integer LSB = 1 (cell ID) | 1 (cell ID) | 1 to 16 (cell ID) |



| | | | | | | |
|---------------------|---|------------|-----------|------------------------------------|---|-----------------|
| PCHG_LIM, 0x071B | Returns the limiting factor in the maximum charge power estimate. 0: Cell 1: Pack | 1, [0] | Read-only | 8-bit unsigned integer | 0 | 0 to 1 |
| STATUS, 0x071C | Indicates the pack's current status. 0: Charge 1: Discharge 2: Rest | 2, [1:0] | Read-only | 8-bit unsigned integer LSB = 1 | 0 | 0 to 2 |
| FG_ITER, 0x071D | Counts the number of iterations executed by the fuel gauge. | 32, [31:0] | Read-only | 32-bit unsigned integer LSB = 1 | 0 | 0 to 4294967295 |

Fuel Gauge Lifetime Log

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|----------------------------|---|----------------------|-----------|--|-----------------------|-----------------|
| VRDG_MIN_CELL1, 0x0C00 | Returns cell 1's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL2, 0x0C02 | Returns cell 2's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL3, 0x0C04 | Returns cell 3's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL4, 0x0C06 | Returns cell 4's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL5, 0x0C08 | Returns cell 5's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL6, 0x0C0A | Returns cell 6's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL7, 0x0C0C | Returns cell 7's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL8, 0x0C0E | Returns cell 8's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL9, 0x0C10 | Returns cell 9's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL10, 0x0C12 | Returns cell 10's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL11, 0x0C14 | Returns cell 11's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL12, 0x0C16 | Returns cell 12's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL13, 0x0C18 | Returns cell 13's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL14, 0x0C1A | Returns cell 14's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MIN_CELL15, 0x0C1C | Returns cell 15's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |

| | | | | | | |
|-------------------------|--|------------|-----------|---|-------------|-------------------|
| VRDG_MIN_CELL16, 0x0C1E | Returns cell 16's minimum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 5000mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL1, 0x0C20 | Returns cell 1's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL2, 0x0C22 | Returns cell 2's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL3, 0x0C24 | Returns cell 3's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL4, 0x0C26 | Returns cell 4's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL5, 0x0C28 | Returns cell 5's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL6, 0x0C2A | Returns cell 6's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL7, 0x0C2C | Returns cell 7's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL8, 0x0C2E | Returns cell 8's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL9, 0x0C30 | Returns cell 9's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL10, 0x0C32 | Returns cell 10's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL11, 0x0C34 | Returns cell 11's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL12, 0x0C36 | Returns cell 12's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL13, 0x0C38 | Returns cell 13's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL14, 0x0C3A | Returns cell 14's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL15, 0x0C3C | Returns cell 15's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_MAX_CELL16, 0x0C3E | Returns cell 16's maximum measured voltage. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 0.1mV | 0mV | 0mV to 6553.5mV |
| VRDG_DMAX, 0x0C40 | Returns the maximum measured voltage difference between the cells. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 2mV | 0mV | 0mV to 512mV |
| VRDG_DMAX_IDMAX, 0x0C41 | Identifies the cell with the highest voltage. This cell is used to calculate the maximum voltage difference between the cells. | 4, [3:0] | Read-only | Non-standard LSB = 1 (cell ID) Offset = 1 (cell ID) | 1 (cell ID) | 1 to 16 (cell ID) |



| | | | | | | |
|--------------------------|---|------------|-----------|---|-------------|-------------------|
| VRDG_DMAX_IDMIN, 0x0C41 | Identifies the cell with the lowest voltage. This cell is used to calculate the maximum voltage difference between the cells. | 4, [7:4] | Read-only | Non-standard LSB = 1 (cell ID) Offset = 1 (cell ID) | 1 (cell ID) | 1 to 16 (cell ID) |
| BAL_ETIME_CELL1, 0x0C42 | Returns the total time that cell 1 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL2, 0x0C44 | Returns the total time that cell 2 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL3, 0x0C46 | Returns the total time that cell 3 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL4, 0x0C48 | Returns the total time that cell 4 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL5, 0x0C4A | Returns the total time that cell 5 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL6, 0x0C4C | Returns the total time that cell 6 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL7, 0x0C4E | Returns the total time that cell 7 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL8, 0x0C50 | Returns the total time that cell 8 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL9, 0x0C52 | Returns the total time that cell 9 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL10, 0x0C54 | Returns the total time that cell 10 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL11, 0x0C56 | Returns the total time that cell 11 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL12, 0x0C58 | Returns the total time that cell 12 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL13, 0x0C5A | Returns the total time that cell 13 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL14, 0x0C5C | Returns the total time that cell 14 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL15, 0x0C5E | Returns the total time that cell 15 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| BAL_ETIME_CELL16, 0x0C60 | Returns the total time that cell 16 has been balanced. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1min | 0min | 0min to 65535min |
| TRDG_MIN_TS1, 0x0C62 | Returns the minimum temperature measured with temperature sensor 1. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MIN_TS2, 0x0C63 | Returns the minimum temperature measured with temperature sensor 2. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MIN_TS3, 0x0C64 | Returns the minimum temperature measured with temperature sensor 2. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |

| | | | | | | |
|----------------------------------|--|------------|-----------|---|-------------|------------------|
| TRDG_MIN_ TS4, 0x0C65 | Returns the minimum temperature measured with temperature sensor 2. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MAX_ TS1, 0x0C66 | Returns the maximum temperature measured with temperature sensor 1. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MAX_ TS2, 0x0C67 | Returns the maximum temperature measured with temperature sensor 2. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MAX_ TS3, 0x0C68 | Returns the maximum temperature measured with temperature sensor 3. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_MAX_ TS4, 0x0C69 | Returns the maximum temperature measured with temperature sensor 4. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 97.5°C | -30°C to +97.5°C |
| TRDG_DMAX, 0x0C6A | Returns the maximum measured temperature difference between temperature the sensors. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 0°C | 0°C to 127.5°C |
| TRDG_DMAX_ IDMAX, 0x0C6B | Identifies the cell with the highest temperature. This cell is used to calculate the maximum temperature difference between the cells. | 4, [3:0] | Read-only | Non-standard LSB = 1 (cell ID) Offset = 1 (cell ID) | 1 (cell ID) | 1 to 4 (cell ID) |
| TRDG_DMAX_ IDMIN, 0x0C6B | Identifies the cell with the lowest temperature. This cell is used to calculate the maximum temperature difference between the cells. | 4, [7:4] | Read-only | Non-standard LSB = 1 (cell ID) Offset = 1 (cell ID) | 1 (cell ID) | 1 to 4 (cell ID) |
| TRDG_ETIME_ UND0, 0x0C6C | Returns the amount of time that the battery's average temperature has been below 0°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 0TO5, 0x0C6E | Returns the amount of time that the battery's average temperature has been between 0°C and 5°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 5TO10, 0x0C70 | Returns the amount of time that the battery's average temperature has been between 5°C and 10°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 10TO15, 0x0C72 | Returns the amount of time that the battery's average temperature has been between 10°C and 15°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 15TO20, 0x0C74 | Returns the amount of time that the battery's average temperature has been between 15°C and 20°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 20TO25, 0x0C76 | Returns the amount of time that the battery's average temperature has been between 20°C and 25°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 25TO30, 0x0C78 | Returns the amount of time that the battery's average temperature has been between 25°C and 30°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ 30TO35, 0x0C7A | Returns the amount of time that the battery's average temperature has been between 30°C and 35°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |

| | | | | | | |
|---------------------------|--|------------|-----------|---|-------------|--------------------------------|
| TRDG_ETIME_35TO40, 0x0C7C | Returns the amount of time that the battery's average temperature has been between 35°C and 40°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_40TO45, 0x0C7E | Returns the amount of time that the battery's average temperature has been between 40°C and 45°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_45TO50, 0x0C80 | Returns the amount of time that the battery's average temperature has been between 45°C and 50°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_50TO55, 0x0C82 | Returns the amount of time that the battery's average temperature has been between 50°C and 55°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_55TO60, 0x0C84 | Returns the amount of time that the battery's average temperature has been between 55°C and 60°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_ETIME_ABV60, 0x0C86 | Returns the amount of time that the battery's average temperature exceeds 60°C. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1hr | 0hr | 0hr to 65535hr |
| TRDG_MIN_DIE, 0x0C88 | Returns the minimum temperature measured on the die. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | 0°C | -30°C to +97.5°C |
| TRDG_MAX_DIE, 0x0C89 | Returns the maximum temperature measured on the die. | 8, [7:0] | Read-only | 8-bit unsigned integer LSB = 0.5°C | -30°C | -30°C to +97.5°C |
| IRDG_MAX_CHG, 0x0C8A | Returns the maximum measured charge current. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | 0mA | 0mA to 131070 mA |
| IRDG_MAX_DIS, 0x0C8C | Returns the maximum measured discharge current. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 2mA | 0mA | 0mA to 131070mA |
| NCHG_CYC, 0x0C8E | Returns the number of completed charge cycles, which is when the battery pack's SOC has charged from below 15% to above 85%. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1 Event(s) | 0 events(s) | 0 events(s) to 65535 events(s) |
| NCHG_10SOC, 0x0C90 | Returns the number of times the pack has been charged by 10%. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1 event | 0 events(s) | 0 events(s) to 65535 events(s) |
| NSHDN, 0x0C92 | Returns the number of times the fuel gauge has woken up after sleep mode, a hard reset, or a shutdown. | 16, [15:0] | Read-only | 16-bit unsigned integer LSB = 1 event | 0 events(s) | 0 events(s) to 65535 events(s) |

Fuel Gauge Flags

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|-------------------------------|---|----------------------|-----------|--------------|-----------------------|--------|
| FG_EXE_FLAG, 0x0F00 | 1: Fuel gauge calculations are ongoing 0: Fuel gauge calculations are complete | 1, [0] | Read-only | Non-standard | 0 | 0 to 1 |
| EDIT_SETTINGS_FLAG, 0x0F00 | 1: Fuel gauge settings can be edited on the fly 0: Fuel gauge settings cannot be edited on the fly | 1, [1] | Read-only | Non-standard | 0 | 0 to 1 |
| CONFIG_MODE_FLAG, 0x0F00 | 1: The fuel gauge is in configuration mode 0: The fuel gauge is in running mode | 1, [2] | Read-only | Non-standard | 0 | 0 to 1 |
| CONFIG_EXIT_FLAG, 0x0F00 | 1: There is a pending exit from configuration mode 0: There is no pending exit from configuration mode | 1, [3] | Read-only | Non-standard | 0 | 0 to 1 |
| CONFIG_RST_FLAG, 0x0F00 | 1: There is a pending reset to set the configurations to their default values 0: There is no pending reset to set the configurations to their default values | 1, [4] | Read-only | Non-standard | 0 | 0 to 1 |
| FG_RST_FLAG, 0x0F00 | 1: There is no pending reset for the fuel gauge algorithm 0: There is a pending reset for the fuel gauge algorithm | 1, [5] | Read-only | Non-standard | 0 | 0 to 1 |
| LOG_RST_FLAG, 0x0F00 | 1: The lifetime log registers are being reset in the current fuel gauge iteration 0: No lifetime log registers are being reset | 1, [6] | Read-only | Non-standard | 0 | 0 to 1 |
| LOG_RST_PNDG_FLAG, 0x0F00 | 1: There is a pending reset to the lifetime log registers 0: There is no pending reset to the lifetime log registers | 1, [7] | Read-only | Non-standard | 0 | 0 to 1 |
| LEDS_ON_BTN_FLAG, 0x0F00 | 1: The LEDs are on due to the push- button 0: No LEDs are on due to the push- button | 1, [8] | Read-only | Non-standard | 0 | 0 to 1 |
| LEDS_ON_CHG_FLAG, 0x0F00 | 1: The LEDs are on due to the charge current being detected 0: The LEDs are not on due to the charge current being detected | 1, [9] | Read-only | Non-standard | 0 | 0 to 1 |
| LEDS_CHG_FLAG, 0x0F00 | 1: The LED controller considers the fuel gauge to be in a charge state 0: The LED controller does not consider the fuel gauge to be in a charge state | 1, [10] | Read-only | Non-standard | 0 | 0 to 1 |
| FG_BKUP_FLAG, 0x0F00 | 1: The fuel gauge is in learning backup mode 0: The fuel gauge is not in learning backup mode Use FG_BKUP_ENABLE to exit/enter this mode. | 1, [11] | Read-only | Non-standard | 0 | 0 to 1 |

Fuel Gauge Settings

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|---------------------|--|----------------------|------|--------------------------------------|-----------------------|-------------|
| EXE_TIME, 0x1000 | Sets the fuel gauge execution time interval. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.1s | 4s | 0.5s to 16s |

| | | | | | | |
|----------------------------|--|------------|-----|---|----------|--------------------|
| WEXE_TIME_REST, 0x1001 | Updates the most time-consuming functions when the fuel gauge is resting each (WEXE_TIME_REST x EXE_TIME), instead of each EXE_TIME. The maximum allowed value for the execution period of this functions is 32s, so if the fuel gauge is set such that (WEXE_TIME_REST x EXE_TIME) is longer than 32s, then the period used is the multiple of EXE_TIME, that is closest to but shorter than 32s. | 5, [4:0] | R/W | 8-bit unsigned integer LSB = 1 | 4 | 1 to 16 |
| DIS_ITH, 0x1002 | Sets the discharge status current threshold. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 1mA | 25mA | 0mA to 65535mA |
| CHG_ITH, 0x1004 | Sets the charge status current threshold. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 1mA | 25mA | 0mA to 65535mA |
| SOC_DMAX, 0x1006 | Sets the maximum SOC for the charge and discharge statuses. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.00005%/s | 0.07%/s | 0.00005%/s to 2%/s |
| SOC_DMAX_REST, 0x1008 | Sets the maximum SOC slope for the resting status. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.00005%/s | 0.001%/s | 0%/s to 2%/s |
| VRDG_PACK_EN, 0x1100 | Enables the VRDG_PACK input (pack voltage reading). | 1, [0] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| IRDG_PACK_EN, 0x1100 | Enables the IRDG_PACK input (pack current reading). | 1, [1] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| IRDG_CELLS_EN, 0x1100 | Enables the IRDG_CELLx inputs (cell current reading). | 1, [2] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| CCRDG_EN, 0x1100 | Enables the CCRDG input (coulomb counting). | 1, [3] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| SOH_LRN_EN, 0x1100 | Enables SOH learning. | 1, [7] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| ICHG_CC_LRN_EN, 0x1100 | Enables charge CC learning. | 1, [10] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| ICHG_END_LRN_EN, 0x1100 | Enables charge termination current learning. | 1, [11] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| VCHG_CV_LRN_EN, 0x1100 | Enables charge CV learning. | 1, [12] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| IDIS_AVG_LRN_EN, 0x1100 | Enables discharge average current learning. | 1, [13] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| IDIS_END_LRN_EN, 0x1100 | Enables discharge termination current learning. | 1, [14] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |

| | | | | | | |
|------------------------------|---|----------|-----|---------------------------------------|---|--------|
| IDIS_END_LRN_CONT_EN, 0x1100 | Enables continuous discharge termination current learning. | 1, [15] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| PCHG_SHW_EN, 0x1100 | Sets the PCHG_SHW default value (show maximum charge power). | 1, [18] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| PDIS_SHW_EN, 0x1100 | Sets the PDIS_SHW default value (show maximum discharge power). | 1, [19] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| TSS_CELL1, 0x1200 | Sets cell 1's temperature sensor source. | 2, [1:0] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL2, 0x1200 | Sets cell 2's temperature sensor source. | 2, [3:2] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL3, 0x1200 | Sets cell 3's temperature sensor source. | 2, [5:4] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL4, 0x1200 | Sets cell 4's temperature sensor source. | 2, [7:6] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL5, 0x1201 | Sets cell 5's temperature sensor source. | 2, [1:0] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL6, 0x1201 | Sets cell 6's temperature sensor source. | 2, [3:2] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL7, 0x1201 | Sets cell 7's temperature sensor source. | 2, [5:4] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL8, 0x1201 | Sets cell 8's temperature sensor source. | 2, [7:6] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL9, 0x1202 | Sets cell 9's temperature sensor source. | 2, [1:0] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL10, 0x1202 | Sets cell 10's temperature sensor source. | 2, [3:2] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL11, 0x1202 | Sets cell 11's temperature sensor source. | 2, [5:4] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |

| | | | | | | |
|-----------------------|---|------------|-----|--|------|-----------------|
| TSS_CELL12, 0x1202 | Sets cell 12's temperature sensor source. | 2, [7:6] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL13, 0x1203 | Sets cell 13's temperature sensor source. | 2, [1:0] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL14, 0x1203 | Sets cell 14's temperature sensor source. | 2, [3:2] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL15, 0x1203 | Sets cell 15's temperature sensor source. | 2, [5:4] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| TSS_CELL16, 0x1203 | Sets cell 16's temperature sensor source. | 2, [7:6] | R/W | Non-standard LSB = 1 Offset = 1 | 1 | 1 to 4 |
| NCELLS_SER, 0x1204 | Sets the number of series-connected cells in the battery pack. | 5, [4:0] | R/W | 8-bit unsigned integer LSB = 1 | 16 | 2 to 16 |
| NCELLS_PAR, 0x1205 | Sets the number of parallel-connected cells in the battery pack. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1 | 1 | 1 to 255 |
| NTSS, 0x1206 | Sets the number of temperature sensors in the battery pack. | 3, [2:0] | R/W | 8-bit unsigned integer LSB = 1 | 1 | 1 to 4 |
| RCXN_PACK, 0x1207 | Sets the connection resistance between the battery and the battery management system (BMS) terminals. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 10mΩ | 0mΩ to 6553.5mΩ |
| RBMS_HSIDE, 0x1209 | Sets the BMS's high-side resistance. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 4mΩ | 0mΩ to 6553.5mΩ |
| RBMS_LSIDE, 0x120B | Sets the BMS's low-side resistance. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 2mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL1, 0x120D | Sets the connection resistance between cell 1's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL2, 0x120F | Sets the connection resistance between cell 2's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL3, 0x1211 | Sets the connection resistance between cell 3's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL4, 0x1213 | Sets the connection resistance between cell 4's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |

| | | | | | | |
|--------------------------|--|------------|-----|--|-------|-----------------|
| RCXN_CELL5, 0x1215 | Sets the connection resistance between cell 5's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL6, 0x1217 | Sets the connection resistance between cell 6's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL7, 0x1219 | Sets the connection resistance between cell 7's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL8, 0x121B | Sets the connection resistance between cell 8's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL9, 0x121D | Sets the connection resistance between cell 9's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL10, 0x121F | Sets the connection resistance between cell 10's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL11, 0x1221 | Sets the connection resistance between cell 11's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL12, 0x1223 | Sets the connection resistance between cell 12's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL13, 0x1225 | Sets the connection resistance between cell 13's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL14, 0x1227 | Sets the connection resistance between cell 14's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL15, 0x1229 | Sets the connection resistance between cell 15's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| RCXN_CELL16, 0x122B | Sets the connection resistance between cell 16's terminals and its corresponding voltage-sensing probes. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mΩ | 1mΩ | 0mΩ to 6553.5mΩ |
| ICHG_MAX, 0x122D | Sets the maximum charge current that the pack can accept. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 4000 | 0mA to 131070mA |
| ICHG_MAX_MRGN, 0x122F | Sets the maximum charge current margin. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 2mA | 120 | 0mA to 1020mA |
| IDIS_MAX, 0x1230 | Sets the maximum discharge current that the pack can accept. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 20000 | 0mA to 131070mA |
| IDIS_MAX_MRGN, 0x1232 | Sets the maximum discharge current margin. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 2mA | 500 | 0mA to 1020mA |

| | | | | | | |
|-------------------------------------|--|---------------|-----|--|---------|--------------------|
| VCELL_MAX, 0x1233 | Sets the maximum voltage accepted at the cell level. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 4200mV | 0mV to 6553.5mV |
| VCELL_MAX_ MRGN, 0x1235 | Sets the maximum cell voltage margin. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.5mV | 5mV | 0mV to 127.5mV |
| VCELL_MIN, 0x1236 | Sets the minimum voltage accepted at the cell level. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 0.1mV | 3000mV | 0mV to 6553.5mV |
| VCELL_MIN_ MRGN, 0x1238 | Sets the minimum cell voltage margin. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.5mV | 5mV | 0mV to 127.5mV |
| VPACK_MAX, 0x1239 | Sets the maximum voltage that the pack can accept. This value is used for charge power estimate. The pack float voltage represents the maximum voltage that the battery pack can reach in a charging scenario. It should be set the be equal to or exceed the charging system's CV. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mV | 66800mV | 0mV to 131070mV |
| VPACK_MAX_ MRGN, 0x123B | Sets the pack's maximum pack voltage margin. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 4mV | 48mV | 0mV to 1020mV |
| VPACK_EMPTY, 0x123C | Sets the pack empty voltage, which represents the lowest usable pack voltage. This is the voltage at which the application considers the battery pack to be completely discharged and stops operation. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mV | 48800mV | 0mV to 131070mV |
| VPACK_EMPTY_ MRGN, 0x123E | Sets the margin for the power learning algorithm, which is applied to the pack empty voltage. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 4mV | 48mV | 0mV to 1020mV |
| IDIS_AVG_SET, 0x1400 | Sets the nominal average discharge current (i.e. IDIS_AVG). | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 2500 | 2mA to 131070mA |
| IDIS_END_SET, 0x1402 | Sets the nominal discharge termination current (i.e. IDIS_END). | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 2500 | 2mA to 131070mA |
| EMPTY_RUNTIME_ FILTER, 0x1406 | Sets the filter time constant for the remaining time-to-empty (i.e. EMPTY_RUNTIME). Increasing EMPTY_RUNTIME_FILTER results in a smoother estimate at the tradeoff of an increased phase delay. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1s | 60s | 0s to 255s |
| ICHG_CC_SET, 0x1500 | Sets the nominal charge constant current (i.e. ICHG_CC).It should be set to the nominal current that applied by the external charger to the battery pack while in the charge CC region. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 2500 | 2mA to 131070mA |
| ICHG_END_ SET, 0x1502 | Sets the nominal charge termination current (i.e. ICHG_END). It should be set to the nominal current, which is used to detect the end of charge while in the charge CV region. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 2500 | 2mA to 131070mA |
| VCHG_CV_SET, 0x1504 | Sets the nominal charge constant voltage (i.e. VCHG_CV). This value should be set to the nominal voltage, which is applied to the battery pack by the external charger during CV charge. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mV | 66400mV | 0mV to 131070mV |

| | | | | | | |
|-----------------------------|---|----------|-----|---|-------------|-------------------------|
| VCHG_CV_MRGN, 0x1506 | Sets the charge constant voltage (i.e. VCHG_CV) margin. | 8, [7:0] | R/W | 16-bit unsigned integer LSB = 2mV | 50mV | 0mV to 510mV |
| FULL_RUNTIME_FILTER, 0x150A | Sets the filter time constant for the remaining time-to-full (i.e. FULL_RUNTIME). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1s | 10s | 0s to 255s |
| ICHG_CC_LRN_RNG, 0x1600 | If ICHG_CC_LRN_EN is enabled, these bits set the charge CC (i.e. ICHG_CC) learning range. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1% | 10% | 0% to 100% |
| ICHG_END_LRN_RNG, 0x1601 | If ICHG_END_LRN_EN enabled, these bits set the charge termination current (i.e. ICHG_END) learning range. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1% | 10% | 0% to 100% |
| VCHG_CV_LRN_RNG, 0x1602 | If VCHG_CV_LRN_EN is enabled, these bits set the charge CV (i.e. VCHG_CV) learning range. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1% | 3% | 0% to 100% |
| ICHG_LRN_ITH, 0x1603 | Sets the charge current threshold for current learnings (i.e. ICHG_END and VCHG_CV). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.01 C-rate | 0.01 C-rate | 0 C-rate to 2.55 C-rate |
| ICHG_CC_LRN_DITH, 0x1604 | Sets the charge current derivative threshold for charge CC learning (i.e. ICHG_CC). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.01mA/s | 0.1mA/s | 0mA/s to 2.55mA/s |
| VCHG_LRN_DVTH, 0x1605 | Sets the charge voltage derivative threshold for charge CV learning (i.e. VCHG_CV and ICHG_END). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.01mV/s | 0.1mV/s | 0mV/s to 2.55mV/s |
| IDIS_END_LRN_RNG, 0x1700 | If IDIS_END_LRN_EN is enabled, these bits set the discharge termination current (i.e. IDIS_END) learning range. | 7, [6:0] | R/W | 8-bit unsigned integer LSB = 1% | 10% | 0% to 100% |
| IDIS_END_LRN_SOCTH, 0x1701 | If IDIS_END_LRN_EN is enabled, these bits set the SOC discharge termination current (i.e. IDIS_END) learning threshold. | 7, [6:0] | R/W | 8-bit unsigned integer LSB = 1% | 50% | 0% to 100% |
| IDIS_END_LRN_KPEAK, 0x1702 | If IDIS_END_LRN_EN is enabled, these bits set the discharge termination current (i.e. IDIS_END) learning peak current gain. | 7, [6:0] | R/W | 8-bit unsigned integer LSB = 1% | 70% | 0% to 100% |
| IDIS_LRN_ITH, 0x1703 | Sets the discharge termination current threshold for learnings (i.e. IDIS_AVG, IDIS_END). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.01 C-rate | 0.05 C-rate | 0 C-rate to 2.55 C-rate |
| IDIS_END_LRN_FILTER, 0x1704 | Sets the filter time constant for discharge current learnings (i.e. IDIS_AVG and IDIS_END). | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1s | 10s | 0s to 255s |
| SOH_LRN_K, 0x1800 | If SOH_LRN_EN is enabled, these bits set the SOH learning gain. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.001 | 0.1s | 0s to 0.255s |
| SOH_LRN_DMAX, 0x1801 | If SOH_LRN_EN is enabled, these bits set the maximum increase for SOH learning. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.1% | 1% | 0% to 25.5% |
| SOH_LRN_DMIN, 0x1802 | If SOH_LRN_EN is enabled, these bits set the maximum decrease for SOH learning. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.1% | 1% | 0% to 25.5% |
| SOH_LRN_MAX, 0x1803 | If SOH_LRN_EN is enabled, these bits set the maximum value for SOH learning. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1% | 110% | 100% to 255% |
| SOH_LRN_MIN, 0x1804 | If SOH_LRN_EN is enabled, these bits set the minimum value for SOH learning. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 1% | 80% | 0% to 100% |
| SOH_LRN_TMAX, 0x1805 | If SOH_LRN_EN is enabled, these bits set the maximum temperature for SOH learning | 8, [7:0] | R/W | 8-bit signed, 2-comp integer LSB = 1°C | 30°C | -128°C to +127°C |

| | | | | | | |
|-------------------------|---|----------|-----|---|------|------------------|
| SOH_LRN_TMIN, 0x1806 | If SOH_LRN_EN is enabled, these bits set the minimum temperature for SOH learning | 8, [7:0] | R/W | 8-bit signed, 2-comp integer LSB = 1°C | 10°C | -128°C to +127°C |
|-------------------------|---|----------|-----|---|------|------------------|

Fuel Gauge Model

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|--------------------------|---|----------------------|------|---------------------------------------|-----------------------|------------------|
| CELL_MFR, 0x2000 | Sets the cell manufacturer. | 128, [127:0] | R/W | ASCII character(s) | Default | - |
| CELL_TYPE, 0x2010 | Sets the cell type model. | 128, [127:0] | R/W | ASCII character(s) | Default | - |
| MDL_QNOM, 0x2036 | Sets the nominal capacity of the cell, which is reported by the manufacturer. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 1mAh | 2500mAh | 1mAh to 65535mAh |
| MDL_QMEAS, 0x2038 | Sets the capacity measured in the cell characterization test. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 1mAh | 2437mAh | 1mAh to 65535mAh |
| CELL_ICHG_MAX, 0x203E | Sets the maximum charge current that the cell can handle, according to the manufacturer. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 4000 | 0mA to 131070mA |
| CELL_IDIS_MAX, 0x2040 | Sets the maximum discharge current that the cell can handle, according to the manufacturer. | 16, [15:0] | R/W | 16-bit unsigned integer LSB = 2mA | 20000 | 0mA to 131070mA |
| CELL_TMIN, 0x2042 | Sets the minimum temperature that the cell can handle, according to the manufacturer. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.5°C | 0°C | -128°C to +127°C |
| CELL_TMAX, 0x2043 | Sets the maximum temperature that the cell can handle, according to the manufacturer. | 8, [7:0] | R/W | 8-bit unsigned integer LSB = 0.5°C | 50°C | -128°C to +127°C |

LED Settings

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|------------------------|---|----------------------|------|--------------------------------------|-----------------------|--------|
| LEDS_ON_MAN, 0x4000 | 1: The LEDs are controlled manually using the dedicated register bits 0: The LEDs are directly controlled to report the SOC | 1, [0] | R/W | Control Enable = 1 Disable = 0 | 0 | 0 to 1 |
| LEDS_ON_CHG, 0x4000 | 1: The LEDs display the pack SOC while charging, if the manual LED controls are set to 0 0: The LEDs do not display the pack SOC while the charging and manual LED controls are set to 0 | 1, [1] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| LEDS_ON_BTN, 0x4000 | 1: The LEDs display the pack SOC after the LED push-button is pressed if the manual LED controls are set to 0 0: The LEDs do not display the pack SOC after the LEDs push-button is pressed and the manual LED controls are set to 0 | 1, [2] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |
| LED1_ON, 0x4000 | 1: If manual LED control is set to 1 the LED turns on 0: If manual LED control is set to 1 the LED turns off | 1, [3] | R/W | Control On = 1 Off = 0 | 0 | 0 to 1 |
| LED2_ON, 0x4000 | 1: If manual LED control is set to 1 the LED turns on 0: If manual LED control is set to 1 the LED turns off | 1, [4] | R/W | Control On = 1 Off = 0 | 0 | 0 to 1 |
| LED3_ON, 0x4000 | 1: If manual LED control is set to 1 the LED turns on 0: If manual LED control is set to 1 the LED turns off | 1, [5] | R/W | Control On = 1 Off = 0 | 0 | 0 to 1 |

| | | | | | | |
|-------------------------|---|----------|-----|---|----|-------------|
| LED4_ON, 0x4000 | 1: If manual LED control is set to 1 the LED turns on 0: If manual LED control is set to 1 the LED turns off | 1, [6] | R/W | Control On = 1 Off = 0 | 0 | 0 to 1 |
| LED5_ON, 0x4000 | 1: If manual LED control is set to 1 the LED turns on 0: If manual LED control is set to 1 the LED turns off | 1, [7] | R/W | Control On = 1 Off = 0 | 0 | 0 to 1 |
| LED_TRANS, 0x4001 | Sets the deglitch time to control when the LED turns on/off upon transitioning between charge and discharge mode. | 7, [6:0] | R/W | 8-bit unsigned integer LSB = 0.1s | 4s | 0s to 12.7s |
| LEDS_ON_TIME, 0x4002 | Sets the time during which the LEDs are on after pressing the LED push-button. | 4, [3:0] | R/W | 8-bit unsigned integer LSB = 1s Offset = 1s | 5s | 1s to 16s |

I²C Settings

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|------------------------|--|----------------------|------|--------------------------------------|-----------------------|----------|
| I2C_ADDRESS, 0x4100 | Selects the slave I ² C address of the device. | 7, [6:0] | R/W | 8-bit unsigned integer LSB = 1 | 8 | 0 to 127 |
| I2C_CRC_EN, 0x4100 | 1: Enable CRC for I ² C communication. Only write commands with the correct CRC are accepted. The response after a read command includes the 4 CRC bytes at the end 0: Disable CRC for I ² C communication. Write commands with no CRC bytes (or with the correct CRC) are accepted. The response after a read command does not include the 4 CRC bytes | 1, [7] | R/W | Control Enable = 1 Disable = 0 | 1 | 0 to 1 |

Firmware Version

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|-------------------|--|----------------------|-----------|--------------------|-----------------------|-------|
| IC_VER, 0x6100 | Returns the IC version. 42790: 2 to 16 cells in series with LED support 42792: 2 to 16 cells in series cells 42795: 2 to 10 cells with LED support 42797: 2S to 10 in series cells 42791: 2 to 16 cells in series with resistance detection and a thermal model | 40, [39:0] | Read-only | ASCII character(s) | 42790 | - |

Commands

| Name, Address | Description | Bit Length, Position | Type | Encoding | Decoded Default Value | Range |
|--------------------|---|----------------------|------------|---|-----------------------|----------|
| RST_CMD, 0x7FFF | Resets the fuel gauge calculations. 0x01: Reset the fuel gauge. This is a self-clearing function | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |

| | | | | | | |
|--------------------------------|---|----------|------------|---|---|----------|
| EXE_CMD, 0x7FFE | Triggers the fuel gauge calculations. The system MCU is responsible for periodically writing this register to trigger an SOC update. This function should be triggered after providing all the updated voltage and current readings that are used to produce the updated SOC estimate. 0x01: Trigger a fuel gauge update refresh | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| EDIT_CONFIG_CMD, 0x7FFD | 0x01: The fuel gauge settings can be edited | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| END_EDIT_CONFIG_CMD, 0x7FFC | 0x01: The fuel gauge settings cannot be edited | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| CONFIG_MODE_CMD, 0x7FFB | Directs the fuel gauge to enter configuration mode (cell profile, battery pack, and algorithm tuning). 0x01: Enter configuration mode | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| CONFIG_EXIT_CMD, 0x7FFA | Directs the fuel gauge to exit configuration mode (saves configuration values to flash) and reset. If not in configuration mode, the device stores the fuel gauge settings from the volatile memory to the NVM. 0x01: Exit configuration mode. The new configuration is stored in the NVM | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| CONFIG_RST_CMD, 0x7FF9 | The configuration values are re-initialized to the default values and the fuel gauge calculations are reset. 0x01: Reset the configuration. This is a self-clearing function | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |
| LOG_RST_CMD, 0x7FF8 | Re-initializes lifetime logging. 0x01: Re-initialize the lifetime log registers. This is a self-clearing function | 8, [7:0] | Write-only | Active high command 1 = Execute LSB = 1 | 0 | 0 to 255 |

APPLICATION INFORMATION

Core Algorithm

The fuel gauge core algorithm accurately estimates core cell-level and pack-level parameters. These core parameters can be found in the Fuel Gauge Output register map.

Cell Mathematical Model

Cell dynamics are influenced by multiple factors, such as chemistry, SOC, current, and temperature. A mathematical model that captures the cell's most important dynamics is required to accurately measure internal variables.

Cell mathematical models are provided by MPS. These models are either from a database or they are a result of the cell characterization routine.

Cell State-of-Charge (SOC)

Cell SOC refers to the cell capacity that is currently available, as a function of the rated capacity. SOC is an unknown, internal variable that cannot be directly measured, meaning it must be estimated using available measurements. The MPF42790 uses voltage, current, and temperature measurements observed over time to produce accurate SOC estimates across a wide range of operating conditions.

Empty State-of-Charge (SOC)

Empty SOC refers to the SOC level at which the battery pack is unable to discharge further without reaching the voltage cutoff limits. The empty SOC prediction relies on the operating conditions of the different cells, as well as the discharge current supplied to the load, which is specific to each application.

In addition, the MPF42790 can predict whether the battery pack discharge is limited by the pack or one of the cells (the cell ID is provided). This is extremely important for accurate empty SOC estimation in unbalanced packs, since the fuel gauge must track the condition that limits the battery pack discharge.

Full State-of-Charge (SOC)

Full SOC refers to the SOC level at which the battery pack is unable to charge further without leaving the safe operating area (SOA). The full SOC prediction relies on the operating

conditions of the different cells, as well as the constant-current/constant-voltage (CC/CV) charging sequence, which is specific to each application.

The MPF42790 can predict whether the battery pack charge is limited by the pack or one of the cells (the cell ID is provided). This is extremely important for accurate full SOC estimation in unbalanced packs, since the fuel gauge must track the condition that limits the battery pack charge.

Pack State-of-Charge (SOC)

Pack SOC provides the capacity, which is a function of the available capacity given the actual pack operating conditions. Pack SOC relies on cell-level quantities and tracks the remaining capacity that the battery can charge or discharge without reaching the cutoff limits.

Pack SOC can be considered the most vital estimate since it is typically the value shown to the application end user.

Pack Unavailable State-of-Charge (SOC)

Pack unavailable SOC refers to the capacity that is not available due to the actual pack operating conditions (e.g. mismatched cells and battery pack configurations). Pack unavailable SOC provides the capacity that the battery cannot use without violating the defined SOA.

Indicators

In addition to the core parameters, an extensive set of fuel gauge indicators are provided, based on the internal battery estimates. These indicators can be found in the Fuel Gauge Output register map.

Remaining Time-to-Empty

Remaining time-to-empty (EMPTY_RUNTIME) refers to how much time is left until the battery pack reaches the empty state while assuming the application-specific discharge current. This estimate informs the user of the remaining runtime until the pack reaches 0% SOC.

Remaining Time-to-Full

Remaining time-to-full (FULL_RUNTIME) refers to how much time is left until the battery pack reaches the cutoff limits using the application-specific charge parameters (i.e. the CC and CV

levels). This estimate informs the user of the remaining charge time until the pack reaches 100% SOC.

Maximum Discharge Power

The maximum discharge power (PDIS) refers to how much power the pack can deliver to the load without reaching the pack empty levels and leaving the SOA.

The MPF42790 can predict whether the limiting factor under the maximum discharge power is caused by the pack or one of the cells (the cell ID is provided).

Maximum Charge Power

The maximum charge power (PCHG) refers to how much power the pack can accept for charging without breaching the cutoff limits.

The MPF42790 can predict whether the limiting factor under the maximum charge power is caused by the pack or one of the cells (the cell ID is provided).

Learnings

Fuel gauge learnings are responsible for automatically fine-tuning the algorithm and tracking aging to keep estimates accurate across the device's lifetime.

Charge Constant Current (CC)

The charge CC learning (ICHG_CC_LRN) feature refines the initial charge CC value defined by the user. This smart adjustment compensates for when the charge CC deviates from its nominal settings, such as when there are production tolerances to the charging system. This feature minimizes the need for tuning while maintaining optimal SOC accuracy. The learning can be completed and leveraged within the same charge cycle.

Charge Termination Current

The charge termination current learning (ICHG_END_LRN) feature refines the initial charge termination current value defined by the user. This smart adjustment compensates for when the charge termination current deviates from its nominal settings, such as when there are production tolerances to the charging system.

Charge termination current smart adjustments are applied in subsequent charge cycles

following the cycle in which the learning occurred.

Charge Constant Voltage (CV)

The charge CV learning (VCHG_CV_LRN) feature refines the initial charge CV value defined by the user. This smart adjustment compensates for when the charge CV deviates from its nominal settings, such as production tolerances to the charging system. Charge CV smart adjustments are applied in subsequent charge cycles following the cycle in which the learning occurred.

Discharge Average Current

The discharge average current learning (IDIS_AVG_LRN) feature calculates the load average discharge current to accurately provide the time-to-empty estimate.

Discharge Termination Current

The discharge termination current learning (IDIS_END_LRN) feature refines the initial discharge termination current value defined by the user. This value adapts to load dynamics and keeps the SOC accurate with minimal user tuning.

In addition, this feature includes two different operation modes depending on the application requirements: continuous and past-cycle learning. In continuous learning mode, the discharge termination current updates continuously while the pack discharges to accurately determine the empty SOC. Past-cycle learning uses the previous discharge cycle data to learn. Continuous learning is recommended for unpredictable discharge termination currents, whereas past-cycle learning is recommended for consistent discharge termination currents.

State-of-Health (SOH)

As battery cells age, their total capacity decreases due to unwanted side reactions and structural deterioration (often referred to as capacity fading). Up-to-date state-of-health (SOH) information is important since it has a major impact on the battery pack's available energy.

The MPF42790 incorporates an SOH learning (SOH_LRN) feature that can track individual cells' capacities (SOH_CELLx) with respect to their nominal capacity. This feature can identify

cells that are negatively impacting the overall pack performance.

Lifetime Log

Important events and certain conditions are reported in the lifetime log to gain insight on how the battery pack has been operating. The MPF42790 backs the relevant registers up to its non-volatile memory (NVM) every time the part enters disabled mode (or for each elapsed hour).

Count Log

The count log is the number of times the pack has been charged from below 15% to above 85%. The number of times the pack has been charged by 10% and the number of shutdowns are also reported in the lifetime log. Full shutdowns and entering disabled mode are considered shutdowns, and increase the counter.

Current Log

The current log is the maximum charge and discharge current that has been applied to the pack. It is reported in the lifetime log.

Voltage Log

The voltage log tracks the maximum and minimum voltage for each cell. This is reported in the lifetime log. In addition, the maximum cell voltage difference between two cells is also reported, as well as their corresponding cell IDs.

Temperature Log

The temperature log tracks the maximum and minimum temperature sensed on the pack. It also reports the maximum temperature difference between two sensors, and their corresponding temperature sensor IDs.

The lifetime log details how much time the pack has operated under different temperature ranges.

Balancing Log

The balancing log tracks the number of times each cell has been balanced.

PCB Layout Guidelines

Place the bypass capacitor between each pair of VDD and GND pins, and as close as possible to the pin.

TYPICAL APPLICATION CIRCUIT

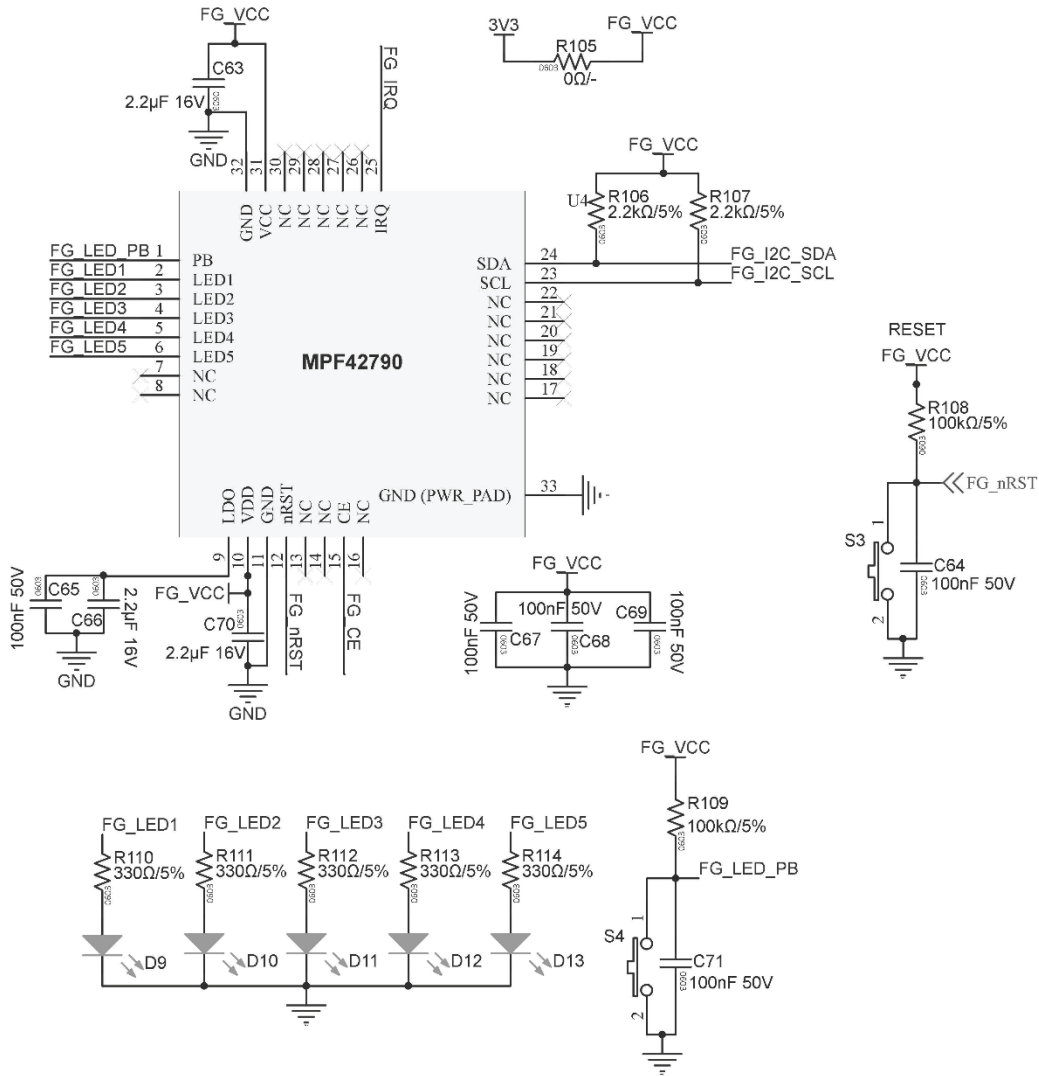
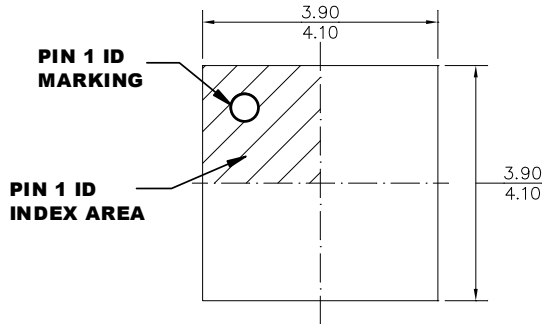


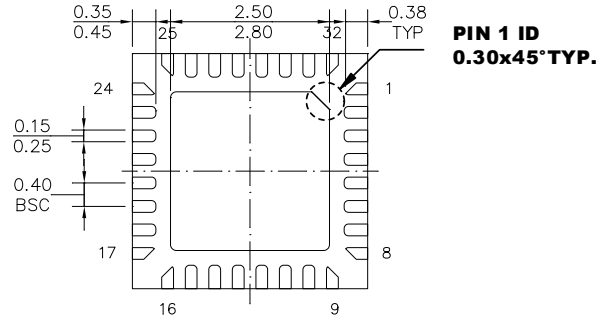
Figure 12: Application Circuit

PACKAGE INFORMATION

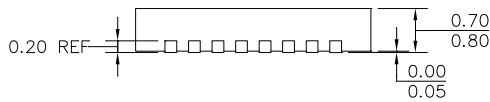
QFN-32 (4mmx4mm)



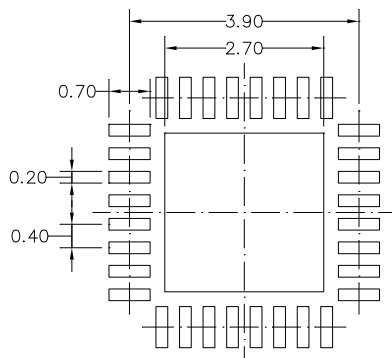
TOP VIEW



BOTTOM VIEW



SIDE VIEW

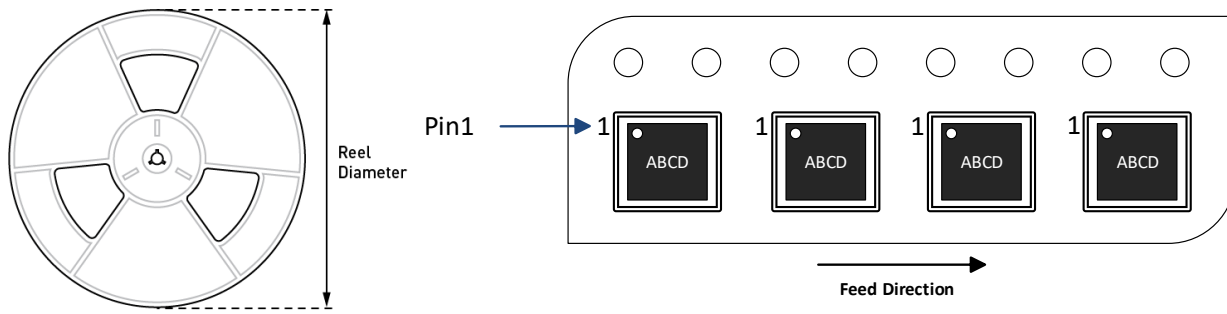


RECOMMENDED LAND PATTERN

NOTE:

- 1) ALL DIMENSIONS ARE IN MILLIMETERS.
- 2) EXPOSED PADDLE SIZE DOES NOT INCLUDE MOLD FLASH.
- 3) LEAD COPLANARITY SHALL BE 0.08 MILLIMETERS MAX.
- 4) JEDEC REFERENCE IS MO-220.
- 5) DRAWING IS NOT TO SCALE.

CARRIER INFORMATION



| Part Number | Package Description | Quantity/ Reel | Quantity/ Tube | Quantity/ Tray | Reel Diameter | Carrier Tape Width | Carrier Tape Pitch |
|-----------------------|---------------------|----------------|----------------|----------------|---------------|--------------------|--------------------|
| MPF42790DRT-0B-yyyy-Z | TQFN-32 (4mmx4mm) | 5000 | N/A | N/A | 13in | 12mm | 8mm |



REVISION HISTORY

| Revision # | Revision Date | Description | Pages Updated |
|------------|---------------|-----------------|---------------|
| 1.0 | 9/7/2022 | Initial Release | - |

Notice: The information in this document is subject to change without notice. Please contact MPS for current specifications. Users should warrant and guarantee that third-party Intellectual Property rights are not infringed upon when integrating MPS products into any application. MPS will not assume any legal responsibility for any said applications.