

**1. Introduction**

The RRC-PMM240, which is shown in Figure 1, is a power management module which can deliver up to 240W through the DC output connector. The module can be integrated into an application to supply power and charge a smart battery compliant with Smart Battery Data Specification Rev 1.1.

**2. General Overview**

The PMM240 is designed to be used as a power management module. This means that it manages the power between the power supply, the smart battery and the user application. Therefore, when a power supply is attached it is able to simultaneously charge the battery and supply power to the application. When the power supply is detached, it switches the application’s power source to the battery. The connectors of the PMM240 are shown in Figure 1.

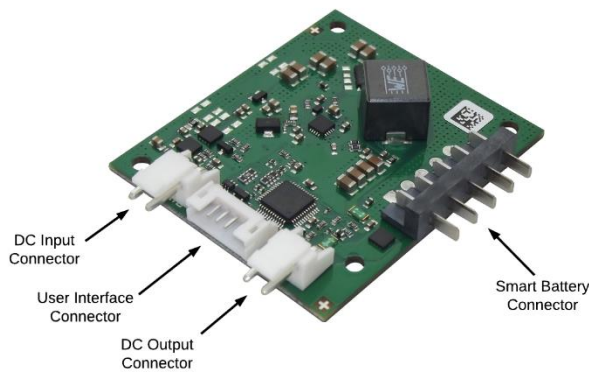


Figure 1 – PMM240 board and connectors

The DC input connector which is protected against reverse polarity is the input for the external power supply unit (PSU). The DC output connector is the power output and is intended to attach to the user’s application. The User Interface (UI) connector establishes an SMBus communication line between the user’s application, the smart battery and the PMM240 itself. It has two GPIOs (General Purpose Input/Output) that inform the user of the PMM240’s current status.

Figure 2 illustrates a typical layout. The PMM240’s charger is capable of establishing communication with the smart battery and requests its desired charging voltage and charging current. The PMM240 queries the smart battery for its desired charging parameters and charges it accordingly.

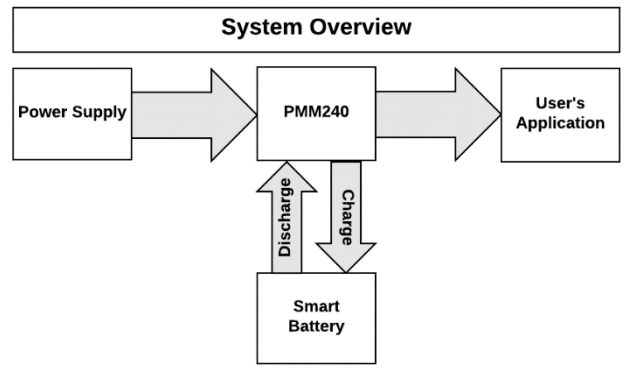


Figure 2 – PMM240’s system overview

**3. Getting Started**

To get started with the PMM240 it is necessary to choose an appropriate PSU that is able to charge the battery and power the user’s application, as described in Chapter 3.1. It is also necessary to configure the content of at least one of the registers according to the PSU and battery used, as described in Chapter 3.2.

**3.1 Selecting an appropriate PSU**

The PMM240 valid input voltage range and input current range are listed in Table 1.

<b>Input Voltage Range</b>	7.5V – 24V
<b>Input Current Range</b>	0A – 8A

Table 1 – Input ratings of the PMM240

The rated voltage of the PSU must be at least 1V above the maximum charging voltage requested by the battery. Table 2 provides suggestions on PSUs which are suitable for different battery architectures based on the number of cells connected in series.

<b>Battery architecture (ySxP)</b> (y cells in series, x cells in parallel)	<b>Typical DC input voltage</b>
1SxP	12V
2SxP (e.g. RRC2057)	12V
3SxP (e.g. RRC2020)	15V
4SxP (e.g. RRC2024)	19V

Table 2 – Recommended input voltage range for different battery architectures

The PSU must be able to maintain its output voltage even in the highest load condition. If the PSU output voltage drops lower than 0.5V above the battery’s maximum requested charging voltage, the PMM240 will disconnect the PSU from the DC input connector and try to support the application through battery power.

The PMM240 neither limits the current drawn from the application nor any inrush current. It is necessary to take care of any abusive inrush currents that may exist in the designed set-up.

**3.2 User Commands: Concept**

The user must configure at least one of the two user command parameters: the “Input Current Limit” and the “Charge Current Limit”. These two parameters have to be written through SMBus either on the EEPROM or RAM, as described in Chapter 4.

Figure 3 presents an explanation of the concept of these parameters. In this figure  $I_{PSUmax}$  is the maximum PSU output current and  $I_{input}$  is the sum of the current required by the application ( $I_{load}$ ) and the current applied to charge the battery ( $I_{bat}$ ) as written in equation (1), where  $V_{bat}$  is the battery voltage and  $V_{PSU}$  is the PSU output voltage.

$$I_{input}(t) = I_{load}(t) + \frac{I_{bat}(t) \times V_{bat}}{V_{PSU}} \quad (1)$$

The “Charge Current Limit” establishes the maximum allowed  $I_{bat}(t)$  as written in expression (2):

$$I_{bat}(t) \leq \text{“Charge Current Limit”} \quad (2)$$

The basic concept of the PMM240’s operation is:

$$I_{input}(t) \leq \text{“Input Current Limit”} \quad (3)$$

When  $I_{input} = \text{“Input Current Limit”}$  and the application requires more power ( $t = t_a$ ), the PMM240 decreases  $I_{bat}(t)$  as the power priority is with the application.

When  $I_{input} > I_{load}$  ( $t = t_b$ ) two different possibilities could occur:

1. The PMM240 enters turbo boost mode.
2. If not, the PMM240 continues to power the application while:

$$I_{input} \leq I_{PSUmax} \quad (4)$$

Once  $I_{input} > I_{PSUmax}$  ( $t = t_c$ ), the PMM240 detects the PSU output voltage drop and opens its input protection MOSFETs thus disconnecting the PSU from the DC input

connector. Not correctly setting the “Input Current Limit” could repeatedly trigger this protection which could permanently damage the PMM240.

The user has to configure the “Input current Limit” and/or the “Charge Current Limit” in compliance with its power supply, application and battery used.

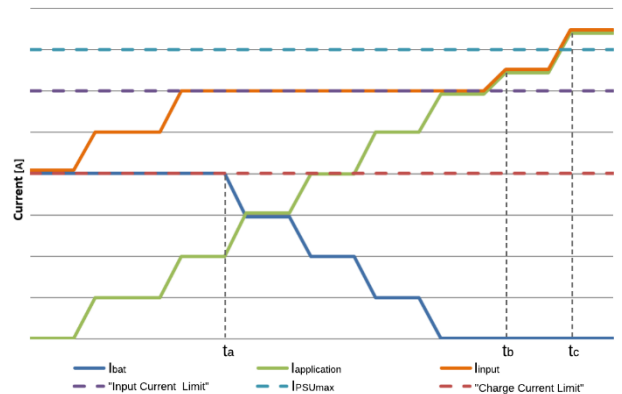


Figure 3 – PMM240 operational example

**4. Configuring the user commands**

To configure “Input Current Limit” and “Charge Current Limit” it is necessary to send an SMBus write word command to the SMBus charger address. For these values to be written it is necessary to have a PSU with valid output voltage attached to the DC input connector. The structure of the command is specified in Figure 4 and detailed in Table 3.

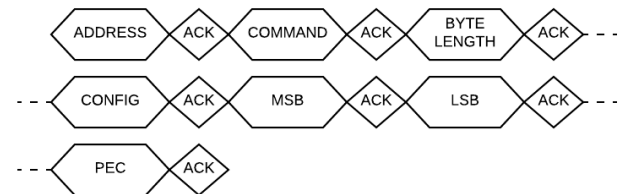


Figure 4 - An overview of the structure of the user commands



Byte	Description	Values
1: ADDRESS	SMBus Charger Address	0x10 <sup>1</sup>
2: COMMAND	Charge current limit	0x3C
	Input current limit	0x3D
3: BYTE LENGTH	Byte length	0x03
4: CONFIG	Store in RAM	0x00
	Store in EEPROM	0x01
5: MSB	Most Significant Byte	See Table 4 & 5
6: LSB	Least Significant Byte	See Table 4 & 5
7: PEC	Packet Error Checking	CRC-8 over the leading 6 bytes including address

Table 3 – Configuration of user commands

The PEC (Packet Error Checking) byte is attached to the end of each message sent over the SMBus communication line. This byte is calculated by the CRC-8 checksum (CRC-8 CCITT) which is calculated according to the following polynomial representation:

$$x^8 + x^2 + x + 1$$

An example code of how to compute the PEC is given in the Appendix A.

Two examples of how to set these commands to the EEPROM are shown below. An example of "Charge Current Limit" is shown in Table 4 and a configuration example of "Input Current Limit" is shown in Table 5.

Addr	Charge Current	Byte Length	Config	2000mA		PEC
0x10	0x3C	0x03	0x01	0x07	0xD0	FF

Table 4 – "Charge Current Limit" of 2A stored in the EEPROM

Addr	Input Current	Byte Length	Config	3000mA		PEC
0x10	0x3D	0x03	0x01	0x0B	0xB8	7E

Table 5 – "Input Current Limit" of 3A stored in the EEPROM

The user commands can be either programmed in the volatile RAM memory or in non-volatile EEPROM memory of the PMM240 using the commands listed in Table 3. The factory default values of the "Charge Current Limit" and

"Input Current Limit" stored in the EEPROM are listed in Table 6.

Parameter	Default setting	Value Range	Step
Charge Current Limit	4096mA	256–6200mA 0x0100-0x1838	128mA
Input Current Limit	8064mA	0–8064mA 0x0000-0x1F80	256mA

Table 6 - Default settings of the current limits

On every startup after power loss (power on reset) the PMM240 reads the user commands out of the EEPROM and loads them into the RAM. The values which are in the RAM are the ones which are valid and directly affect the behavior of the PMM240. By writing new values into the EEPROM, they will only become valid (loaded into RAM) after the next power-on reset.

To temporarily change the current limits, the value within the RAM can be overwritten during operation by using the related command shown in Table 3. This new setting will become valid immediately and directly affect the behavior of the PMM240.

Due to the limitation of erase/write cycles into EEPROM, it is recommended to do temporary changes to user commands only to the RAM. The EEPROM setting should be used to change the initial loaded settings after power loss to fit the user's application needs.

<sup>1</sup> Seven bit address shifted by one to the left. To send a read or a write word command the last bit must be 0 or 1 respectively (0x10)<sub>hex</sub> = (0010 000\_)<sub>2</sub>

**5. Document Revisions**

The revisions of this document are listed in Table 7.

Revision	Valid from	Changes
D	26.Jul.2018	Corrections introduced

Table 7 – List of document revisions

Document subject to technical changes without notice \* Released document  
Copy is not subject to revision service.

© RRC Power Solutions 2018.

Alle Rechte vorbehalten.

Kein Teil dieses Dokumentes darf in irgendwelcher Form wiedergegeben oder unter Verwendung irgendwelcher Übermittlungsmittel übermittelt werden, einschließlich der Herstellung von Photokopien, der Herstellung von Aufnahmen und sämtlicher anderer Formen der Speicherung und Wiedergabe, ohne zuvor die schriftliche Genehmigung der RRC Power Solutions erhalten zu haben.

ALL RIGHTS RESERVED.

NO PART OF THIS DOCUMENT MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY, RECORDING, OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT PRIOR WRITTEN PERMISSION FROM RRC POWER SOLUTIONS



**Appendix A**

The following reference code is intended to provide the user the correct way to generate PEC.

```
/**
 * Bulk buffer CRC8 calculation without a lookup table
 * (size optimized). Generic data types used.
 *
 * @param const void* data
 * @param unsigned size
 * @return unsigned char
 */
static unsigned char crc8(const void *data, unsigned size)
{
    const unsigned char *p = data;
    unsigned crc = 0;
    int i, j;
    for(j = size; j != 0; --j, ++p) {
        crc ^= ((*p) << 8);
        for(i = 8; i > 0; --i) {
            crc = ((crc & 0x8000) ? (crc ^ (0x1070 << 3)) : (crc)) << 1;
        }
    }
    return (unsigned char) (crc >> 8);
}
```

## The Netherlands



Elektrostraat 17  
NL-7483 PG Haaksbergen

T: +31 (0)53 573 33 33  
F: +31 (0)53 573 33 30  
E: [nl@texim-europe.com](mailto:nl@texim-europe.com)

## Belgium



Zuiderlaan 14 bus 10  
B-1731 Zellik

T: +32 (0)2 462 01 00  
F: +32 (0)2 462 01 25  
E: [belgium@texim-europe.com](mailto:belgium@texim-europe.com)

## UK & Ireland



St. Mary's House, Church Lane  
Carlton Le Moorland  
Lincoln LN5 9HS

T: +44 (0)1522 789 555  
F: +44 (0)845 299 22 26  
E: [uk@texim-europe.com](mailto:uk@texim-europe.com)

## Germany North



Bahnhofstrasse 92  
D-25451 Quickborn

T: +49 (0)4106 627 07-0  
F: +49 (0)4106 627 07-20  
E: [germany@texim-europe.com](mailto:germany@texim-europe.com)

## Germany South



Martin-Kollar-Strasse 9  
D-81829 München

T: +49 (0)89 436 086-0  
F: +49 (0)89 436 086-19  
E: [germany@texim-europe.com](mailto:germany@texim-europe.com)

## Austria



Warwitzstrasse 9  
A-5020 Salzburg

T: +43 (0)662 216 026  
F: +43 (0)662 216 026-66  
E: [austria@texim-europe.com](mailto:austria@texim-europe.com)

## Nordic region



Sdr. Jagtvej 12  
DK-2970 Hørsholm

T: +45 88 20 26 30  
F: +45 88 20 26 39  
E: [nordic@texim-europe.com](mailto:nordic@texim-europe.com)

## Italy



Via Matteotti 43  
IT-20864 Agrate Brianza (MB)

T: +39 (0)395 967 226  
F: +39 (0)395 967 226  
E: [italy@texim-europe.com](mailto:italy@texim-europe.com)

## General information



[info@texim-europe.com](mailto:info@texim-europe.com)  
[www.texim-europe.com](http://www.texim-europe.com)