Kevin Evans

This document explains how the PMIC and the IHU are communication down to the byte by byte level

PMIC Interface

Detailed Description of PMIC and IHU Interface

Revision: 1.2.1



Table of Contents

[1 Hardware Layer 2](#_Toc511814398)

[2 Software Layer 3](#_Toc511814399)

[2.1 Subsystem Power Change 3](#_Toc511814400)

[2.2 Voltage Data Request 3](#_Toc511814401)

[2.3 Current Data Request 4](#_Toc511814402)

[2.4 Temperature Data Request 5](#_Toc511814403)

[2.5 Power Channel Status 5](#_Toc511814404)

[3 Example Communication 8](#_Toc511814405)

[3.1 Simple Data Request 8](#_Toc511814406)

[3.2 Repeated Request 8](#_Toc511814407)

# 1 Hardware Layer

The hardware interface between the PMIC and the IHU is I2C. It is a standard implementation of I2C with a clock wire and a data wire. See Wikipedia’s [I2C article](https://en.wikipedia.org/wiki/I%C2%B2C) for details on how it works.

# 2 Software Layer

The IHU is the master and the PMIC is a slave, 8b address 0x0E. The data blocks interchanged between the IHU and PMIC are explained in the following sub-sections. For each command, if a command is sent multiple times before its response is read, when read, the response reflects the most recent command.

## 2.1 Subsystem Power Change

This command is used when a subsystem is desired to be turned off or on. The first byte is the identifier. The second byte is the subsystem. See table below for details. The PMIC replies with the result of the request. See [Code SOP Section 6.4.1](https://github.com/CougsInSpace/Resources/tree/master/StandardOperatingProcedures) for a list of error codes.

|  |  |  |
| --- | --- | --- |
| Index | 0x00 | 0x01 |
| Function | 0x00: Turn off  0x01: Turn on | Subsystem:  0x00: IHU  0x01: IFJR  0x02: ADCS  0x03: ADCS Coils  0x04: Comms  0x05: Comms Amplifier  0x06: Payload 0  0x07: Payload 1  0x08: Payload 2 |

## 2.2 Voltage Data Request

This command is used to read the voltage of various locations on the EPS. The first byte is the identifier. The second byte is the location. See table below for details. The PMIC replies with the voltage as an unsigned 16b integers with 100µV/LSB.

|  |  |  |
| --- | --- | --- |
| Index | 0x00 | 0x01 |
| Function | 0x02 | 0x00: Solar Panel 0  0x01: Solar Panel 1  0x02: Solar Panel 2  0x03: Solar Panel 3  0x04: Solar Panel 4  0x05: Solar Panel 5  0x06: Solar Panel 6  0x07: Solar Panel 7  0x08: Battery A  0x09: Battery B  0x0A: 3.3V Regulator A  0x0B: 3.3V Regulator B |

## 2.3 Current Data Request

This command is used to read the currents of various paths on the EPS. The first byte is the identifier. The second byte is the path. See table below for details. The PMIC replies with the current as a signed 16b integers with 150µA/LSB.

|  |  |  |
| --- | --- | --- |
| Index | 0x00 | 0x01 |
| Function | 0x03 | 0x00: Solar Panel 0  0x01: Solar Panel 1  0x02: Solar Panel 2  0x03: Solar Panel 3  0x04: Solar Panel 4  0x05: Solar Panel 5  0x06: Solar Panel 6  0x07: Solar Panel 7  0x08: Battery A  0x09: Battery B  0x0A: 3.3V Regulator A  0x0B: 3.3V Regulator B  0x0C: PR\_3.3V-0  0x0D: PR\_3.3V-1  0x0E: PR\_3.3V-2  0x0F: PR\_3.3V-3  0x10: PR\_3.3V-4  0x11: PR\_3.3V-5  0x12: PR\_3.3V-6  0x13: PR\_3.3V-7  0x14: PR\_3.3V-8  0x15: PR\_3.3V-9  0x16: PR\_3.3V-10  0x17: PR\_3.3V-11  0x18: PR\_3.3V-12  0x19: PR\_BATT-0  0x1A: PR\_BATT-1  0x1B: PR\_BATT-2  0x1C: PR\_BATT-3  0x1D: PR\_BATT-4  0x1E: PR\_BATT-5  0x1F: PR\_BATT-6  0x20: PV\_3.3V-0  0x21: PV\_3.3V-1  0x22: PV\_3.3V-2  0x23: PV\_3.3V-3  0x24: PR\_BH-0  0x25: PR\_BH-1  0x26: PR\_DEPLOY  0x27: PMIC  0x28: MPPT 0  0x29: MPPT 1  0x2A: MPPT 2  0x2B: MPPT 3  0x2C: MPPT 4  0x2D: MPPT 5  0x2E: MPPT 6  0x2F: MPPT 7 |

## 2.4 Temperature Data Request

This command is used to read the temperatures of various components of the EPS. The first byte is the identifier. The second byte is the component. See table below for details. The PMIC replies with the temperature as a signed 8b integer with 1°C/LSB.

|  |  |  |
| --- | --- | --- |
| Index | 0x00 | 0x01 |
| Function | 0x04 | 0x00: Solar Panel 0  0x01: Solar Panel 1  0x02: Solar Panel 2  0x03: Solar Panel 3  0x04: Solar Panel 4  0x05: Solar Panel 5  0x06: Solar Panel 6  0x07: Solar Panel 7  0x08: Battery A  0x09: Battery B  0x0A: 3.3V Regulator A  0x0B: 3.3V Regulator B  0x27: PMIC  0x28: MPPT 0  0x29: MPPT 1  0x2A: MPPT 2  0x2B: MPPT 3  0x2C: MPPT 4  0x2D: MPPT 5  0x2E: MPPT 6  0x2F: MPPT 7  0x30: PCB -X -Y  0x31: PCB -X +Y  0x32: PCB +X -Y  0x33: PCB +X +Y |

## 2.5 Power Channel Status

This command is used to read the current status of the power channels, send 0x05. The reply is a 56b value. Each bit corresponds to a specific path being on (1) or off (0), see table below. Bit 0 is the LSB.

|  |  |
| --- | --- |
| Bit Index | Result |
| 00 | PR\_3.3V-0 from 3.3V Regulator A |
| 01 | PR\_3.3V-0 from 3.3V Regulator B |
| 02 | PR\_3.3V-1 from 3.3V Regulator A |
| 03 | PR\_3.3V-1 from 3.3V Regulator B |
| 04 | PR\_3.3V-2 from 3.3V Regulator A |
| 05 | PR\_3.3V-2 from 3.3V Regulator B |
| 06 | PR\_3.3V-3 from 3.3V Regulator A |
| 07 | PR\_3.3V-3 from 3.3V Regulator B |
| 08 | PR\_3.3V-4 from 3.3V Regulator A |
| 09 | PR\_3.3V-4 from 3.3V Regulator B |
| 10 | PR\_3.3V-5 from 3.3V Regulator A |
| 11 | PR\_3.3V-5 from 3.3V Regulator B |
| 12 | PR\_3.3V-6 from 3.3V Regulator A |
| 13 | PR\_3.3V-6 from 3.3V Regulator B |
| 14 | PR\_3.3V-7 from 3.3V Regulator A |
| 15 | PR\_3.3V-7 from 3.3V Regulator B |
| 16 | PR\_3.3V-8 from 3.3V Regulator A |
| 17 | PR\_3.3V-8 from 3.3V Regulator B |
| 18 | PR\_3.3V-9 from 3.3V Regulator A |
| 19 | PR\_3.3V-9 from 3.3V Regulator B |
| 20 | PR\_3.3V-10 from 3.3V Regulator A |
| 21 | PR\_3.3V-10 from 3.3V Regulator B |
| 22 | PR\_3.3V-11 from 3.3V Regulator A |
| 23 | PR\_3.3V-11 from 3.3V Regulator B |
| 24 | PR\_3.3V-12 from 3.3V Regulator A |
| 25 | PR\_3.3V-12 from 3.3V Regulator B |
| 26 | PR\_BATT-0 from VBATT-A |
| 27 | PR\_BATT-0 from VBATT-B |
| 28 | PR\_BATT-1 from VBATT-A |
| 29 | PR\_BATT-1 from VBATT-B |
| 30 | PR\_BATT-2 from VBATT-A |
| 31 | PR\_BATT-2 from VBATT-B |
| 32 | PR\_BATT-3 from VBATT-A |
| 33 | PR\_BATT-3 from VBATT-B |
| 34 | PR\_BATT-4 from VBATT-A |
| 35 | PR\_BATT-4 from VBATT-B |
| 36 | PR\_BATT-5 from VBATT-A |
| 37 | PR\_BATT-5 from VBATT-B |
| 38 | PR\_BATT-6 from VBATT-A |
| 39 | PR\_BATT-6 from VBATT-B |
| 40 | PV\_3.3V-0 from 3.3V Regulator A |
| 41 | PV\_3.3V-0 from 3.3V Regulator B |
| 42 | PV\_3.3V-1 from 3.3V Regulator A |
| 43 | PV\_3.3V-1 from 3.3V Regulator B |
| 44 | PV\_3.3V-2 from 3.3V Regulator A |
| 45 | PV\_3.3V-2 from 3.3V Regulator B |
| 46 | PV\_3.3V-3 from 3.3V Regulator A |
| 47 | PV\_3.3V-3 from 3.3V Regulator B |
| 48 | PR\_BH-0 from VBATT-A |
| 49 | PR\_BH-0 from VBATT-B |
| 50 | PR\_BH-1 from VBATT-A |
| 51 | PR\_BH-1 from VBATT-B |
| 52 | PR\_DEPLOY from VBATT-A |
| 53 | PR\_DEPLOY from VBATT-B |

## 2.5 Solar Panel Channel Status

This command is used to read the current status of the solar panel channels, send 0x06. The reply is a 16b value. Each bit corresponds to a specific path being on (1) or off (0), see table below. Bit 0 is the LSB.

|  |  |
| --- | --- |
| Bit Index | Result |
| 00 | PV\_IN-0 to VIN-A |
| 01 | PV\_IN-0 to VIN-B |
| 02 | PV\_IN-1 to VIN-A |
| 03 | PV\_IN-1 to VIN-B |
| 04 | PV\_IN-2 to VIN-A |
| 05 | PV\_IN-2 to VIN-B |
| 06 | PV\_IN-3 to VIN-A |
| 07 | PV\_IN-3 to VIN-B |
| 08 | PV\_IN-4 to VIN-A |
| 09 | PV\_IN-4 to VIN-A |
| 10 | PV\_IN-5 to VIN-A |
| 11 | PV\_IN-5 to VIN-B |
| 12 | PV\_IN-6 to VIN-A |
| 13 | PV\_IN-6 to VIN-B |
| 14 | PV\_IN-7 to VIN-A |
| 15 | PV\_IN-7 to VIN-B |

# 3 Example Communication

## 3.1 Simple Data Request

In this example, the IHU wants the voltage of solar panel 2.

IHU: [0x0E]0x0203 [PMIC write] voltage? solar panel 2

PMIC reads the voltage and stores the value into its buffer

IHU: [0x0F] [PMIC read]

PMIC: 0xBFEC (49132 \* 100µV = 4.9132 V)

## 3.2 Repeated Request

In this example, the IHU wants the voltage of solar panel 2 but asks for the 3.3V-A regulator first.

IHU: [0x0E]0x020A [PMIC write] voltage? 3.3V-A regulator

PMIC reads the voltage and stores the value into its buffer

IHU: [0x0E]0x02032 [PMIC write] voltage? solar panel 2

PMIC reads the voltage and stores the value into its buffer. The voltage of the 5.0V rail is discarded.

IHU: [0x0F] [PMIC read]

PMIC: 0xBFEC (49132 \* 100µV = 4.9132 V)