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: 1.1.2

Bradley Davis

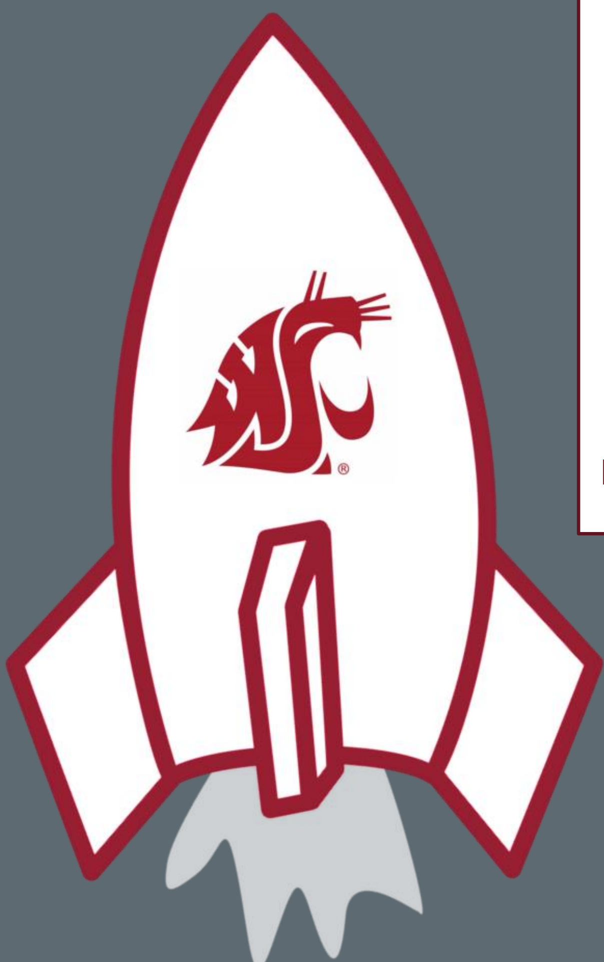

Table of Contents

1 Hardware Layer

The hardware interface between the PMIC and the IHU is I^2C. It is a standard implementation of I^2C with a clock wire and a data wire. See Wikipedia's I^2C article for details on how it works.

When this document reference multiple components that exist on both boards of the EPS, the first half of instances are referenced to the hardware selected primary board and the second half reference the secondary board. The software defined roles of primary and secondary do not affect the reference in this document.

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 1 for details. The PMIC replies with the result of the request. This reply is read at `0x80`. See table 2 for a list of results.

Table 1: Subsystem Power Change Request

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

Table 2: Subsystem Power Change Reply

Value	Result
0x00	Change Success
0x01	Failed: Limited power
0x02	Failed: Overcurrent

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 3 for details. The PMIC replies with the voltage as an unsigned 16b integers with 100µV/LSB. This reply is read at `0x82`.

Table 3: Voltage Data Request

Index	0x00	0x01
Function	0x02	0x00: Solar Panel 0 0x01: Solar Panel 1 0x02: Solar Panel 2 0x03: Solar Panel 3 0x04: Battery 0 0x05: Battery 1 0x06: 3.3V Rail 0x07: 5.0V Rail

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 4 for details. The PMIC replies with the current as a signed 24b integers with 100μA/LSB. This reply is read at `0x83`.

Table 4: Current Data Request

Index	0x00	0x01
Function	0x03	0x00: Solar Panel 0 In 0x01: Solar Panel 1 In 0x02: Solar Panel 2 In 0x03: Solar Panel 3 In 0x04: Battery 0 0x05: Battery 1 0x06: 3.3V Rail 0x07: 5.0V Rail 0x08: MPPT 0 Out 0x09: MPPT 1 Out 0x0A: MPPT 2 Out 0x0B: MPPT 3 Out 0x0C: 3.3V Channel 0 0x0D: 3.3V Channel 1 0x0E: 3.3V Channel 2 0x0F: 3.3V Channel 3 0x10: 3.3V Channel 4 0x11: 3.3V Channel 5 0x12: 5.0V Channel 0 0x13: 5.0V Channel 1 0x14: Unregulated Channel 0 0x15: Unregulated Channel 1 0x16: Unregulated Channel 2 0x17: Unregulated Channel 3 0x18: Unregulated Channel 4 0x19: Unregulated Channel 5 0x1A: Battery Heater 0 0x1B: Battery Heater 1

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 5 for details. The PMIC replies with the temperature as a signed 8b integer with 1°C/LSB. This reply is read at `0x84`.

Table 5: Temperature Data Request

Index	0x00	0x01
Function	0x03	0x04: Battery 0 0x05: Battery 1 0x06: 3.3V Rail Regulator 0

		0x07: 5.0V Rail Regulator 0 0x08: MPPT 0 0x09: MPPT 1 0x0A: MPPT 2 0x0B: MPPT 3 0x1C: 3.3V Rail Regulator 1 0x1D: 5.0V Rail Regulator 1 0x1E: PMIC 0 0x1F: PMIC 1 0x20: PCB +X+Y 0 0x21: PCB +X+Y 1 0x22: PCB +X-Y 0 0x23: PCB +X-Y 1 0x24: PCB -X+Y 0 0x25: PCB -X+Y 1 0x26: PCB -X-Y 0 0x27: PCB -X-Y 1

2.5 Power Channel Status

This command is used to read the current status of the power channels. The reply is a 16b value. Each bit corresponds to a specific channel being on (`1`) or off (`0`), see table 6. This status is read at `0x85`.

Table 6: Power Channels

Value	Result
0x0001	3.3V Channel 0
0x0002	3.3V Channel 1
0x0004	3.3V Channel 2
0x0008	3.3V Channel 3
0x0010	3.3V Channel 4
0x0020	3.3V Channel 5
0x0040	5.0V Channel 0
0x0080	5.0V Channel 1
0x0100	Unregulated Channel 0
0x0200	Unregulated Channel 1
0x0400	Unregulated Channel 2
0x0800	Unregulated Channel 3
0x1000	Unregulated Channel 4
0x2000	Unregulated Channel 5

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) (0x82)      (PMIC read) (voltage buffer)
PMIC: (0x00BFEC)        (4.9132 V)
```

3.2 Repeated Request

In this example, the IHU wants the voltage of solar panel 2 but asks for the 5.0V rail first.

```
IHU: (0x0E) (0x0207)    (PMIC write) (voltage? 5.0V rail)
      PMIC reads the voltage and stores the value into its buffer
IHU: (0x0E) (0x0203)    (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) (0x82)      (PMIC read) (voltage buffer)
PMIC: (0x00BFEC)        (49132 * 100µV = 4.9132 V)
```