**PSU Software interface**

A Christodoulou and N. Attree

|  |  |  |
| --- | --- | --- |
| Date | Updated Reference Number | change |
|  |  |  |
| 18/02/2009 | PLM-PSU-SoftInter-318-1 | first version issued |
| 15/07/2010 | PLM-PSU-SoftInter-318-2 | Added all commands (not just ADC). Removed satellite modes section. |
|  |  |  |
|  |  |  |
|  |  |  |

**Data**

PSU telemetry can be stored on the SD card by OBDH for transmission to the surface. Data includes panel voltage, current and temperature, and battery voltage, current, current direction and temperature.

# I2C Bus Commands

The I2C bus is used for all communications between the master (the microcomputer of the CubeSat or a PC to which the I2C bus is connected) and the telemetry and telecommand node (TTC node), which is configured as a slave.

The microcontroller of the TTC node has already been programmed. It accepts two types of messages from the master, a write which initiates an action to be performed by the TTC node and a read command that tells the TTC node to send the result of the action back to the master. All commands sent from the master to the slave start with the 7 bit slave address which is set to be 0x01. Figure 1 shows the sequence of bytes sent by the master to the slave (in gray) and the reply bytes, sent by the slave to the master.

The maximum data rate supported on the I2C bus is 400kbit. ***Figure 1.*** *Message format.*

A

7 Bit Node Address

A

nW

Command Value

Command Type

A

A

Sr

7 Bit Node Address

A

R

Reply Byte 0

Reply Byte 1

A

N

P

Write Message

Read Message

Write Addressing Byte

Write Bytes

Read Addressing Byte

Reply Response Bytes

S – Start Condition R – Read Command

nW – Write Command N – Not acknowledged

A – Acknowledged Sr – Repeat Start Condition

The command type determines what type of action is to be performed, from requesting telemetry data to checking for error messages, and then the command value determines the precise action. A list of command types is shown in the table below.

If a read message is sent without a preceding write message 0xF000 is returned.

|  |  |  |  |
| --- | --- | --- | --- |
| **Command Types** | | **Command Value Range** | **Description** |
| **Decimal** | **Name** | **Decimal** |
| 0 | ADC | 0-31 | Reads ADC (telemetry) channel from 0-31. See table 2 below. |
| 1 | Status | N/A | Request status bytes. See table 3 below. |
| 2 | GPIO | 0-7 | Selects General Purpose Input Output pin to pulse. |
| 3 | Pulse | 0-255 | Controls length of GPIO pulse. |
| 4 | Version | N/A | Requests firmware version of the board. |
| 5 | Heater | 0-1 | Turns battery heater off. |
| 6 | GPIO Status | N/A | Requests GPIO and battery heater status. |
| 128 | Watchdog | N/A | Causes soft reset of the micro. |

***Table 1****. Command types*

**ADC**

At this stage, the team is focusing on the ADC Commands and Replies which are used for telemetry. The sequence that is followed for this process is to define the command type, which in this case is the decimal 0. Once the command type has been sent, the command value is then sent that defines which ADC channel will be monitored. These two bytes are write bytes.

The next step is to send a real command to the slave. The slave replies with two reply bytes. The actual reply is a 10 bit value with the lower two bits in reply byte 1 and the lower 8 in reply byte 0. The upper 6 bits of reply byte 1 are just zeros.

The delay between the commands should be 1ms. In case of a problem, set it to 2ms.

Below there is a list of the Command Type and Command Value bytes available for the ADC channels. To use these commands the four bytes of hexadecimal returned must be converted to decimal and substituted into the calibration equations for (ADC), bearing in mind that the first six bits of the first byte (byte 1) are all zero.

A spreadsheet called TTCconversion has been written and will be uploaded to the wiki to aid with this. To use it enter the last three Hex digits returned from the EPS board into the ‘Hex Output’ column and the physical measurement will be displayed in the ‘Calculated Value’ column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **READ ADC CHANNELS** | | | | |
| **SIGNAL** | **Byte SEQUENCE** | | **CALIBRATION EQUATION** | **UNIT** |
| **Hexadecimal** | **Binary** |
| Panel Y1 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0x0 | 00000000 |
| Panel Y1 Current | 0x1 | 0000001 | I = (-0.486\*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0x1 | 00000001 |
| Panel Y1 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | DegC |
|  | 0x0 | 00000000 |
|  | 0x2 | 00000010 |
| Panel X2 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0x3 | 00000011 |
| Panel X2 Current | 0x1 | 0000001 | I = (-0.486\*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0x4 | 00000100 |
| Panel X2 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | DegC |
|  | 0x0 | 00000000 |
|  | 0x5 | 00000101 |
| Panel X1 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0x6 | 00000110 |
| Panel X1 Current | 0x1 | 0000001 | I = (-0.486\*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0x7 | 00000111 |
| Panel X1 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | DegC |
|  | 0x0 | 00000000 |
|  | 0x8 | 00001000 |
| Panel Z1 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0x9 | 00001001 |
| Panel Z1 Current | 0x1 | 0000001 | I = (-0.486\*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0xA | 00001010 |
| Panel Z1 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | DegC |
|  | 0x0 | 00000000 |
|  | 0xB | 00001011 |
| Panel Y2 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0xC | 00001100 |
| Panel Y2 Current | 0x1 | 0000001 | I = (-0.486\*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0xD | 00001101 |
| Panel Y2 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | DegC |
|  | 0x0 | 00000000 |
|  | 0xE | 00001110 |
| Panel Z2 Voltage | 0x1 | 0000001 | V = (-0.009\*ADC) + 8.703 | V |
|  | 0x0 | 00000000 |
|  | 0xF | 00001111 |
| - (gnd) | 0x1 | 0000001 |  |  |
|  | 0x0 | 00000000 |  |  |
|  | 0x10 | 00010000 |  |  |
| - (gnd) | 0x1 | 0000001 |  |  |
|  | 0x0 | 00000000 |  |  |
|  | 0x11 | 00010001 |  |  |
| Battery 2 Temp | 0x1 | 0000001 | T = (-0.163\*ADC) + 110.7 | DegC |
|  | 0x0 | 00000000 |
|  | 0x12 | 00010010 |
| Battery 2 Voltage | 0x1 | 0000001 | V = (-0.01\*ADC) + 9.75 | V |
|  | 0x0 | 00000000 |
|  | 0x13 | 00010011 |
| Cell 2 Voltage | 0x1 | 0000001 | V = (0.00488\*ADC) - 0.30256 | V |
|  | 0x0 | 00000000 |
|  | 0x14 | 00010100 |
| Battery 2 Current Direction | 0x1 | 0000001 | High - Bat Charging |  |
| 0x0 | 00000000 | Low - Bat Discharging |  |
|  | 0x15 | 00010101 |  |  |
| Battery 2 Current | 0x1 | 0000001 | I = (-3.473\*ADC) + 3256.644 | mA |
|  | 0x0 | 00000000 |
|  | 0x16 | 00010110 |
| Battery 1 Temp | 0x1 | 0000001 | T = (-0.163\*ADC) + 110.7 | DegC |
|  | 0x0 | 00000000 |
|  | 0x17 | 00010111 |
| Battery 1 Voltage | 0x1 | 0000001 | V = (-0.01\*ADC) + 9.75 | V |
|  | 0x0 | 00000000 |
|  | 0x18 | 00011000 |
| Cell 1 Voltage | 0x1 | 0000001 | V = (0.00488\*ADC) - 0.30256 | V |
|  | 0x0 | 00000000 |
|  | 0x19 | 00011001 |
| |  | | --- | | 5 Volt Bus Current | |  | | 0x1 | 0000001 | I =(-1.2009\*D28)+1.239284 | mA |
|  | 0x0 | 00000000 |  |  |
|  | 0x1A | 00011010 |  |  |
| 3.3 Volt Bus Current | 0x1 | 0000001 | I  = (-1.30099\*D29)+1334.687 | mA |
|  | 0x0 | 00000000 |  |  |
|  | 0x1B | 00011011 |  |  |
| Battery 1 Current Direction | 0x1 | 0000001 | High - Bat Charging |  |
| 0x0 | 00000000 | Low - Bat Discharging |  |
|  | 0x1C | 00011100 |  |  |
| Battery 1 Current | 0x1 | 0000001 | I = (3/(1023\*0.94)) \* ADC | mA |
|  | 0x0 | 00000000 |
|  | 0x1D | 00011101 |
| Panel Z2 Temp | 0x1 | 0000001 | T = (-0.1619\*ADC) + 110.119 | mA |
|  | 0x0 | 00000000 |
|  | 0x1E | 00011110 |
| Panel Z2 Current | 0x1 | 0000001 | I = (-0.486 \*ADC) + 502.524 | mA |
|  | 0x0 | 00000000 |
|  | 0x1F | 00011111 |

*Table 2 ADC monitoring commands and calibration equations*

**Status**

Sending a command type bit of 1 requests the TTC node status from the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte** | **Bit** | **Name** | **Description** |
| 1 | 0 | I2C Error | 1= An I2C error has occurred. Error flags are cleared |
|  |  | 0 = No I2C errors |
| 1 | I2CWCOL | 1 = A write collision has occurred due to the message transmission |
|  |  | 0 = No write collisions have occurred |
| 2 | I2CSSPOV | 1 = A receive message overflow has occurred |
|  |  | 0 = No overflows have occurred |
| 3 | RxMessageToLong | 1 = A received message has more than 2 bytes |
|  |  | 0 = All received messages were the correct length |
| 4-7 | Not Used | Always reads 0 |
| 0 | 0 | UnknownCommandType | 1 = Unknown command type was received, no action was taken |
|  |  | 0 = All command types received have been successfully decoded |
| 1 | UnknownCommandValue | 1 = An out of range command value has been received |
|  |  | 0 = All command values have been in range |
| 2 | ADCNotReady | 1= The ADC was not ready to be read |
|  |  | 0 = No ACD error |
| 3 | Not Used | Always reads 0 |
| 4 | OSFIF | 1 = External oscillator failure (interface PIC bit) |
|  |  | 0 = External oscillator running |
| 5 | WatchDogReset | 1 = Last reset was caused by the watchdog timing out |
|  |  | 0 = Watchdog did not cause reset |
| 6 | nPOR | 1 = Power On did not cause reset |
|  |  | 0 = Power On caused the last reset |
| 7 | nBOR | 1 = A brown out did not caused the last reset |
|  |  |  | 0 = A brown out caused the last reset |

***Table 3****. Status bits.*

**GPIO**

There are three General Purpose Input Output signals available. Using the command that pulses them, they can be set to HIGH or LOW for a given period. The defined pulse length is 32.768ms. The pulse length can be changed using the command that defines the length of the GPIO pulse. There are no reply bytes for these sets of commands. The status of each of the pins (high or low) can be checked using GPIO status.

For more information see the Clyde Space manual.

**Firmware**

The firmware version command returns two reply bytes with byte 1 containing the major version number and byte 0 containing the minor version number.

**Watchdog**

This command prevents the board’s internal watchdog from being reset, causing a soft reboot of the TTC node firmware. No command value is required and if a read command is sent 0xF000 will be returned.