# . OCTOPUS Internal CAN Description

Table of Contents

[OCTOPUS Internal CAN Description 1](#_Toc96345595)

[Message Frame 3](#_Toc96345596)

[MC Register List 4](#_Toc96345597)

[Auto-discovery 11](#_Toc96345598)

[SC-MCU Register List 12](#_Toc96345599)

## Message Frame

Table 1 - CAN frame

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SoF | ID pt.1 | SRR | IDE | ID pt.2 | RTR | reserved | Data Len. | Data | CRC | CRC delim | ACK | ACK delim | EOFmodule ala |
| # of bits | 1 | 11 | 1 | 1 | 18 | 1 | 2 | 4 | 64 | 15 | 1 | 1 | 1 | 7 |

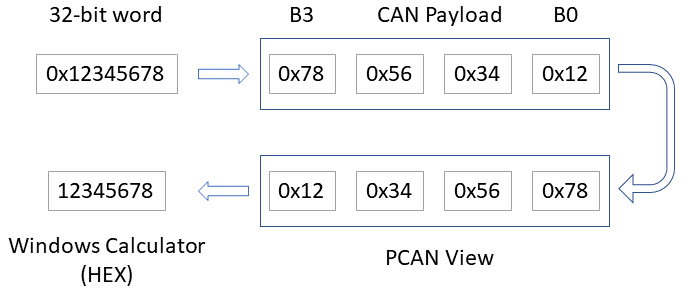
The CAN ID is then:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | CAN ID pt.1 | | | | | |  | **PRIO** | **DST** | | **SRC** | | | **bit** | 28 | 27-23 | | 22-18 | | | **Content** | 0 = High | 0 | Broadcast | 0 | unconfigured | | 1 = Low | 1-29 | MC | 1-29 | MC | |  | 30 | SC MCU | 30 | SC MCU | |  | 31 | SC MPU | 31 | SC MPU | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | CAN ID pt.2 | | | | |  | **reserved** | **ParamID** | **reserved** | **Index** | | **bit** | 17 | 16-9 | 8 | 7-0 | | **Content** |  | See  *Register List*  chapter |  | 0 = END | |  |  | 1-255 = Index | |  |  |  | |  |  |  | |

Table 2 - Internal CAN frame

\* MCU listens to specific CAN messages to the MPU (particularly, alarms) despite not being directly addressed. Separate addressing maintained to retain potential future implementation.

Payload is handled as BIG endian:



## MC Register List

| ID | Name | Values | Report intvl |
| --- | --- | --- | --- |
| 0 | Device Statuses | Byte (READONLY)  **MC Status** 2^0 = State error 2^1 = Fail Safe Error 2^2 = Firmware failed 2^3 = Flag reset | 2s |
| Byte (READONLY)  **FPGA Status**  Bitwise flags:  2^0 = Under cell voltage  2^1 = Over cell voltage  2^2 = Under cell temp  2^3 = Over cell temp  2^4 = Hardware fault |
| Byte (READONLY)  **Active states**  2^0 = Balancing state 2^1 = Simulation state 2^2 = Firmware updating state |
| 1 | Internal times | UInt32 (READWRITE)  **System Time**  Linux Epoch Timestamp |  |
| UInt32 (READONLY)  **Uptime**  Uptime seconds |
| 2 | Warnings/Alarms | UInt32 (READONLY)  **Warnings**  Bitwise flags:  2^0 = Cell V High Warn  2^1 = Cell V Low Warn  2^2 = Cell T High Warn  2^3 = Cell T Low Warn  2^4 = Ext 1 T High Warn  2^5 = Ext 1 T Low Warn  2^6 = Ext 2 T High Warn  2^7 = Ext 2 T Low Warn  2^8 = Balancing T Low Warn  2^9 = Balancing T High Warn  2^10 = RTC fail  2^11 = External Flash Fail  2^12 = HVIL Detection  2^13 = Cooling fan fail  2^14 = Voltage fuse 1 (3.0V) active  2^15 = Fail Safe Signal Active  2^16 = Temperature sensor out of spec.  2^17 = Simulation mode denied/timeout[[1]](#footnote-2) | 2s |
| UInt32 (READONLY)  **Alarms**  Bitwise flags:  2^0 = Cell V High Error  2^1 = Cell V Low Error  2^2 = Cell T High Error  2^3 = Cell T Low Error  2^4 = Ext 1 T High Error  2^5 = Ext 1 T Low Error  2^6 = Ext 2 T High Error  2^7 = Ext 2 T Low Error  2^8 = LTC Self-test Failure  2^9 = Current sensor over/under spec  2^10 = Uncalibrated Current Sensor  2^11 = Cell Open Wire  2^12 = Leak Detection  2^13 = FPGA Status not OK  2^14 = Voltage fuse 2 (2.7V) active |
| 3 | Cell 1-4 Voltages | UInt16 (READONLY)  **Cell1 mV** |  |
| UInt16 (READONLY)  **Cell2 mV** |
| UInt16 (READONLY)  **Cell3 mV** |
| UInt16 (READONLY)  **Cell4 mV** |
| 4-7 | … |  |  |
| 8 | Cell 21-24 Voltages | UInt16 (READONLY)  **Cell21 mV** |  |
| UInt16:  **Cell22 mV** |
| UInt16 (READONLY)  **Cell23 mV** |
| UInt16 (READONLY)  **Cell24 mV** |
| 9 | Temperature Monitoring Channels 1-8 | SByte (READONLY)  **Cell Ch1 °C** |  |
| SByte (READONLY)  **Cell Ch 2 °C** |
| SByte (READONLY)  **Cell Ch 3 °C** |
| SByte (READONLY)  **Cell Ch 4 °C** |
| SByte (READONLY)  **Cell Ch 5 °C** |
| SByte (READONLY)  **Cell Ch 6 °C** |
| SByte (READONLY)  **Cell Ch 7 °C** |
| SByte (READONLY)  **Cell Ch 8 °C** |
| 10 | Temperature Auxiliary and Balancing Channels | SByte (READONLY)  **Bal Ch1 °C** |  |
| SByte (READONLY)  **Bal Ch2 °C** |
| SByte (READONLY)  **Ext Ch1 °C** |
| SByte (READONLY)  **Ext Ch2 °C** |
| 11 | Module Current | Int32 (READONLY)  **mA** |  |
| 12 | Module Voltage | UInt32 (READONLY)  **mV** |  |
| 13 | Balancing status | UInt32 (READONLY)  **Balancing status flags**  Bitwise flags:  2^0 = Cell 1 Balancing enabled  2^1 = Cell 2 Balancing enabled  2^2 = Cell 3 Balancing enabled  2^3 = Cell 4 Balancing enabled  2^4 = Cell 5 Balancing enabled  2^5 = Cell 6 Balancing enabled  2^6 = Cell 7 Balancing enabled  2^7 = Cell 8 Balancing enabled  2^8 = Cell 9 Balancing enabled  2^9 = Cell 10 Balancing enabled  2^10 = Cell 11 Balancing enabled  2^11 = Cell 12 Balancing enabled  2^12 = Cell 13 Balancing enabled  2^13 = Cell 14 Balancing enabled  2^14 = Cell 15 Balancing enabled  2^15 = Cell 16 Balancing enabled  2^16 = Cell 17 Balancing enabled  2^17 = Cell 18 Balancing enabled  2^18 = Cell 19 Balancing enabled  2^19 = Cell 20 Balancing enabled  2^20 = Cell 21 Balancing enabled  2^21 = Cell 22 Balancing enabled  2^22 = Cell 23 Balancing enabled  2^23 = Cell 24 Balancing enabled |  |
| 14 | Detected Open Wires | UInt32 (READONLY)  **Cell Open Wires**  Bitwise flags:  2^0 = Cell 1 Open Wire  2^1 = Cell 2 Open Wire  2^2 = Cell 3 Open Wire  2^3 = Cell 4 Open Wire  2^4 = Cell 5 Open Wire  2^5 = Cell 6 Open Wire  2^6 = Cell 7 Open Wire  2^7 = Cell 8 Open Wire  2^8 = Cell 9 Open Wire  2^9 = Cell 10 Open Wire  2^10 = Cell 11 Open Wire  2^11 = Cell 12 Open Wire  2^12 = Cell 13 Open Wire  2^13 = Cell 14 Open Wire  2^14 = Cell 15 Open Wire  2^15 = Cell 16 Open Wire  2^16 = Cell 17 Open Wire  2^17 = Cell 18 Open Wire  2^18 = Cell 19 Open Wire  2^19 = Cell 20 Open Wire  2^20 = Cell 21 Open Wire  2^21 = Cell 22 Open Wire  2^22 = Cell 23 Open Wire  2^23 = Cell 24 Open Wire |  |
| 15 | HW Serial Number Low | UInt64 (READONLY)  **HW Serial Number Low** |  |
| 16 | HW Serial Number High | UInt64 (READONLY)  **HW Serial Number High** |  |
| 17 | Production Serial | String[64] (READWRITE)  **Production Serial** |  |
| 18 | Config Number of Cells | UInt8 (READWRITE)  **Config Number of Battery Cells** |  |
| 19 | Config Cell Voltage Thresholds | UInt16 (READWRITE)  **Config Cell V High Warn Ths** |  |
| UInt16 (READWRITE)  **Config Cell V Low Warn Ths** |
| UInt16 (READWRITE)  **Config Cell V High Error Ths** |
| UInt16 (READWRITE)  **Config Cell V Low Error Ths** |
| 20 | Config Temperature Warn Thresholds | SByte (READWRITE)  **Config Cell Temp High Ths** |  |
| SByte (READWRITE)  **Config Cell Temp Low Ths** |
| SByte (READWRITE)  **Config Bal Temp High Ths** |
| SByte (READWRITE)  **Config Bal Temp Low Ths** |
| SByte (READWRITE)  **Config Ext1 Temp High Ths** |
| SByte (READWRITE)  **Config Ext1 Temp Low Ths** |
| SByte (READWRITE)  **Config Ext2 Temp High Ths** |
| SByte (READWRITE)  **Config Ext2 Temp Low Ths** |
| 21 | Config Temperature Alarm Thresholds | SByte (READWRITE)  **Config Cell Temp High Ths** |  |
| SByte (READWRITE)  **Config Cell Temp Low Ths** |
| SByte (READWRITE)  **Config Ext1 Temp High Ths** |
| SByte (READWRITE)  **Config Ext1 Temp Low Ths** |
| SByte (READWRITE)  **Config Ext2 Temp High Ths** |
| SByte (READWRITE)  **Config Ext2 Temp Low Ths** |
| 22 | Config Low Voltage Fuses | UInt8 (READWRITE)  **Low Voltage Fuses**  Bitwise flags:  2^0 = Fuse 1 (3V) (True to reset)  2^1 = Fuse 2 (2.7V) (True to reset) |  |
| 23 | Config External temp. sensors | UInt8 (READWRITE)  **External temperature sensors**  Bitwise flags:  2^0 = Enable external temp. 1  2^1 = Enable external temp. 2  2^3 = Enable alarm on ext. temp. 1  2^4 = Enable alarm on ext. temp. 2 |  |
| 24 | Config Cooling Fans | UInt8 (READWRITE)  **Cooling fans**  Bitwise flags:  2^0 = Activate fans  2^1 = Config cooling fan 1  2^2 = Config cooling fan 2  2^3 = Config cooling fan 3  2^4 = Config cooling fan 4 |  |
| 25 | Config Current sensor Zero | Int32 (READONLY)  **Current sensor Zero**  If payload is 0UL, automatic offset calculation will start. Otherwise, the sent value will be registered as the offset value, if EEPROM is busy, the configuration rejected register will be updated. |  |
| 26 | Config Current sensor Delta | Int32 (READONLY)  **Current sensor Delta**  If payload is 0UL, automatic delta calculation will start, offset should be calculated before and calibration current should have been sent. Otherwise, the sent value will be registered as the delta value, if EEPROM is busy, the configuration rejected register will be updated. |  |
| 27 | Hardware Revision | Char (READONLY)  **Hardware Revision** |  |
| 28 | FW Version Numbers  *(Reset firmware failed status)* | UInt32 (READONLY)  **Current FW**  U8BIT Major  U8BIT Minor  U16BIT Build |  |
| UInt32 (READONLY)  **Backup FW**  U8BIT Major  U8BIT Minor  U16BIT Build |
| 29 | Valuelog Size | UInt16 (READONLY)  **Valuelog Size**  Size of Flash pages written, index of the last written page is returned. |  |
| 30 | Valuelog Read Entry | UInt16 (READONLY)  **(REQUESTED WITH PAYLOAD)**  Payload: Valuelog index requested (UInt16) Response: 32 messages, for 4 log entries |  |
| 31 | Command Reboot FW | Byte (WRITEONLY)  **Reboot FW**  Reboot from one of available FW |  |
| 32 | Command Reset Config to factory | Byte (WRITEONLY)  **Reset to Config factory**  Revert configurations to factory (**for future dev**) |  |
| 33 | Command Start Balancing | U32int (WRITEONLY)  **Start Balancing**  24BIT Cell #  Bitwise flags:  2^0 = Cell 1  2^1 = Cell 2  2^2 = Cell 3  2^3 = Cell 4  2^4 = Cell 5  2^5 = Cell 6  2^6 = Cell 7  2^7 = Cell 8  2^8 = Cell 9  2^9 = Cell 10  2^10 = Cell 11  2^11 = Cell 12  2^12 = Cell 13  2^13 = Cell 14  2^14 = Cell 15  2^15 = Cell 16  2^16 = Cell 17  2^17 = Cell 18  2^18 = Cell 19  2^19 = Cell 20  2^20 = Cell 21  2^21 = Cell 22  2^22 = Cell 23  2^23 = Cell 24  6BIT Repetitions [0-63] [times]  2BIT On-Time [0-3] [sec]  0 = 1 sec  1 = 2 sec  2 = 3 sec  3 = 5 sec |  |
| 34 | Command Current Sensor Calibration | Int32 (WRITEONLY)  **Current Sensor Calibration**  Starts calibration of sensor  U32 7 current [mA] |  |
| 35 | Debug Enable Serial Stream | Boolean (READWRITE)  **Enable Serial Stream** |  |
| 36 | Debug Serial Stream | String [64](REPORT\_ONLY)  **Serial Stream** |  |
| 37 | Debug Serial Command | String (WRITEONLY)  **Serial Command** |  |
| 38 | Set CAN ID | U8BIT (READWRITE)  **CAN ID**  U8BIT  **XOR(CAN\_ID, PRIVATE\_KEY)**  Send ID 0 to reset the module controller to unconfigured ID mode |  |
| 39 | Module controller reset cause | UInt8 (READONLY)  **Reset cause number:**  0 = Power cycle   1. = Illegal state 2. = Watchdog timer interrupt 3. = Firmware upgrade |  |
| 40 | Module SoH Calculator Params | UInt8  **Reserved** |  |
| Int8  **Avg Temperature** |  |
| UInt16  **Avg Temperature Counter** |  |
| Int16  **SoC** |  |
| Int16  **SoH** |  |
| 41 | Configuration rejected | UInt8 (READONLY)  **Configuration index** |  |
| 42 | Simulation mode configuration | Byte (READWRITE)  **Simulation Mode State**  1 = Inactive  2 = Active |  |
| 43 | Temperature Monitoring Channels 9-12 | SByte (READONLY)  **Cell Ch 9 °C** |  |
| SByte (READONLY)  **Cell Ch 10 °C** |  |
| SByte (READONLY)  **Cell Ch 11 °C** |  |
| SByte (READONLY)  **Cell Ch 12 °C** |  |
| 44 | Config Number of Monitoring Temperature Channels | UInt8 (READWRITE)  **Config Number of Temperature Channels** |  |

Table 3 – MC CAN registers

**Notes:**

* To write portions of registers, set the remaining part with the maximum number for the specific format. For example, to set Cell1 voltage as 100mV, in register 3 write the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Cell 1-4 Voltages | | | |
| UInt16: Cell1 mV | UInt16: Cell2 mV | UInt16: Cell3 mV | UInt16: Cell4 mV |
| 100 | 65535 | 65535 | 65535 |

* Writing into a value register is only possible if Simulation Mode State is “Active”.

## Auto-discovery

The auto-discovery and ID assignment scheme which should guarantee each MC is uniquely identified by the SC and assigned its own CAN ID, by using the 128-bit HW Serial Number embedded in the µC.

The principle is initiated by the MC booting up with no CAN ID when the 24V is connected, periodically (e.g. 1Hz) sending the bottom half of its SN to the SC until it is assigned (it receives) an addressed message containing its own SN, at which point it will consider the addressed CAN ID to be its own CAN ID.

The very unlikely case where two bottom halves of the SN match is covered by - after assigning the non-duplicate MCs their IDs - the SC broadcasting a request for the top halves of the SNs. Any modules that do not yet have an ID will now be uniquely identifiable (as guaranteed by the µC’s specification) and assigned a CAN ID.

This scheme is required due to the maximum payload of a CAN message being 8 bytes, or 64 bits, meaning not the entire SN can be communicated in a single message, making concatenation of two different SNs difficult to avoid when sending both halves as sequential messages.

The final step is to exchange the production serial (number or string) to also make the modules externally identifiable.

The routing table (associations “Full SN – CAN ID”) is stored on the SC so that the same CAN ID is assigned to the same module at the next reboot of the system or reconnection of the module. The SC will assign free CAN IDs to the modules not present yet in the table. The table is updated only when the synchronization is complete (all Production Serials are available on the SC).

The routing table can be manipulated to assign specific CAN IDs to modules manually (e.g. based on place of the module in the actual string installation).

Graphical user interface

Description automatically generated

Can Bootloader

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Values** | **Report intvl** |
| 117 | Acknowledge | Report the status of the given command by MPU during firmware upgrade.    Payload – 1B  **OK**                              = 0x70  **ERROR**                       = 0x71  **SEQ\_ERROR**             = 0x74 |  |
| 118 | Progress Notification | Bootloader notifies MPU about the progress in certain operations    Payload – 1B  **START** = 0x20  Sent every 5 second until the unlock command  **IMAGE\_VERIFY** = 0x21  **COPY\_FW** = 0x22  **RESETTING**                = 0x24  **ABORTING** = 0x25 |  |
| 122 | Unlock | Unlocks the memory region to be flashed, indicating start of the flashing operation as well.    Payload – 1B  **Bootloader** = 0x00  **Application** = 0x01 |  |
| 123 | Data | This command indicates firmware bytes are being sent in the payload    Payload – 8B Firmware    Data command send 8 consecutive and separate messages indexes increase from 0 to 7. In total 64B should be sent in 8 messages before the next step. If indexing is wrong, SEQ\_ERROR will be sent in ACK field and the last sent bunch of bytes are discarded. |  |
| 124 | Verify | From 64B of firmware (8 Data messages), CRC32 should be calculate with polynomial 0xEDB88320 and seed 0xFFFFFFFF. Module controller will compare this value with its own calculated CRC and acknowledge the result with ACK field. |  |
| 125 | Verify Image | After sending the full image, image CRC32 should be calculate with polynomial 0xEDB88320 and seed 0xFFFFFFFF. Module controller will compare this value with its own calculated CRC and acknowledge the result with ACK field. If it is unsuccessful, bootloader will get a clean soft reset. |  |
| 126 | Reset | This command initiates the final operations before the end of firmware upgrade process. Should be called once at the end for module controller to boot into new firmware. |  |
| 127 | Abort | Aborting command. Module controller will reboot into the unlocked region. If it was an application update requested, it will boot into the old application. Otherwise will boot into old bootloader. |  |

## SC-MCU Register List

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Name | Values | Report intvl |
| 128 | SC-MCU Statuses | UInt8 (READONLY)  **Primary state**  See description below | 2s |
| UInt8 (READONLY)  **Contactor states**  See description below |
| UInt8 (READONLY)  **Safety relays & FPGA status**  See description below |
| UInt8 (READONLY)  **Simulation Mode State**  2^0 = Simulation Active  2^4 = Sim. Expired  2^5 = Sim. Denied |
| UInt32 (READONLY)  **Alarms**  See description below |
| 129 | String & Bus Voltages | Int32 (READONLY)  **Vstring mV** |  |
| Int32 (READONLY)  **Vbus mV** |
| 130 | Current & CM Voltage | Int32 (READONLY)  **I mA** |  |
| Int32 (READONLY)  **Vcm mV** |
| 131 | Fuse Voltage & ADC Errors | Int32 (READONLY)  **Vfuse mV** |  |
| UInt8 (READONLY)  **ADCerr**  Bitwise flags:  2^0 = ADC1  2^1 = ADC2  2^2 = ADC3  2^3 = ADC4  2^4 = ADC5 |
| 132 | FPGA revision ID | UInt8 (READONLY)  **Revision ID** |  |
| 144 | Commands | UInt8 (WRITEONLY)  **CMD**  Bitwise flags:  2^0 = On/Off  2^1 = Brown-out  2^2 = Reset  2^3 = Simulation Mode |  |
| 145 | FPGA Configuration 1 | UInt16 (READWRITE)  **V\_string ADC clkdiv** |  |
| UInt16 (READWRITE)  **V\_bus ADC clkdiv** |  |
| UInt16 (READWRITE)  **I\_string clkdiv** |  |
| UInt16 (READWRITE)  **V\_fuse clkdiv** |  |
| 146 | FPGA Configuration 2 | UInt16 (READWRITE)  **V\_cm ADC clkdiv** |  |
| UInt16 (READWRITE)  **String-bus delta threshold (mV)** |  |
| 147 | FPGA Configuration 3 | UInt16 (READWRITE)  **Report intv Status** |  |
| UInt16 (READWRITE)  **Report intv Error** |  |
| UInt16 (READWRITE)  **Report intv V\_string** |  |
| UInt16 (READWRITE)  **Report intv V\_bus** |  |
| 148 | FPGA Configuration 4 | UInt16 (READWRITE)  **Report intv I\_string** |  |
| UInt16 (READWRITE)  **Report intv V\_fuse** |  |
| UInt16 (READWRITE)  **Report intv V\_cm** |  |
| 149 | FPGA Configuration 5 | UInt16 (READWRITE)  **Filter config** |  |
| 150 | MCU Configuration | Int32 (READWRITE)  **Number of Modules** |  |
| 151 | “ | Int32 (READWRITE)  **String Volt High Alarm [mV /module]** |  |
| 152 | “ | Int32 (READWRITE)  **String Volt High Warn [mV /module]** |  |
| 153 | “ | Int32 (READWRITE)  **String Volt Low Warn** |  |
| 154 | “ | Int32 (READWRITE)  **String Volt Low Min Switch-on [mV /module]** |  |
| 155 | “ | Int32 (READWRITE)  **Bus Volt High Alarm [mV /module]** |  |
| 156 | “ | Int32 (READWRITE)  **Bus Volt High Warn [mV /module]** |  |
| 157 | “ | Int32 (READWRITE)  **String-Bus Delta Voltage [mV/module]** |  |
| 158 | “ | Int32 (READWRITE)  **String Current Charge Alarm [mA]** |  |
| 159 | “ | Int32 (READWRITE)  **String Current Charge Warn [mA]** |  |
| 160 | “ | Int32 (READWRITE)  **String Current Discharge Alarm [mA]** |  |
| 161 | “ | Int32 (READWRITE)  **String Current Discharge Warn [mA]** |  |
| 162 | “ | Int32 (READWRITE)  **Common Mode Threshold [mV]** |  |
| 163 | “ | Int32 (READWRITE)  **Fuse Threshold [mV]** |  |
| 164 | “ | Int32 (READWRITE)  **Contactor Timeout [ms]** |  |
| 165 | “ | Int32 (READWRITE)  **Pre-charge Minimum Ramp-up [mV/ms]** |  |
| 166 | “ | Int32 (READWRITE)  **Pre-charge Timeout** |  |
| 167 | “ | Int32 (READWRITE)  **Loose Power Lead Check Intv [ms]** |  |
| 168 | “ | Int32 (READWRITE)  **Loose Power Lead Check Threshold [mV]** |  |
| 169 | “ | Int32 (READWRITE)  **MPU Comm Timeout [ms]** |  |
| 170 | “ | Int32 (READWRITE)  **Alarm Delay Short** |  |
| 171 | “ | Int32 (READWRITE)  **Alarm Delay Long** |  |
| 172 |  | Int32 (READWRITE)  **ADC conversion offset ch0** |  |
| 173 | “ | Int32 (READWRITE)  **ADC conversion offset ch1** |  |
| 174 | “ | Int32 (READWRITE)  **ADC conversion offset ch2** |  |
| 175 | “ | Int32 (READWRITE)  **ADC conversion offset ch3** |  |
| 176 | “ | Int32 (READWRITE)  **ADC conversion offset ch4** |  |
| 177 |  | Int32 (READWRITE)  **ADC conversion gain ch0** |  |
| 178 | “ | Int32 (READWRITE)  **ADC conversion gain ch1** |  |
| 179 | “ | Int32 (READWRITE)  **ADC conversion gain ch2** |  |
| 180 | “ | Int32 (READWRITE)  **ADC conversion gain ch3** |  |
| 181 | “ | Int32 (READWRITE)  **ADC conversion gain ch4** |  |
| 182 | Configuration rejected | UInt8 (READONLY)  **Config parameter ID\*** |  |
| 183 | Reset Cause | UInt8 (READONLY)  0: Power cycle  1: Soft reset  2: Watchdog  3: Unknown |  |
| 184 | MCU Soft Reset Request | UInt8 (WRITEONLY)  Character ‘R’ should be sent to the MCU to request a soft reset. |  |
| 185 | Simulation Mode | Uint8 (WRITEONLY)  Simulation mode is enabled if 0xA5 is sent as a single byte payload. |  |
| 255 | Internal CAN Echo | UInt16 (WRITEONLY)  **CAN Echo Data** |  |

Table 4 - SC-MCU CAN Register

\*: If configuration is accepted, 0xFF is the payload

**SC-MCU status**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controller State** | **String State** |  |  | **Code** |
| Power-On | Disconnected | 0000 | 0000 | 0x00 |
| Disconnected |  | 0000 | 0001 | 0x01 |
| Check conditions | Connecting | 0001 | 0000 | 0x10 |
| Close pre-charge CT |  | 0001 | 0001 | 0x11 |
| Wait for pre-charge CT FB |  | 0001 | 0010 | 0x12 |
| Close NEG CT |  | 0001 | 0011 | 0x13 |
| Wait for NEG CT FB |  | 0001 | 0100 | 0x14 |
| Check Ramp-up |  | 0001 | 0101 | 0x15 |
| Pre-charging |  | 0001 | 0110 | 0x16 |
| Close POS CT |  | 0001 | 0111 | 0x17 |
| Wait for POS CT FB |  | 0001 | 1000 | 0x18 |
| Open pre-charge CT |  | 0001 | 1001 | 0x19 |
| Connected | Connected | 0010 | 0000 | 0x20 |
| Open All CTs | Disconnecting | 0100 | 0000 | 0x40 |
| Wait for CT FB |  | 0100 | 0001 | 0x41 |
| Error | Error | 1111 | 1111 | 0xFF |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Contactor states | | (overlapping, states care OR-ed between contactors) | | | | | |
| Pre-charge | Open |  |  |  | 0000 | 0000 | 0x00 |
|  | Closed |  |  |  | 0000 | 0001 | 0x01 |
|  | Waiting |  |  |  | 0000 | 0010 | 0x02 |
|  | Error |  |  |  | 0000 | 0011 | 0x03 |
| NEG | Open |  |  |  | 0000 | 0000 | 0x00 |
|  | Closed |  |  |  | 0000 | 0100 | 0x04 |
|  | Waiting |  |  |  | 0000 | 1000 | 0x08 |
|  | Error |  |  |  | 0000 | 1100 | 0x0C |
| POS | Open |  |  |  | 0000 | 0000 | 0x00 |
|  | Closed |  |  |  | 0001 | 0000 | 0x10 |
|  | Waiting |  |  |  | 0010 | 0000 | 0x20 |
|  | Error |  |  |  | 0011 | 0000 | 0x30 |

|  |  |
| --- | --- |
| Alarms & Warnings (Bitmapped) | Bit |
| String Voltage Low Alarm | 0 |
| String Voltage High Alarm | 1 |
| String Current Low Alarm (Discharging) | 2 |
| String Current High Alarm (Charge) | 3 |
| Bus Voltage High Alarm | 4 |
| Fuse Voltage Alarm | 5 |
| String Voltage Low Warning | 6 |
| String Voltage High Warning | 7 |
| String Current Low Warning (Discharging) | 8 |
| String Current High Warning (Charging) | 9 |
| Bus Voltage High Warning | 10 |
| Common Mode Warning | 11 |
| FPGA Errors | 12 |
| FPGA Communication Error | 13 |
| Module Alarm | 14 |
| Loose Power Lead | 15 |
| Loose Power Lead Check Timeout | 16 |
| String Watchdog Failure | 17 |
| String Controller Invalid State | 18 |
| String Controller Ramp-up Timeout | 19 |
| String Controller Pre-charge Timeout | 20 |
| String Controller Pre-charge Contactor Not Open | 21 |
| String Controller Contactor Feedback Timeout | 22 |
| String Controller Contactor Invalid State | 23 |
| String Controller Contactor Unexpected State Change | 24 |
| String Controller Intercore Keepalive Error | 25 |
| String Controller Module Number Mismatch | 26 |
| String Controller Too Many CAN Misses  String Controller Connection Conditions NOT Met **1** | 27  28 |

**1** : Bit 28 is used as a flag to indicate if the connection conditions are met. The current condition is not having any alarms and no EMS wire break.

|  |  |  |
| --- | --- | --- |
| Safety Relay & FPGA Status |  | |
| HVIL feedback | 0000 | 0001 | | 2^0 |
| Fail-safe feedback | 0000 | 0010 | | 2^1 |
| Emergency Stop feedback | 0000 | 0100 | | 2^2 |
| Emergency Stop Wire-Break | 0000 | 1000 | | 2^3 |
| FPGA NOT Running | 0001 | 0000 | | 2^4 |
| FPGA Delta Threshold OK | 0010 | 0000 | | 2^5 |
| FPGA ADC Failsafe Status Bit0 | 0100 | 0000 | | 2^6 |
| FPGA ADC Failsafe Status Bit1 | 1000 | 0000 | | 2^7 |

|  |  |  |
| --- | --- | --- |
| **FPGA ADC Failsafe Status** | **Bit1** | **Bit0** |
| Normal | 0 | 0 |
| ADC Not Okay | 0 | 1 |
| ADC Sampling Error | 1 | 0 |
| ADC Data Not Valid | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **Emergency Stop Circuit STATE** | **EM Stop Feedback** | **Wire-break Feedback** |
| Normal | 0 | 0 |
| Emergency Stop | 1 | 0 |
| Wire-break | 0 | 1 |
| Short Circuit | 1 | 1 |

1. Simulation mode denied/timeout flag will be cleared when the simulation mode is de-activated again using the “Simulation mode Configuration” register. [↑](#footnote-ref-2)