MILLSWOOD engineering

250W PMU V2

CAN PROTOCOL

1 Introduction

The 250W PMU has both RS232 and CANbus interfaces that perform essentially the same functions, these being:

- Configuration (of parameters stored in the PMU's non-volatile memory),
- Control (real-time control of the PMU's various features), and
- Monitoring (of measured voltages, currents, temperatures, etc).

Once the PMU has been configured, there is no requirement to connect anything to either communications interface – the PMU will operate quite normally with no communications at all.

This document describes the default CAN communications interface and protocol (RS232 is described in the 250W RS232 protocol document). Custom CAN protocols can also be developed to suit existing CAN ID structures; please contact us to discuss your requirements.

2 Overview of CAN

CAN is a multi-master broadcast serial bus, originally developed for automotive applications but now used extensively across a wide range of industries. CAN provides more robust communications than is possible with RS232, and includes automatic arbitration-free transmission, message prioritisation, automatic retries, CRC data protection, fault confinement and more.

Physically CAN is usually implemented as a 2-wire differential serial bus, although a third ground wire is always recommended. The bus must be terminated at each end. This can be a simple 120 Ohm resistor connected across the two signal lines, or it can be a pair of 60 Ohm resistors connecting each signal line to a rail biased midway between the minimum and maximum signal voltages. The second arrangement is superior as it provides far greater immunity from electrical noise. The 250W PMU does not terminate the bus.

The 250W PMU supports Baudrates of 125, 250, 500 and 1000 kb/s.

The CAN specification defines four frame types (data, remote, error and overload), but only the data frame can actually transmit any payload data. Like many CAN implementations, only the data frame is used here. Data frames can have 0 to 8 bytes of payload data.

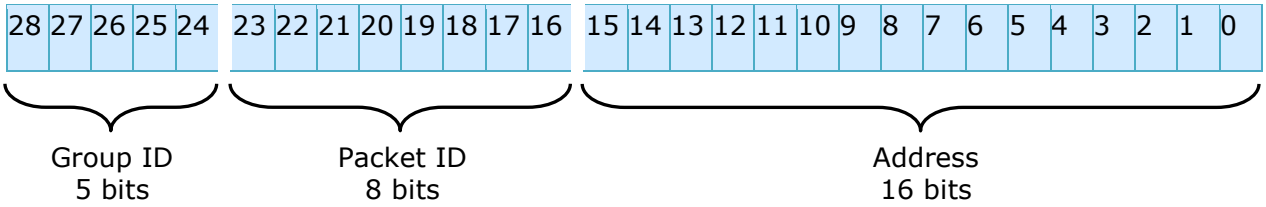
This protocol is based on CAN 2.0B; i.e. CAN frames have a 29-bit message identifier associated with them. The message ID is divided into 3 parts as described in the next section.

This page intentionally left blank.

3 Contents

4 CAN Message ID

Each CAN message is preceded by the 29-bit CAN message ID, which specifies the type of hardware (group ID), the type of content (packet ID), and the address of the device associated with the message:



4.1 Group ID

Specifies the type of hardware that this CAN frame came from or is being sent to. The PMU always has a group ID of 30 (0x1E).

4.2 Packet ID

Specifies the contents of a packet. The following packet types are defined for the 250W PMU:

Packet ID	Name	Direction (with respect to PMU)	Length (number of data bytes)	Description
0x00	Voltages	Out	7	Contains measured voltages
0x01	Currents	Out	6	Contains measured currents
0x02	Battery statuses	Out	4	Contains calculated battery energies
0x03	Temperatures	Out	5	Contains measured temperatures
0x04	Miscellaneous	Out	5	Contains miscellaneous quantities
0x0F	Measurement request	In	1 -> 0 to 5 packets	Request a set of measurement values
0x10	Set V_A	In	1	Set Avionics voltage
0x11	Set V_P	In	1	Set Payload voltage
0x12	Set V_S	In	1	Set Servo voltage
0x13	Set V_B	In	1	Set Battery voltage
0x14	Set PP	In	1	Set packet period
0x15	Set PS	In	1	Set packets streamed
0x18	Set S_0	In	1	Set power-up state
0x19	Set CT	In	1	Set cranking time
0x1B	Set CA	In	2	Set CAN address
0x20	Set output states	In	1	Control or determine the enable status of the various outputs
0x21	Enable outputs	In	1	Turn specified outputs on
0x22	Disable outputs	In	1	Turn specified outputs off
0x23	Set generation state	In	1	Turn electrical power generation on and off
0x24	Start	In	0	Start cranking the engine
0x25	Stop	In	0	Stop cranking the engine
0x26	Reset	In	0	Restart the PMU
0x30	Serial number	In	0 -> 2	Request PMU serial number
0x31	Firmware	In	0 -> 6	Request firmware information

Table 1 – Blue: Measured/calculated values; Green: User-defined values stored in non-volatile memory; Red: Volatile values & commands; Black: Fixed values stored in non-volatile memory.

For packets with IDs up to 0x20 inclusive, if sent to the PMU with zero data bytes the PMU will return a packet containing the current values.

4.3 Address

Each PMU within any given network must have a unique address between 0 and 65534 inclusive. 65535 (0xFFFF) is reserved to form a broadcast message ID to which all PMUs will respond. PMUs are shipped with a default address of 1.

5 CAN Packet Types

Please note that all 2-byte quantities are transmitted and received in big-endian format; i.e. high byte first, followed by the low byte.

5.1 Packet ID 0x00 – Voltages

This packet contains measured voltages. The PMU can be configured to stream this packet at regular intervals, or it can be requested by issuing this packet with zero data bytes.

Byte	Name	Description
0	Avionics output voltage	An unsigned byte with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 250 (0.0 to 25.0V).
1	28VDC output voltage	An unsigned byte with value 5 times the measured voltage (i.e. in 0.2V increments). Value ranges from 0 to 250 (0.0 to 50.0V).
2	Payload output voltage	An unsigned byte with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 250 (0.0 to 25.0V).
3	Servo output voltage	An unsigned byte with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 125 (0.0 to 12.5V).
4	Battery A voltage	An unsigned byte with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 250 (0.0 to 25.0V).
5	Battery B voltage	An unsigned byte with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 250 (0.0 to 25.0V).
6	Generator voltage	An unsigned byte with value equal to the measured voltage. Value ranges from 0 to 250 (0 to 250V).

Table 2

5.2 Packet ID 0x01 – Currents

This packet contains measured currents. The PMU can be configured to stream this packet at regular intervals, or it can be requested by issuing this packet with zero data bytes.

Byte	Name	Description
0	Avionics output current	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 100 (0.0 to 10.0A).
1	28VDC output current	A signed byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from -125 to +125 (-12.5 to +12.5A).
2	Payload output current	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 100 (0.0 to 10.0A).
3	Servo output current	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 100 (0.0 to 10.0A).
4	Battery A current	A signed byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from –125 to +125 (-12.5 to +12.5A).
5	Battery B current	A signed byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from –125 to +125 (-12.5 to +12.5A).

Table 3

5.3 Packet ID 0x02 – Battery statuses

This packet contains calculated battery charge statuses. The PMU can be configured to stream this packet at regular intervals, or it can be requested by issuing this packet with zero data bytes.

Byte	Name	Description
0, 1	Battery A energy	A signed integer with value equal to the time-integral of current into battery A since power up in mAH. Value ranges from −32768 to +32767.
2, 3	Battery B energy	A signed integer with value equal to the time-integral of current into battery B since power-up in mAH. Value ranges from −32768 to +32767.

Table 4

5.4 Packet ID 0x03 – Temperatures

This packet contains measured temperatures. The PMU can be configured to stream this packet at regular intervals, or it can be requested by issuing this packet with zero data bytes.

Byte	Name	Description
0	Internal PMU temperature	A signed byte with value equal to the temperature inside the PMU in degrees Celsius. Value ranges from −128 to +127.
1	Battery A temperature	A signed byte with value equal to the temperature of battery A in degrees Celsius. Value ranges from −128 to +127 (−128 = no sensor).
2	Battery B temperature	A signed byte with value equal to the temperature of battery B in degrees Celsius. Value ranges from -128 to +127 (−128 = no sensor).
3	Generator temperature	A signed byte with value equal to the temperature of the generator (BLDC motor) in degrees Celsius. Value ranges from -128 to +127 (−128 = no sensor).
4	Engine starter module temperature	This temperature is no longer sensed. The value returned is always -128.

Table 5

5.5 Packet ID 0x04 – Miscellaneous

This packet contains miscellaneous measured and derived quantities. The PMU can be configured to stream this packet at regular intervals, or it can be requested by issuing this packet with zero data bytes.

Byte	Name	Description
0, 1	Generator speed	An unsigned integer with value equal to the generator speed in RPM. Value ranges from 0 to 65535.
2	Flag register 0	The following bits are defined: Bit 2: Battery A charge termination flag (1 = terminated) Bit 3: Battery B charge termination flag (1 = terminated) Bit 4: Electrical power generation flag (1 = enabled) Bit 5: Thermal shutdown flag (1 = shutdown) Bit 6: Overvoltage shutdown flag (1 = shutdown) Bit 7: Payload shedding flag (1 = shed)
3	Flag register 1	The following bits are defined: Bit 7: Engine starter module status flag (1 = ready)
4	Flag register 2	The following bits are defined: Bit 0: Avionics and servo outputs (1 = enabled) Bit 1: Payload output (1 = enabled) Bit 2: Battery A charger (1 = enabled) Bit 3: Battery B charger (1 = enabled)

Table 6

5.6 Packet ID 0x0F – Measurement request

This packet requests a set of measurements from the PMU. The measurements that are required may be specified in the data byte. If this packet is issued with zero data bytes all measurements will be returned.

Use of this packet allows the supervising entity to poll the PMU, as an alternative to having the PMU push values onto the CAN bus at regular intervals using the PMU's packet streaming features.

Byte	Name	Description
0	MR	Measurements requested. This byte indicates the measurements that are to be returned. Bits have the following significance: Bit 0: Packet ID 0x00 (Voltages) Bit 1: Packet ID 0x01 (Currents) Bit 2: Packet ID 0x02 (Battery Statuses) Bit 3: Packet ID 0x03 (Temperatures) Bit 4: Packet ID 0x04 (Miscellaneous) Bits 5–7: X (don't care) 0 = disabled, 1 = enabled.

Table 7

5.7 Packet ID 0x10 – Set (or Get) V_A

This packet sets the stored value of V_A, the avionics output voltage. Note that V_A is the configuration value stored in non-volatile memory, not the measured value. The value of V_A may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	V_A	Avionics output voltage. An unsigned byte with value 10 times the configured voltage (i.e. in 0.1V increments). V_A may be set to any value from 12.0 to 24.0V inclusive, corresponding to unsigned byte values of 120 to 240.

Table 8

5.8 Packet ID 0x11 – Set (or Get) V_P

This packet sets the stored value of V_P, the payload output voltage. Note that V_P is the configuration value stored in non-volatile memory, not the measured value. The value of V_P may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	V_P	Payload output voltage. An unsigned byte with value 10 times the configured voltage (i.e. in 0.1V increments). V_P may be set to any value from 12.0 to 24.0V inclusive, corresponding to unsigned byte values of 120 to 240.

Table 9

5.9 Packet ID 0x12 – Set (or Get) V_S

This packet sets the stored value of V_S, the servo output voltage. Note that V_S is the configuration value stored in non-volatile memory, not the measured value. The value of V_S may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	V_S	Servo output voltage. An unsigned byte with value 10 times the configured voltage (i.e. in 0.1V increments). V_S may be set to any value from 5.0 to 12.0V inclusive, corresponding to unsigned byte values of 50 to 120.

Table 10

5.10 Packet ID 0x13 – Set (or Get) V_B

This packet sets the stored value of V_B, the battery charging voltage. Note that V_B is the configuration value stored in non-volatile memory, not the measured value. The value of V_B may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	V_B	Battery charging voltage. An unsigned byte with value 10 times the configured voltage (i.e. in 0.1V increments). V_B may be set to any value from 20.0 to 25.2V inclusive, corresponding to unsigned byte values of 200 to 252.

Table 11

5.11 Packet ID 0x14 – Set (or Get) PP

This packet sets the stored value of PP, the packet period. This is the interval of time between successive transmissions of streamed data. The packets that are streamed are defined by PS, the packets streamed value. The value of PP may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	PP	Packet period. An unsigned byte with value 10 times the packet period (i.e. in 0.1S increments). The packet period may be set to any value from 0.1 to 25.5 seconds, corresponding to unsigned byte values of 1 to 255.

Table 12

5.12 Packet ID 0x15 – Set (or Get) PS

This packet sets the stored value of PS, the packets that are streamed. The value of PS may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	PS	Packets streamed. This byte indicates the packets that are streamed. Bits have the following significance: Bit 0: Packet ID 0x00 (Voltages) Bit 1: Packet ID 0x01 (Currents) Bit 2: Packet ID 0x02 (Battery Statuses) Bit 3: Packet ID 0x03 (Temperatures) Bit 4: Packet ID 0x04 (Miscellaneous) Bits 5–7: X (don't care) 0 = disabled, 1 = enabled.

Table 13

5.13 Packet ID 0x18 – Set (or Get) S_0

This packet sets the stored value of S_0, the PMU's initial state. S_0 defines how the PMU's outputs and features are configured immediately after power-up or reset. The value of S_0 may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	S_0	Initial state. Bits have the following significance: Bit 0: Avionics and servo outputs Bit 1: Payload output Bit 2: Battery A charger Bit 3: Battery B charger Bit 5: Payload shedding Bits 4, 6 and 7: X (don't care) 0 = disabled, 1 = enabled.

Table 14

5.14 Packet ID 0x19 – Set (or Get) CT

This packet sets the stored values of CT, the cranking time. This parameter is only relevant for units fitted with an engine starter. The value of CT may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	CT	Cranking time. An unsigned byte with value 10 times the maximum cranking time (i.e. in 0.1S increments). The cranking time may be set to any value from 0.1 to 25.5 seconds, corresponding to unsigned byte values of 1 to 255. A value of 0 disables the timeout.

Table 15

5.15 Packet ID 0x1B – Set (or Get) CA

This packet sets the stored values of CA, the CAN address. The value of CA may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0, 1	CA	CAN address. The CAN address may be set to any value from 0 to 65534 (0x0000 to 0xFFFE) inclusive. 65535 (0xFFFF) is a "broadcast" address to which all PMUs will respond (provided that the rest of the ID is valid). This is useful for determining unknown or forgotten addresses.

Table 16

5.16 Packet ID 0x20 – Set (or Get) output states

This packet enables or disables the various outputs. Conversely, the enable status of the various outputs may be obtained by issuing this packet with zero data bytes.

When this packet is used to control the various outputs, it may be prudent to perform a Get usage followed by a Set usage, in order to not change the state of other outputs unintentionally. Output states are also streamed in the Miscellaneous packet.

Enabling and disabling the various outputs may be more easily accomplished using the Enable and Disable outputs packets, as these packets can modify the state of an arbitrary combination of outputs without affecting the remaining outputs.

Byte	Name	Description
0	State	Bits have the following significance: Bit 0: Avionics and servo outputs Bit 1: Payload output Bit 2: Battery A charger Bit 3: Battery B charger Bits 4–7: X (don't care) 0 = disabled, 1 = enabled.

Table 17

5.17 Packet ID 0x21 – Enable outputs

This packet turns one or more outputs on. Other outputs are unaffected.

Byte	Name	Description
0	Enable	This byte determines which outputs are to be enabled. Bits have the following significance: Bit 0: Avionics and servo outputs Bit 1: Payload output Bit 2: Battery A charger Bit 3: Battery B charger Bits 4–7: X (don't care) 0 = no action, 1 = enable.

Table 18

5.18 Packet ID 0x22 – Disable outputs

This packet turns one or more outputs off. Other outputs are unaffected.

Byte	Name	Description
0	Disable	This byte determines which outputs are to be disabled. Bits have the following significance: Bit 0: Avionics and servo outputs Bit 1: Payload output Bit 2: Battery A charger Bit 3: Battery B charger Bits 4–7: X (don't care) 0 = no action, 1 = disable.

Table 19

5.19 Packet ID 0x23 – Set (or Get) generation state

This packet turns electrical power generation on and off. The current state of electrical power generation may be obtained by issuing this packet with zero data bytes. It is also streamed in the Miscellaneous packet.

Turning electrical power generation off removes the electromechanical load from the internal combustion engine; it **DOES NOT** affect the avionics, servo or payload outputs (as long as at least one battery is connected).

Byte	Name	Description
0	Gen	Electrical power generation. Bit 0 of this byte determines the current state of electrical power generation. 0 = disabled, 1 = enabled.

Table 20

5.20 Packet ID 0x24 – Start

Starts cranking the engine. This parameter is only relevant for units fitted with an engine starter. Must be issued with zero data bytes.

5.21 Packet ID 0x25 – Stop

Stops cranking the engine. This parameter is only relevant for units fitted with an engine starter. This command is included for safety reasons only (the engine starter automatically disengages when it detects that the engine has started). Must be issued with zero data bytes.

5.22 Packet ID 0x26 – Reset

Restarts the PMU. Outputs are set to their power-up states, the battery energies are reset to zero, and electrical power generation is enabled. Must be issued with zero data bytes.

Provided that the avionics and servo outputs are on when the Reset packet is received, and the power-up state S_0 specifies that they are also enabled at power-up, these outputs will remain stable throughout the restart process.

5.23 Packet ID 0x30 – Serial number

On reception of this packet ID (with zero data bytes), the PMU will respond with a packet containing the following data:

Byte	Name	Description
0, 1	Serial number	Unsigned word, range 2000 – 65535. Every PMU is given a unique serial number at manufacture. This is a read-only value and cannot be changed. It is not affected by firmware updates.

Table 21

5.24 Packet ID 0x31 – Firmware

On reception of this packet ID (with zero data bytes), the PMU will respond with a packet containing the following data:

Byte	Name	Description
0	Version major	Unsigned byte, range 2 – 99.
1	Version minor	Unsigned byte, range 0 – 99.
2	Build day	Unsigned byte, range 1 – 31.
3	Build month	Unsigned byte, range 1 – 12.
4, 5	Build year	Unsigned word, ranges from 2024 and up.

Table 22

6 Document version history

6.1 1.9 -> 2.0

- Packet ID 0x00 – Generator voltage scaling changed to accommodate voltages over 99V.
- Packet ID 0x03 – The engine starter module temperature is no longer sensed and will always be reported as -128C.
- Packet ID 0x04 – Flag register 0 bits 2 and 3 now indicate battery charge termination, and bit 6 indicates overvoltage shutdown.
- Packet ID 0x16 deleted. Calibration is no long required
- Packet ID 0x17 deleted. The upper temperature limit is now fixed internally.
- Packet ID 0x18 – Battery disconnect detect (bit 4) and Safe operating area management (bit 6) removed. These bits are now ignored by the firmware.
- Packet IDs 0x30, 0x31 – Serial number and firmware version ranges updated.