1700W GCU

CAN PROTOCOL

1 Introduction

The 1700W GCU has both RS232 and CANbus interfaces that perform essentially the same functions, these being:

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

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

This document describes the default CAN communications interface and protocol (RS232 is described in the 1700W 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 1700W GCU does not terminate the bus.

The 1700W GCU 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.

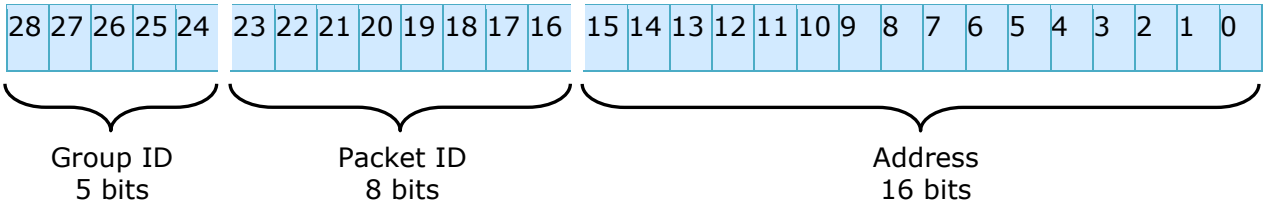
This protocol is based on CAN 2.0B; i.e. CAN frames have a 29-bit message identifier associated with them. The message ID is divided into 3 parts as described in the next section.

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 GCU 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 1700W GCU:

Packet ID	Name	Direction (with respect to GCU)	Length (number of data bytes)	Description
0x00	Voltages	Out	12	Contains measured voltages
0x01	Currents	Out	7	Contains measured currents
0x02	Battery statuses	Out	4	Contains calculated battery energy
0x03	Temperatures	Out	5	Contains measured temperatures
0x04	Miscellaneous	Out	7	Contains miscellaneous quantities
0x0F	Measurement request	In	1 -> 0 to 6 packets	Request a set of measurement values
0x12	Set I_B	In	1	Set battery charging current
0x13	Set V_B	In	2	Set battery voltage
0x14	Set PP	In	1	Set packet period
0x15	Set PS	In	1	Set packets streamed
0x17	Set T_U	In	1	Set upper temperature limit
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
0x1C	Set CS	In	1	Set cranking speed
0x1D	Set I_C	In	1	Set cranking current
0x1E	Set CF	In	1	Set cranking flags
0x1F	Set PC	In	1	Set pole count
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
0x24	Start	In	0	Start cranking the engine
0x25	Stop	In	0	Stop cranking the engine
0x26	Reset	In	0	Restart the GCU
0x30	Serial number	In	0 -> 2 bytes	Request GCU serial number
0x31	Firmware	In	0 -> 6 bytes	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 GCU with zero data bytes the GCU will return a packet (or pair of packets) containing the current values.

4.3 Address

Each GCU 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 GCUs will respond. GCUs 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 pair of packets contain measured voltages. The GCU can be configured to stream these packets at regular intervals, or they can be requested by issuing this packet ID with zero data bytes.

Byte	Name	Description
0, 1	Battery A voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 620 (0.0 to 62.0V).
2, 3	Battery B voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 620 (0.0 to 62.0V).
4, 5	Main power output voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 620 (0.0 to 62.0V).
6, 7	Generator (BLDC motor) voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 1300 (0.0 to 130.0V).

Table 2 – Voltages (packet 1 of 2)

Byte	Name	Description
0, 1	Umbilical input voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 1300 (0.0 to 130.0V).
2, 3	Internal HV bus voltage	An unsigned integer with value 10 times the measured voltage (i.e. in 0.1V increments). Value ranges from 0 to 1300 (0.0 to 130.0V).

Table 3 – Voltages (packet 2 of 2)

5.2 Packet ID 0x01 – Currents

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

Byte	Name	Description
0	Battery A current	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 120 (0.0 to 12.0A).
1	Battery B current	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 120 (0.0 to 12.0A).
2	Main power output current, A-side	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 120 (0.0 to 12.0A).
3	Main power output current, B-side	An unsigned byte with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 120 (0.0 to 12.0A).
4, 5	Generator (BLDC motor) current	An unsigned integer with value 10 times the measured current (i.e. in 0.1A increments). Value ranges from 0 to 300 (0.0 to 30.0A).
6	Cranking current	An unsigned integer with value equal to the measured current. Value ranges from 0 to 60.

Table 4 – Currents

5.3 Packet ID 0x02 – Battery statuses

This packet contains calculated battery charge statuses. The GCU 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 5

5.4 Packet ID 0x03 – Temperatures

This packet contains measured temperatures. The GCU 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 temperature	A signed byte with value equal to the temperature inside the GCU 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	A signed byte with value equal to the temperature of the engine starter module in degrees Celsius. Value ranges from –128 to +127 (–128 = no module).

Table 6

5.5 Packet ID 0x04 – Miscellaneous

This packet contains miscellaneous measured and derived quantities. The GCU 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 5: Thermal shutdown flag (1 = shutdown) Bit 6: Start button pressed flag (1 = pressed) Bit 7: Arm input active flag (1 = active)
3	Flag register 1	The following bits are defined: None
4	Flag register 2	The following bits are defined: Bit 0: Main power output (1 = enabled) Bit 2: Battery charger A (1 = enabled) Bit 3: Battery charger B (1 = enabled)
5	Flag register 3	The following bits are defined: Bit 7: Engine starter module status flag (1 = ready)
6	PWM duty cycle	An unsigned byte with value equal to the engine starter PWM duty cycle during cranking. Value ranges from 0 to 100%.

Table 7

5.6 Packet ID 0x0F – Measurement request

This packet requests a set of measurements from the GCU. 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 GCU, as an alternative to having the GCU push values onto the CAN bus at regular intervals using the GCU'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 8

5.7 Packet ID 0x12 – Set (or Get) I_B

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

Byte	Name	Description
0	I_B	Battery charging current. An unsigned byte with value 10 times the configured current (i.e. in 0.1A increments). I_B may be set to any value from 0.5 to 8.0A inclusive, corresponding to unsigned integer values of 5 to 80.

Table 9

5.8 Packet ID 0x13 – Set (or Get) V_B

This packet sets the stored values 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, 1	V_B	Battery charging voltage. An unsigned integer with value 10 times the configured voltage (i.e. in 0.1V increments). V_B may be set to any value from 40.0 to 55.0V inclusive, corresponding to unsigned integer values of 400 to 550.

Table 10

5.9 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 11

5.10 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 12

5.11 Packet ID 0x17 – Set (or Get) T_U

This packet sets the stored value of T_U, the upper temperature limit. The value of T_U may be obtained by issuing this packet with zero data bytes.

There is approximately 10% hysteresis. The main power output and both battery chargers are turned off during thermal shutdown.

Byte	Name	Description
0	T_U	Upper temperature limit. An unsigned byte with value equal to the upper temperature limit in degrees Celsius. T_U may be set to any value from 0 to 255 inclusive. Setting T_U to 0 will force the GCU into thermal shutdown, and setting T_U to 255 will disable thermal shutdown.

Table 13

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

This packet sets the stored value of S_0, the GCU's initial state. S_0 defines how the GCU'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: Main power output Bit 2: Battery charger A Bit 3: Battery charger B Bit 6: Keep fans on Bits 1, 4, 5 and 7: X (don't care) 0 = disabled, 1 = enabled.

Table 14

5.13 Packet ID 0x19 – Set (or Get) CT

This packet sets the stored value 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 0.5 to 25.0 seconds, corresponding to unsigned byte values of 5 to 250.

Table 15

5.14 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 GCUs will respond (provided that the rest of the ID is valid). This is useful for determining unknown or forgotten addresses.

Table 16

5.15 Packet ID 0x1C – Set (or Get) CS

This packet sets the stored value of CS, the speed at which the engine is deemed to have started. The engine starter will disengage once the BLDC motor reaches this speed. The value of CS may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	CS	Cranking speed. An unsigned byte with value 100 times the speed in RPM. The cranking speed may be set to any value from 001 to 255, corresponding to speeds of 100 to 25500 RPM.

Table 17

5.16 Packet ID 0x1D – Set (or Get) I_C

This packet sets the stored value of I_C, the cranking current. The value of I_C may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	I_C	Cranking current. An unsigned byte with value equal to the current in Amps. I_B may be set to any value from 1 to 50A inclusive.

Table 18

5.17 Packet ID 0x1E – Set (or Get) CF

This packet sets the stored value of CF, the cranking flags. The value of CF may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	CF	Cranking flags. Bits have the following significance: Bit 0: Direction of rotation Bits 1–7: X (don't care)

Table 19

5.18 Packet ID 0x1F – Set (or Get) PC

This packet sets the stored value of PC, the BLDC motor's pole count. This value is used to calculate the mechanical RPM of the BLDC motor from its electrical RPM, and must be set correctly if engine starting is used. The value of PC may be obtained by issuing this packet with zero data bytes.

Byte	Name	Description
0	PC	Pole count. An unsigned byte with value equal to the number of poles in the BLDC motor/alternator. The number of poles must be an even number in the range 2 to 32 inclusive.

Table 20

5.19 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: Main power output Bit 2: Battery charger A Bit 3: Battery charger B Bits 1, 4–7: X (don't care) 0 = disabled, 1 = enabled.

Table 21

5.20 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: Main power output Bit 2: Battery charger A Bit 3: Battery charger B Bits 1, 4–7: X (don't care) 0 = no action, 1 = enable.

Table 22

5.21 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: Main power output Bit 2: Battery charger A Bit 3: Battery charger B Bits 1, 4–7: X (don't care) 0 = no action, 1 = disable.

Table 23

5.22 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.

5.23 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.

5.24 Packet ID 0x26 – Reset

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

5.25 Packet ID 0x30 – Serial number

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

Byte	Name	Description
0, 1	Serial number	Unsigned word, range 0 – 65535. Every GCU 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 24

5.26 Packet ID 0x31 – Firmware

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

Byte	Name	Description
0	Version major	Unsigned byte, range 0 – 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 2021 and up.

Table 25

6 Document version history

6.1 0.9 -> 1.0

- Initial release.

6.2 1.0 -> 1.1

- Updated product name from 1600W GCU to 1700W GCU.
- Packet ID 0x18 – Set (or Get) S_0 updated to allow configuration of battery chargers and main power output at power-up.
- Packet ID 0x01 – Currents updated to allow for currents greater than 9.9 Amps.

6.3 1.1 -> 1.2

- Packet ID 0x04 – Engine starter PWM duty cycle added.