Gryphon Hardware Information: Dual Single-Wire CAN card

External HD-15 connector pinout

Pin #	Channel	Assignment
01	2	GND
02	1	GND
06	2	CAN BUS
07	1	CAN BUS
10	Both	V- in
15	Both	V+ in

BUS termination requirements

(None.) NOTE: The Gryphon should be powered from the same voltage level as other nodes on a connected SWCAN network. Mismatch of supply voltages can cause misdetection of high-voltage wakeup messages; the high voltage used in such messages is referenced to each module's supply voltage.

Event definitions

The following values may be transmitted as FT_EVENT type GCProtocol frames.
The event value will be in the *event* field of such a frame.
Note: Please see the include files dev_527.h or dev_sja.h for numeric values of the following symbols.

Event name	Event meaning
EVENT_MSG_SENT	An FT_DATA frame with a timestamp value of EVENT_ON_SEND or EVENT_ON_SEND_PROXY has been transmitted. See CMD_SERVER_SET_TIMED_XMIT for more information
GCANBUSOFF	Bus OFF state reached
GCANBUSWARN	Bus WARNING state reached
GCANBUSOK	Bus OK (return from Bus WARNING state)
GCANERRFRAME	Error Frame / Stuff Error detected (1 data byte for position; see table 1)
GCANARBLOST	Arbitration lost (1 data byte for position; see table 1)
GCANRXOVER	Receiver overflow detected
GCANFORM	Form error detected (1 data byte for position; see table 1)
GCANACK	Acknowledgment error (none received during attempted transmit or receive)
GSJABIT	Bit error (1 data byte for position; see table 1)
GSJAOTHER	Other error (1 data byte for position; see table 1)

The following table decodes the single data byte that accompanies the GCANERRFRAME, GCANFORM, GSJABIT and GSJAOTHER event. Two events, one GSJAOTHER and the other GCANFORM are interpreted by the driver and

will never be reported as expected. Both GSJAOTHER events with a data byte of 25 (19 hexadecimal) and GCANFORM events with a data byte of 59 (3B hexadecimal) are reported as GCANACK events. (With no data byte.)

The data byte is shown in binary format. Bit 5 of the values, indicated by x, specifies whether the error occurred while transmitting or receiving the message. A value of 1 indicates the error occurred during reception. A value of 0 indicates that the error occurred during transmission.

TABLE 1

Data byte	Error location
00x0 0011	start of frame
00x0 0010	ID.28 to ID.21
00x0 0110	ID.20 to ID.18
00x0 0100	bit SRTR
00x0 0101	bit IDE
00x0 0111	ID.17 to ID.13
00x0 1111	ID.12 to ID.5
00x0 1110	ID.4 to ID.0
00x0 1100	bit RTR
00x0 1101	reserved bit 1
00x0 1001	reserved bit 0
00x0 1011	data length code
00x0 1010	data field
00x0 1000	CRC sequence
00x1 1000	CRC delimiter
00x1 1001	acknowledge slot
00x1 1011	acknowledge delimiter
00x1 1010	end of frame
00x1 0010	intermission
00x1 0001	active error flag
00x1 0110	passive error flag
00x1 0011	tolerate dominant bits
00x1 0111	error delimiter
00x1 1100	overload flag

The following table decodes the single data byte that accompanies the GCANARBLOST event.

TABLE 2

Data byte	Description
0	arbitration lost in bit 1 of identifier
1	arbitration lost in bit 2 of identifier
2	arbitration lost in bit 3 of identifier
3	arbitration lost in bit 4 of identifier
4	arbitration lost in bit 5 of identifier
5	arbitration lost in bit 6 of identifier
6	arbitration lost in bit 7 of identifier
7	arbitration lost in bit 8 of identifier

8	arbitration lost in bit 9 of identifier
9	arbitration lost in bit 10 of identifier
10	arbitration lost in bit 11 of identifier
11	arbitration lost in bit SRTR (Bit RTR for standard frame messages)
12	arbitration lost in bit IDE
13	arbitration lost in bit 12 of identifier (Extended frame messages only)
14	arbitration lost in bit 13 of identifier (Extended frame messages only)
15	arbitration lost in bit 14 of identifier (Extended frame messages only)
16	arbitration lost in bit 15 of identifier (Extended frame messages only)
17	arbitration lost in bit 16 of identifier (Extended frame messages only)
18	arbitration lost in bit 17 of identifier (Extended frame messages only)
19	arbitration lost in bit 18 of identifier (Extended frame messages only)
20	arbitration lost in bit 19 of identifier (Extended frame messages only)
21	arbitration lost in bit 20 of identifier (Extended frame messages only)
22	arbitration lost in bit 21 of identifier (Extended frame messages only)
23	arbitration lost in bit 22 of identifier (Extended frame messages only)
24	arbitration lost in bit 23 of identifier (Extended frame messages only)
25	arbitration lost in bit 24 of identifier (Extended frame messages only)
26	arbitration lost in bit 25 of identifier (Extended frame messages only)
27	arbitration lost in bit 26 of identifier (Extended frame messages only)
28	arbitration lost in bit 27 of identifier (Extended frame messages only)
29	arbitration lost in bit 28 of identifier (Extended frame messages only)
30	arbitration lost in bit 29 of identifier (Extended frame messages only)
31	arbitration lost in bit RTR of identifier (Extended frame messages only)

Please also see generic events for non card-specific events.

Card-specific IOCTL definitions

These values are used with the CMD_CARD_IOCTL command.
Please see the include files dev_527.h and dev_sja.h for numeric values of IOCTLS and data field bit assignments.

IOCTL name	Function	Data field length/purpose
GCANGETBTRS	Get SJA1000 BTR values	2 data bytes returned: current BTR0 and BTR1 values
GCANSETBTRS	Set SJA1000 BTR values	2 data bytes: new BTR0 and BTR1 values Set values valid only after next INIT ioctl.
GCANGETBC	Get SJA1000 Output Control Register	1 data byte: current output control register
GCANSETBC	Set SJA1000 Output Control Register	1 data byte: new output control register Value valid only after next INIT ioctl.
GCANSETMODE	Set internal/external transceiver selection	1 data byte: new transceiver selection (effective immediately) *(Version 1.1 and above)*
		1 data byte: current transceiver selection

GCANGETTRANS	Get internal/external transceiver selection	0 - internal transceiver 1 - external transceiver
GCANSETTRANS	Set internal/external transceiver selection	1 data byte: new transceiver selection (effective immediately). As above.
GCANSENDERR	Send an error frame. (The error level must have its least significant bit set for this to work.)	None
GCANRSETSTDID	set remote standard ID	None
GCANRSETEXTID	set remote extended ID	None
GCANRSETDATA	set remote frame data	None
GCANRENABLE	enable remote object	None
GCANRDISABLE	disable remote object	None
GCANRSETMASKS	set remote frame masks	None
GCANSWSETMODE	Set transceiver mode	1 data byte: new mode. Bits control tool resistance, sleep mode, hi voltage mode, hi speed mode. See dev_527.h for bit assignments.

GGETERRLEV	Get driver error level reporting	1 data byte: current mode
		Modes include (values may be ORed):
		   0 - Report Bus Warn and Bus Off conditions
		   1 - Report all bus errors
		   2 - Report data overrun
		   4 - Report arbitration lost

GSETERRLEV	Set driver error level reporting	1 data byte: new mode (effective after next INIT ioctl). As above.

GSJAGETLISTEN	Get listen/normal mode	1 data byte: current mode
		   0 - Normal
		   1 - Listen only

GSJASETLISTEN	Set listen/normal mode	1 data byte: new mode Mode changes only after next INIT ioctl. As above.

GSJAGETSELFTEST	Get self test/normal mode	1 data byte: current mode
		   0 - Normal
		   1 - Self test

GSJASETSELFTEST	Set self test/normal mode	1 data byte: new mode Mode changes only after next INIT ioctl. As above.

GSJAGETXMITONCE	Get transmit once/normal mode	1 data byte: current mode
		   0 - Normal
		   1 - Transmit once

GSJASETXMITONCE	Set transmit once/normal mode	1 data byte: new mode Mode changes immediately. As above.
GCANSWGETMODE	Get transceiver mode	1 data byte: current mode. Bits indicate tool resistance, sleep mode, hi voltage mode, hi speed mode.

		See [dev_527.h](dev_527.h) for bit assignments.
GCANSWSETMODE	Set transceiver mode	1 data byte: new mode. Bits control tool resistance, sleep mode, hi voltage mode, hi speed mode. See [dev_527.h](dev_527.h) for bit assignments.

Notes:

- While in the listen mode no data may be transmitted and no Acks are appended to received messages. All data on the bus is received as normal. If a message is not Ack'ed by another node, it is not valid message and is not be available to be read. If the error level reporting is set to report bus errors, a GCANACK (Ack error) event will be reported for each unacknowledged message on the bus.

- While in the self-test mode, the channel appends an Ack to each message it transmits on the CAN bus. This allows the channel to be the only device on a CAN network.

- While in the transmit once mode, all message transmissions are attempted only once. If arbitration is lost, the message is not transmitted. If an Ack error occurs for a transmitted message and the transmit error count on the SJA1000 chip (available as register 15 (0x0F) via the generic GETREG ioctl) is greater than 127, an Ack is appended to the transmitted message. If the transmit error count is less than 128, an Ack is not appended.

- Individual error frames may be generated by transmitting a message with the invalid 11-bit ID **0xff 0xff**, or the invalid 29-bit ID **0xff 0xff 0xff 0xff.** Error level reporting must have its least significant bit set to generate error frames.

- Reception of a CAN frame with the RTR bit set causes the MODE_REMOTE bit to be set in the mode element of the Data Header of an FT_DATA frame sent to Gryphon clients. Likewise setting the MODE_REMOTE bit in the mode element of the Data Header of an FT_DATA frame sent to the Gryphon server causes it to send a CAN frame with the RTR bit set.

- High voltage wake-up messages are flagged as such in received messages by setting the GCANSWHIVOLT bit in the stat field in the FT_DATA header.
  High voltage messages may be sent two different ways:

  - Set the GCANSWHIVOLT bit in the FT_DATA stat field of a transmitted message

  - Turn on HI VOLTAGE by use of the GCANSWSETMODE IOCTL.

  The first method allows sending of individual high-voltage wake-up frames, but may result in a delay before and after the message as the voltage is switched.
  The second method is preferable if more than one high-voltage wake-up frames are to be set; high-voltage will remain ON until turned off by means of another GCANSWSETMODE IOCTL.