

Gryphon Hardware Information: Dual SJA1000 Fault Tolerant CAN card

External HD-15 connector pinout

Note: We recommend that you not hot swap the connector on this module. We recommend that you turn off power to the Gryphon and the connected device before connecting or disconnecting to this connector.

Pin #	Channel	Assignment
01	2	CAN-LOW
02	1	CAN-LOW
06	2	CAN-HIGH
07	1	CAN-HIGH
10	Both	V- in / GND
11	2	VREF-
12	1	VREF-
13	2	VREF+
14	1	VREF+
15	Both	V+ in (6-32 VDC)

VREF+

VREF+ should be connected to the network nodes' supply voltage. This reference input is used to detect faults; this connection is required for proper network communication.

CAN BUS transceivers, load resistors

The SJA1000FT card utilizes Philips TJA1054 fault-tolerant CAN transceivers.
The SJA1000FT card includes 1K bus load resistors.

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	Acknowledgement 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)
GSJATRIGSET	Input trigger made transition and the current value is 1. (See documentation for Ioc1 GSJAGETTRIGCTRL below for more information.)
GSJATRIGRESET	Input trigger made transition and the current value is 0. (See documentation for Ioc1 GSJAGETTRIGCTRL below for more information.)
GSJAFT	Fault tolerant transceiver changed state (1 data byte for current state: no error/error 0/1)

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
<code>GCANGETBTRS</code>	Get SJA1000 BTR values	2 data bytes returned: current BTR0 and BTR1 values
<code>GCANSETBTRS</code>	Set SJA1000 BTR values	2 data bytes: new BTR0 and BTR1 values Set values valid only after next INIT ioctl.
<code>GCANGETBC</code>	Get SJA1000 Output Control Register	1 data byte: current output control register
<code>GCANSETBC</code>	Set SJA1000 Output Control Register	1 data byte: new output control register Value valid only after next INIT ioctl.
<code>GGETERRLEV</code>	Get driver error level reporting	1 data byte: current mode Modes include (values may be ORed): 0 - No error reporting 1 - Report bus warning and bus error 2 - Report data overrun 4 - Report arbitration lost
<code>GSETERRLEV</code>	Set driver error level reporting	1 data byte: new mode (effective after next INIT ioctl). As above.
<code>GSJAGETFTTRANS</code>	Get fault tolerant transceiver state	1 data byte: current transceiver state 0 - disabled 1 - enabled
<code>GSJASETFTTRANS</code>	Set fault tolerant transceiver state	1 data byte: new transceiver state (effective immediately). As above.
<code>GSJAGETFTERROR</code>	Get current fault tolerant error status	1 data byte: fault tolerant error status 0 - no error 1 - error
<code>GCANSENDERR</code>	Send an error frame. (The error level must have its least significant bit set for this to work.)	None
<code>GSJAGETLISTEN</code>	Get listen/normal mode	1 data byte: current mode 0 - Normal 1 - Listen only
<code>GSJASETLISTEN</code>	Set listen/normal mode	1 data byte: new mode Mode changes only after next INIT ioctl. As above.
<code>GSJAGETSELFTEST</code>	Get self test/normal mode	1 data byte: current mode 0 - Normal 1 - Self test
<code>GSJASETSELFTEST</code>	Set self test/normal mode	1 data byte: new mode Mode changes only after next INIT ioctl. As above.
<code>GSJAGETXMITONCE</code>	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.
GSJAGETTRIGSTATE	Get input trigger state	1 data byte: current state 0 - Reset 1 - Set
GSJAGETTRIGCTRL	Get input trigger control	1 data byte: current mode 0 - Do not send event 1 - Send only 1 event 2 - Send event for each 0 to 1 and 1 to 0 transition
GSJASETTRIGCTRL	Set input trigger control	1 data byte: new mode Mode changes immediately. As above.
GSJAGETOUTSTATE	Get digital output state	1 data byte: current state 0 - Reset 1 - Set
GSJASETOUTSTATE	Set digital output state	1 data byte: new state State changes immediately. As above.
GSJAGETFILTER	Get the filter	3 or 5 data bytes depending on the length of the filter Byte 1: number of header bytes = 2 Bytes 2 and 3: the standard header to match, in network order, high byte first - OR - Byte 1: number of header bytes = 4 Bytes 2 thru 5: the extended header to match, in network order, high byte first
GSJASETFILTER	Set the filter	3 or 5 data bytes depending on the length of the filter Filter changes immediately. As above.
GSJAGETMASK	Get the filter mask	3 or 5 data bytes depending on the length of the mask Byte 1: number of mask bytes = 2 Bytes 2 and 3: mask for a standard header, in network order, high byte first - OR - Byte 1: number of mask bytes = 4 Bytes 2 thru 5: mask for an extended header, in network order, high byte first
GSJASETMASK	Set the filter mask	3 or 5 data bytes depending on the length of the mask Mask changes immediately. As above.

Notes:

- While in the listen mode no data may be transmitted and no acks are sent in response to received messages. All other data on the bus is received as normal.

- While in the self-test mode, the channel sends an ack after 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 subsequent message transmissions are attempted only once. If arbitration is lost or if a bus error of some kind occurs, the message is not transmitted.
- The ability to pulse the trigger output based on an incoming message's header (CAN ID) is provided by a single filter and mask that can be set and read via the GSJASETFILTER, GSJAGETFILTER, GSJASETMASK and GSJAGETMASK ioctls described above. This filter does not affect data flow between the CAN bus and the Gryphon server or any Gryphon client. This filter is separate and distinct from the filtering that may be configured by a Gryphon client using the CMD_CARD_ADD_FILTER command.
- Setting the filter causes the mask to be set to 0x07FF or 0x1FFFFFFF which means that all bits of a conforming message must match the filter. If one or more bits should be ignored, set those bits in the mask to zero.
- Only one filter and mask are available per card. In order to cause the output trigger to pulse, a conforming message must arrive on the channel used to specify the filter and mask. In other words, with a filter and mask set, a conforming message on one channel will cause the trigger output to pulse, but an identical message on the other channel will not pulse the trigger output.
- 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.