Copley J1939 Implementation

Introduction

J1939 is a CAN based network protocol defined by the SAE (Society of Automotive Engineers) which is primarily used in on-vehicle communication networks for trucks, buses, farm equipment, etc. The network protocol itself is fairly simple and primarily consists of single CAN messages which use predefined network ID values to identify the type of data they contain. The J1939 standard defines the network ID values for standard messages.

The J1939 standard message types are primarily targeted at vehicle specific systems such as engine, transmission, brake control, etc. These applications generally fall outside the normal operation of Copley drives and therefore Copley drives do not respond to or generate any of these standard message types. J1939 also reserves several CAN message ID values for manufacturer specific proprietary messages. These are the message types used by Copley drives.

Firmware requirements

Copley's FPGA based plus drives added support for J1939 starting with version 4.80 firmware. Copley's ARM processor based standard drives added support for this protocol in version 2.06 firmware.

The J1939 protocol can be enabled by setting drive parameter 0x121 (network options) to a value of 2.

Node ID

Every device on a J1939 network should have a unique 8-bit node ID value. This value is used as part of the CAN message ID to identify the source and destination of the message.

Copley drives use parameter 0xC1 to set the network ID and bit rate. Please refer to the parameter dictionary document for details about this parameter.

CAN message ID format

Every message sent over the CAN bus starts with a message ID value. The message ID value identifies the type of message and also it's priority on the bus. Lower message ID values have a higher priority then higher ID values and if two message are being transmitted at the same time then the one with the lower message ID will prevail.

The CAN bus specification defines two message ID formats, an 11-bit standard ID format and a 29-bit extended ID. J1939 only uses the 29-bit extended format, any message sent with an 11-bit ID value will be ignored by J1939 devices including Copley drives running in J1939 mode.

The 29-bit CAN message ID is broken down into a number of different fields when used in a J1939 environment.

Bits	Description
0-7	Source address. These bits identify the network node that transmitted the message.
8-15	Destination address. For Copley drives these bits give the node number that the message is being transmitted to.
16-23	Message format. These bits identify the type of message. Copley drives currently only support the proprietary format value of 239.
24	Data page. This bit is currently set to 0 for any message used by Copley drives.
25	Extended data page. This bit is always zero.
26-28	Priority. These bits can be used to set the priority of the message. Lower values will result in higher priority. Copley drives will accept any value in these fields and will respond to the message using the same priority value.

Copley proprietary message format

Copley drives running in J1939 protocol mode will only respond to a message ID with the correct node ID in bits 8-15, the value 239 (0xEF) in bits 16-23, and zeros in bits 24 and 25. Any source address value is accepted and any priority value is accepted.

All messages sent to Copley drives must include at least one byte of data. This first byte of message data identifies the command that the Copley drive should execute. The format of any additional data sent with the command is dependent on the value of this command byte. Any message sent to a Copley drive with no data or an unsupported command code will be ignored.

Most commands to Copley drives will result in a response message. The CAN ID used in the response will have the source and destination node ID value swapped and use the same priority as the command message. Responses are always at least two bytes long, the first byte consists of the command code with bit 7 set and the second byte contains an error code (0 on success). Any additional data sent with the response will depend on the command that was executed.

0 - No operation

This command can be used to reset the watchdog timer on the drive and perform no other action. The drive will not transmit any message in response to this command code.

Command format:

Data bytes	Description
0	Command code. Must be 0x00.

Response format:

No response.

1 – Set parameter

This command can be used to set any 16-bit or 32-bit drive parameter.

Command format:

Data bytes	Description
0	Command code. Must be 0x01.
1-2	Parameter ID. This identifies which drive parameter to set. The Copley parameter dictionary lists the available drive parameters. Bits 0-9 identify the parameter number. Bit 12 should be set when setting a value stored in flash and clear when setting a value stored in RAM.
3-6	Parameter value. For 16-bit parameters the value is passed in bytes 3-4. For 32-bit parameters the value is passed in bytes 3-6. An error will be returned if insufficient data is passed for the parameter being set.

Response format:

Data bytes	Description
0	Response code. Fixed value of 0x81.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

2 – Get parameter

This command can be used to read any 16-bit or 32-bit drive parameter.

Command format:

Data bytes	Description
0	Command code. Must be 0x02.
1-2	Parameter ID. This identifies which drive parameter to read. The Copley parameter dictionary lists the available drive parameters. Bits 0-9 identify the parameter number. Bit 12 should be set when reading a value stored in flash and clear when reading a value stored in RAM.

Response format:

Data bytes	Description
0	Response code. Fixed value of 0x82.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.
2-5	Parameter value. 16-bit parameters will return 2 bytes of data, 32-bit parameters will return 4 bytes.

3 – Set indexer register

This command can be used to set a 32-bit indexer register.

Command format:

Data bytes	Description
0	Command code. Must be 0x03.
1	Register number. The drive has 32 indexer registers numbered from 0 to 31. This byte identifies which indexer register to access.
3-6	Register value. These bytes give the value to be written to the register.

Response format:

Data bytes	Description
0	Response code. Fixed value of 0x83.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

4 – Read indexer register

This command can be used to read a 32-bit indexer register.

Command format:

Data bytes	Description
0	Command code. Must be 0x04.
1	Register number. The drive has 32 indexer registers numbered from 0 to 31. This byte identifies which indexer register to access.

Response format:

Data bytes	Description
0	Response code. Fixed value of 0x84.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.
2-5	Value of indexer register.

16 – Trajectory update

This command can be used to start a position mode move, update the parameters of a trajectory currently in progress, or abort a move. It's comparable to the T command described in the ASCII programmer's manual.

Command format:

Data bytes	Description
0	Command code. Must be 0x10.
1	Trajectory command code. The following values are supported: 0 – Abort the move in progress. 1 – Start a new move or update the parameters used by the current move. 2 – Start a homing sequence.
2	This optional byte can be used to perform a trajectory update on multiple axes of a multi-axis drive. The byte is a bit-mask with one bit for each axis. Bit 0 for axis 1, bit 1 for axis 2, etc. If zero or omitted then only the addressed axis is updated.

Response format:

Data bytes	Description
0	Command code. Fixed value of 0x90.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

17 – Start move

This command can be used to start a move to a specific position or update the position of a move in progress.

When this command is executed it performs the following functions:

Set parameter 0xCA (trajectory generator position) to the value passed in bytes 1-4 of the message.

Set parameter 0x24 (desired state) to either 21 or 31 depending on whether the motor in use is a servo motor or stepper motor.

Perform a trajectory update to either start a new move or update the parameters of the move currently in progress.

Several drive parameters control the type of move that this command executes. These parameters should be either stored in flash memory or set using command 1 prior to executing this command. Some important parameters are:

0xC8: Trajectory configuration. Identifies the type of trajectory generator to use and whether the position passed with this command is an absolute position or relative move.

0xCB: Velocity limit used during the move

0xCC: Acceleration limit.

0xCD: Deceleration limit.

0xCE: Jerk limit for S-curve moves.

Command format:

Data bytes	Description
0	Command code. Must be 0x11.
1-4	Position to move to or distance to move.

Response format:

Data bytes	Description
0	Command code. Fixed value of 0x91.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

18 – Velocity command

This command is used to configure the drive to run in velocity mode with the specified velocity command.

When this command is executed it performs the following functions:

Set parameter 0x2F (Programmed velocity) to the value passed in bytes 1-4 of the message.

Set parameter 0x24 (desired state) to 11 (programmed velocity mode).

Command format:

Data bytes	Description
0	Command code. Must be 0x12.
1-4	Velocity command for velocity loop. Passed as 32-bit signed integer value in 0.1 encoder count/second units.

Response format:

Data bytes	Description
0	Command code. Fixed value of 0x92.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

19 – Velocity command (floating point)

This command is identical to command 18 except that the velocity value passed with the command is passed as a floating value.

Command format:

Data bytes	Description
0	Command code. Must be 0x13.
1-4	Velocity command for velocity loop. Passed as 32-bit floating point value in encoder count/second units.

Response format:

Data bytes	Description
0	Command code. Fixed value of 0x93.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

20 – Current command

This command is used to configure the drive to run in current mode with the specified current command.

When this command is executed it performs the following functions:

Set parameter 0x02 (Programmed current) to the value passed in bytes 1-2 of the message.

Set parameter 0x24 (desired state) to 1 (programmed current mode).

Command format:

Data bytes	Description
0	Command code. Must be 0x14.
1-4	Current command for current loop. Passed as 16-bit signed integer value in 0.01 Amp units.

Response format:

Data bytes	Description
0	Command code. Fixed value of 0x94.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

21 – Current command (floating point)

This command is identical to command 20 except that the current value passed with the command is passed as a floating point value.

Command format:

Data bytes	Description
0	Command code. Must be 0x15.
1-4	Current command for current loop. Passed as 32-bit floating point value in units of Amps.

Response format:

Data bytes	Description
0	Response code. Fixed value of 0x95.
1	0 on success. On failure an error code will be returned here. A list of error codes can be found in the Copley ASCII programmer's guide.

Watchdog timer

When running in J1939 mode the drive supports an optional watchdog timer feature which can be used to abort motion in the event of loss of communications over the network.

Two drive parameters must be set to enable the use of the watchdog timer:

0x10C – This parameter gives the heartbeat time in millisecond units. Set to 0 to disable the use of the heartbeat feature.

0x159 – This parameter configures the action the drive takes in the event of a heartbeat timeout.

0 – No action. This setting disables the heartbeat timer.

1 – Disable the drive on heartbeat timeout

2 – Abort any motion in progress and hold position on heartbeat timeout

When the watchdog timer is enabled, the drive must be addressed with a valid command periodically to prevent the timer from expiring. If the timer expires the drive will take the programmed action. Once a new command is received the drive will resume normal operation.

Revision History

Date	Version	Revision
07/25/2022	Rev 00	Initial release