

Duet 3 CAN protocol

Overview

Duet 3 uses CAN-FD to connect expansion boards to Duet 3 main boards and to coordinate motion and other command between them.

This document describes the high level operation of the CAN communication. The CAN-FD messages used to implement this can be found in file `CANLib/src/CanMessageFormats.h` and the corresponding CAN IDs can be found in `CANLib/src/CanId.h`.

Nomenclature

- “CAN” in this document refers to ISO-standard CAN-FD. Plain CAN is not supported because the maximum message length (8 bytes) is too limiting. Pre-ISO versions of CAN-FD are not supported.
- “Master” in this document refers to the board on the CAN bus whose job it is to receive GCode (or other) commands and to coordinate the actions of all boards to execute them. There must be exactly one master.
- “Slave” in this document refers to a CAN expansion board which executes commands received from the master. There may be many CAN slaves.
- A “request” message is a message sent by the master to a particular slave.
- A “response” message is a message sent by a slave in response to a request from the master.
- A “broadcast” message is a message sent by either the master or a slave and addressed to all boards.

Speeds

Normal data rate

The default speed is 1Mbps. Where a different speed is needed (e.g. a lower speed to handle increased distances), some type of configuration mechanism will be needed to set all boards to a lower speed.

For Duet 3 boards, the procedure for changing speed is as follows:

- Connect the slave boards to a Duet 3 main board, with all of them running at the default 1Mbps initially.
- Use the M952 command (https://duet3d.dozuki.com/Wiki/Gcode#Section_M952_Set_CAN_FD_expansion_board_address_and_or_normal_data_rate) to change the data rate of each slave. The change will not take effect until the slave is rebooted.
- The master data rate can then be changed and all boards rebooted.
- Established that communication at the new data rate is working.

- The boards can now be relocated to their intended locations.

Fast data rate

The bit rate switching (BRS) facility of CAN-FD is not currently used; but we plan to use it in future. When it is used, first communication between the master and slave boards will not use BRS. The master will use a CAN message to inform the slave what the fast speed will be when BRS is used, and the slave will acknowledge this. After that, the master may choose to use BRS with that fast data rate when sending messages to that slave, and the slave may use BRS when sending replies regardless of whether the master has used BRS.

High-importance short messages such as Emergency Stop shall be sent without using BRS.

Time synchronisation

In order to synchronise movement and other operations, all CAN-connected boards must use the same sense of time.

Synchronisation is achieved as follows. Duet electronics use a Step Clock to time stepper motor operations. For Duet 3 electronics, the frequency of this step clock has been standardised at 48MHz. This frequency can be generated by all Duet 3 boards, including boards based on the SAMC21 processor (one of the lowest-cost microcontrollers that offers an integrated CAN-FD controller). At regular intervals, the master broadcasts its time as a 32-bit number of step clocks since startup. All motion and other commands with precise timing requirements are tagged with the time (as a step clock number) at which they should start. Slave boards have their own step clock counters, and use appropriate means (e.g. a software frequency locked loop) to compute and maintain the difference between master time and local time. Then, when a command is received which includes a start time, that start time must be converted to local time. In the event that a slave does not use a 48MHz step clock, it would also need to convert between master step clock tick rate and its own step clock tick rate.

Assumptions

- CAN messages that are addressed to a single node are never lost without the sending node knowing. So far, this has proved to be a reasonable assumption. However, in some areas the protocol makes allowances for the possibility that this may not always be true, so that lost messages can be detected. For a message to be lost and the sender not to know, the intended recipient would have to not receive the message intact **and** the bus would have to pick up enough noise to change a recessive ACK bit to an active one.
- A broadcast message may not have been received by all CAN boards. This is because the ACK bit merely indicates that at least one board has received the message. Important broadcast messages are repeated. Other broadcast messages (e.g. temperature reports) are repeated at frequent intervals during normal operation, so failing to receive one occasionally does not matter.
- CAN messages may occasionally be duplicated. Many CAN messages are idempotent, so duplication is of no consequence. Messages that are not idempotent are protected against duplication. For example, a movement message that has the same start time as the previous movement message is assumed to be a duplicate.

- Received CAN messages have not been corrupted. This is a reasonable assumption, because the CAN protocol includes CRC checking.

Message categories

All CAN messages fall into one of the following categories.

Request/response pairs

The master send a request to a slave, and the slave responds with one or more response fragments. The request includes a Requester ID. The slave repeats this in the response, so that even in the event of lost or duplicate messages, each response can be paired with the corresponding request.

Responses normally have a standard form, which is a status code (type GCodeResult) accompanied by text. The text may be too long to fit in a single CAN packet, so the response also carries a fragment number and a “last fragment” indication.

Request/response pairs typically arise from configuration commands.

Time sync broadcasts

This is a short, high-priority message broadcast by the master. The message carries the time (in step clocks) at which it was placed in the buffer for sending, and also the time at which the transmission of the previous time sync broadcast was time-stamped by the CAN hardware.

Time sync messages are sent and received using dedicated CAN buffers so that they bypass the transmit and receive FIFOs.

Status messages

These are messages that can be sent by both the master and slaves to provide information about the status of the board. For example, each board sends periodically the temperatures of all temperature sensors that have been configured on that board. Status messages generated by slaves may be either broadcast messages, or messages addressed to the master.

Announcements

When a board starts up, it announced its presence via a broadcast message. A slave board will continue to announce its presence until it receives a response from the master.

To be continued

Last updated 2021-08-16