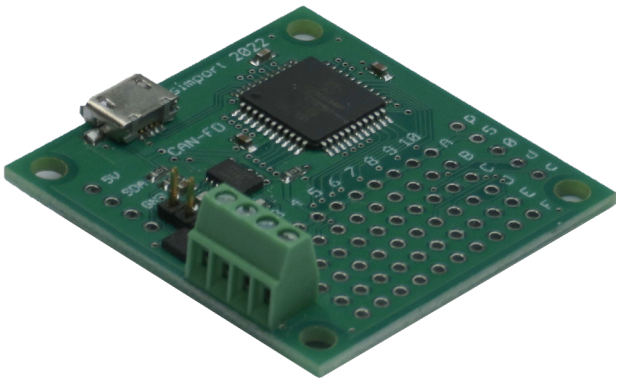
CAN in Simulation

# Interface Module





**Rev. 1.0**

The interface module actually features two distinct interfaces: a ***Keyboard*** and a ***Generic HID Device***.

When a keyboard message is received over the CAN bus, it is translated into a keystroke, mimicking the behavior of a regular keyboard. All other messages are encapsulated within a 15-byte report of the generic Human Interface Device (HID). This report can be easily parsed by any program capable of reading HID reports.

| ***Byte*** | ***Description*** |
|---|---|
| 0 | 1 (Generic HID) or 2 (Keyboard) |
| 1 | CAN-ID (High Byte) |
| 2 | CAN-ID (Low Byte) |
| 3 | CAN-Data (Byte 0) |
| 4 | CAN-Data (Byte 1) |
| 5 | CAN-Data (Byte 2) |
| 6 | CAN-Data (Byte 3) |
| 7 | CAN-Data (Byte 4) |
| 8 | CAN-Data (Byte 5) |
| 9 | CAN-Data (Byte 6) |
| 10 | CAN-Data (Byte 7) |
| 11 | Timestamp (Upper Byte) |
| 12 | Timestamp (High Byte) |
| 13 | Timestamp (Middle Byte) |
| 14 | Timestamp (Low Byte) |

Bytes 3 to 10 in the report mirror the CAN message data transmitted with the CAN-ID specified in Bytes 1 and 2 of the report above:

| CAN ID | node ID | data type | service code | message code | data byte 0 | data byte 1 | data byte 2 | data byte 3 |
|---|---|---|---|---|---|---|---|---|
| xxxx | node | type | item | num | data | data | data | data |

The timestamp is appended by the interface module as the message is received. It is a free running 32-bit counter and increments every millisecond. The counter starts from zero when the module is powered up and overflows to zero when the maximum count is reached.

In order to send a message over the CAN bus, a 15-byte report has to be sent from the PC host. Byte 0 (report ID) should be set to <1>, and the timestamp bytes are disregarded, while the remaining bytes align with the established scheme above.

The Interface module is powered through the USB Micro connector. The CAN bus connector provides a 5V output that can be used to power additional modules. The USB port has a maximum current draw limit of 500 mA.

## USB connection

When you connect the Interface Module to the PC it is recognized as two devices: ***CiS-Interface*** and ***CiS-Keyboard***. No special drivers are necessary.

The ***CiS-Interface*** is a generic HID (Human Interface Device) that transparently relays all CAN messages it receives through its dedicated CAN connector, excluding those falling within the CAN-ID range from 0x720 to 0x727. These messages are routed to the ***CiS-Keyboard***.

The ***CiS-Keyboard*** is a virtual keyboard that transmits keystrokes, following the instructions within the CAN message, to the operating system. For more details, please refer to the manual for the keyboard module.

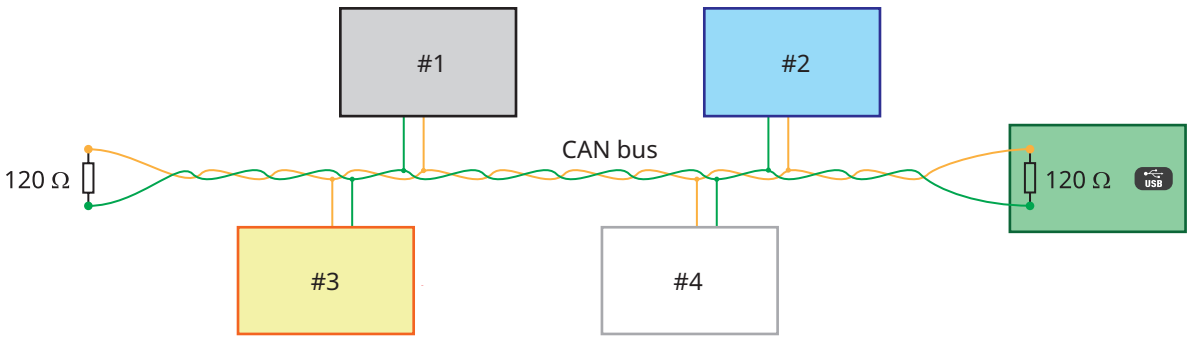
## The CAN bus

The CAN bus is a two-wire network in which all devices are connected via short stubs to a common line. The common line must meet specific conditions to ensure the reliable transmission of data at high rates. ***CAN in Simulation*** defaults to a data rate of 1 Mbit/s, but it can be adjusted to 500 kbit/s, 250 kbit/s, or 125 kbit/s as needed.

The physical cable typically employed for CAN transmission is a twisted pair with a 120-ohm characteristic impedance. To prevent signal reflections and, consequently, signal distortion, it is essential to terminate the cable at both ends with a 120-ohm resistor. At a data rate of 1 Mbit/s and with high-quality cabling, a maximum cable length of 40 meters is achievable.

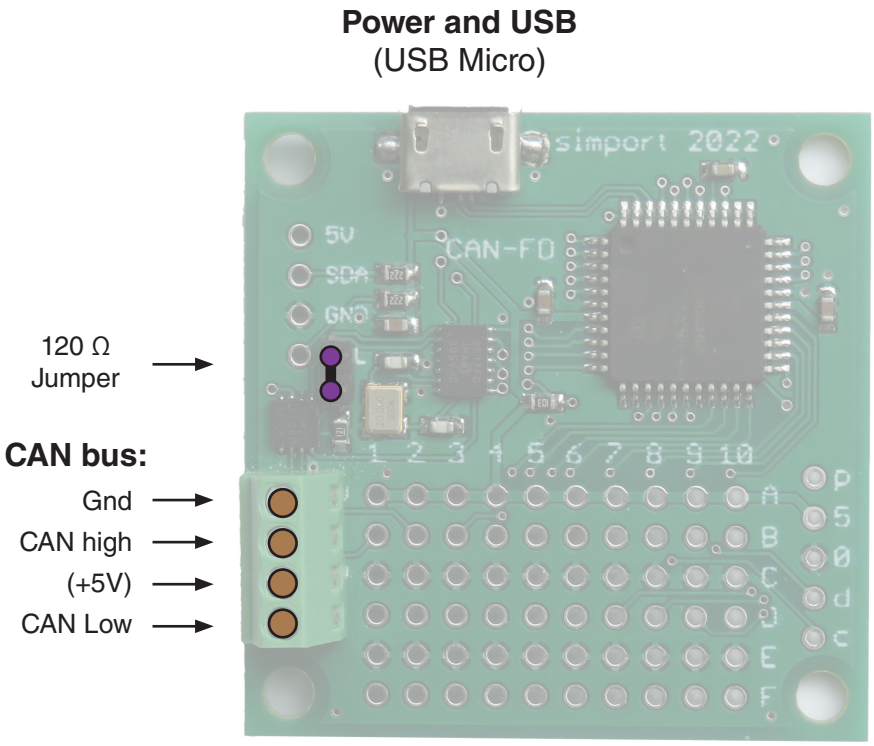
Nevertheless, for a cockpit within a typical home environment or a small flight simulator, the CAN bus cable is considerably less sensitive to quality concerns. However, the inclusion of 120-ohm resistors is a mandatory requirement.

For added convenience, each module comes with a built-in termination resistor, which can be connected or disconnected using a jumper. However, it‘s essential to keep in mind that these resistors are required only at the cable ends and should not be attached elsewhere along the cable.




Typical wiring involving four modules and a CAN-USB interface. The left termination resistor can be positioned on the final board in the chain, which is board #3.

## Board Layout




## Board Dimensions