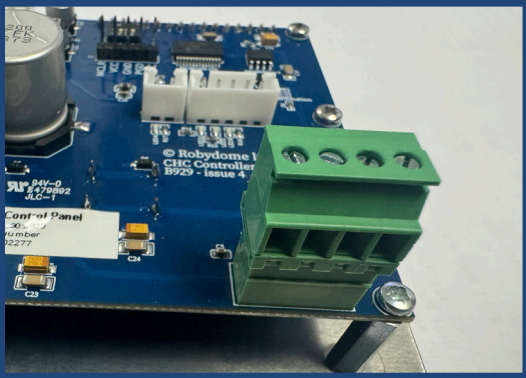
# Constant Humidity Controller API Connection Guide

The Constant Humidity Controller has an API that enables external control panels to control the system via an RS-485 ComBus connection. With this connection, commands can be sent to the system to allow changes to the set point and mode (relative humidity or temperature), whilst providing sensor readings, flame stages, and lockout information.

### Cable Requirements

You must use a Belden 9841 (or equivalent) cable, a low-capacitance, shielded, single-pair cable primarily used for RS-485 serial communication in industrial environments. Generally, this is a highly shielded 24 AWG single pair cable.




### Wiring at the CHC End

For the CHC connection, two cores are required, which should be connected to the A+ and B- connections. The two cores are wired, preferably with ferules, to the middle two points on the RS-485 port on the back of the CHC display (see photo).

For more information call:
+44 (0)1795 533903

### Port/Serial Settings

When connecting, ensure you use the following settings:

Bits Per Second: 9600
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

### Message Structure

**Start of message - Command code - [Data separator] - [Data] - End of message**

Note. Items in square brackets [] are optional depending on the message type.

For each element, you need to look for:

**Start of Message:** Line feed (10 decimal, OA hex.)
**End of Message:** Carriage return (13 decimal, OD hex.)
**Data Separator:** Colon (58 decimal, 3A hex.)

## Communication

Once you have established a connection with the CHC unit, you will begin receiving data. Messages are automatically sent whenever data changes. To facilitate messages being received, a “gate” is periodically opened (nominally for 500ms) during which time the unit guarantees not to transmit. To send a message to the unit, the sender must wait for a message indicating that the gate is open before sending the message.

It is worth noting that, in most cases, the data needs to be subscribed to and stored in a buffer, which is then read using the Start of Message and End of Message characters as delimiters, since many communicators bring down partial messages into the stream.

This can be achieved with a FIFO queue or buffer as the messages are sent at the time, the order will be maintained by the buffer.

## Messages/Command Codes

| Code | Function | Transmitted/Received | Data |
|---|---|---|---|
| G | Gate Open/Closed | Transmitted | 1 = Gate Open<br>0 = Gate Closed |
| H | Heater On/Off | Transmitted | 1 = Heater On<br>0 = Heater Off |
| I | Serial Number | Transmitted | 8 ASCII Digits |
| L | Lockout | Transmitted | 1 = Lockout<br>0 = No Lockout |
| M | Mode | Both | T = Temperature<br>R = RH |
| P | Power | Transmitted | 1 = Power On |
| Q | Query Unit Status | Received | No date (1) |
| R | RH Value | Transmitted | 2 ASCII Digits (2) |
| S | Set Point | Both | 2 ASCII Digits |
| T | Temp. Value | Transmitted | 2 ASCII Digits (2) |
| V | Valve (Stage Number) | Transmitted | 1 ASCII Digit (3) |

(1) Query unit status does not require data separator or data. When received, the unit will respond by sending all current values for messages H,I,L,M,S and either R or T depending on current mode.
(2) A sensor fault condition is transmitted as - - (2 dashes)
(3) The stage number 0 to 7 corresponds to the combination of valves 1,2 and 3