

To implement in DIN-MOD4 (first) then CFMini, by 18 Dec

2015

(to implement in all other CFLink units if possible)

Solo should have the CON command added too, with ability to compare the longer variable names already in use in Solo.

CFLink Conditional Commands

Conditional commands allow an action to be sent based on the evaluation of a variable state.

The conditional command is sent to the device that contains the variable we want to query the state of. Then an action is triggered if the condition evaluates true, which can target any device on the CFLink bus. **Variables stored on other devices CANNOT be compared in conditions.**

So there is a new command for this: TCFXCON

The CFX can be substituted with a specific 3 char model name as well. Eg. TMINCON to send a conditional command that only CF Mini devices will accept.

Syntax

Send: \xF2<ID>\xF3TCFXCON\xF4<CONDITION>\xF6<ACTION>\xF5\xF5

Reply: \xF2<ID>\xF3RCFXCON\xF4<BOOLEAN>\xF5\xF5

\xF2<ID>\xF3TCFXCON\xF4<VARIABLE><OPERATOR><VALUE>\xF6<TARGET_ID>:<COMMAND>:<DATA>\xF5\xF5

Send: \xF2<ID>\xF3TCFXCON\xF4A==1\xF600:TRLYSET:M2|P02:1\xF5\xF5

Reply: \xF2<ID>\xF3RCFXCON\xF4TRUE\xF5\xF5

- \xF6 splits the condition and the action. It is a single hex byte. The first \xF6 byte in the data should be considered the split location, so that the actual action can contain any byte including \xF6.
- <DATA> can contain anything, except \xF5\xF5, including additional colons.
- <VARIABLE> the data represents the variable type and variable character to get the value of, A-Z. This allows you to compare a variable value in the condition.
- <OPERATOR> can be one of the following symbol sequences (always 2 bytes)

- == **EQUAL TO**
- != **NOT EQUAL TO**
- >> **GREATER THAN**
- >= **GREATER THAN OR EQUAL TO**
- << **LESS THAN**
- <= **LESS THAN OR EQUAL TO**
- <VALUE> must be a digit, 0-65535. This is used (with the <OPERATOR>) to compare the state of the <VARIABLE>.
- <TARGET_ID> should be the CFLink ID of the unit receiving the command, in plain ascii (range 2 - EF). Single character will work (2-9) or two characters (02-EF). 00 = local device (ie. target the same device the condition is running on).
- **For each reply, a boolean TRUE or FALSE will be returned on each condition evaluation, as the reply data payload. RCFXCON TRUE for example.**

Variable storage

1. User (prefixed with U) - User can assign values in 00000-65535 range (two bytes).

Max of 26 variables per variable type can be stored in each device. Referenced as A through Z.

Syntax

Store user variable

\xF2<ID>\xF3CCFXVAR\F4<VARIABLE_CHAR>:<VALUE>\xF5\F5

- <ID> The CFLink ID of the device to store the variable on. If this is FF, then all devices on the bus would process the command and store the variable value.
- <VARIABLE_CHAR> Valid values are A-Z (case-sensitive). Internally, always store upper case. Single character only.
- <VALUE> Any value from 0-65535.
- STORING the variable gives a reply with the value stored - See Reply section below.

Query

\xF2<ID>\xF3QCFXVAR\F4<VARIABLE_CHAR>\xF5\F5

- <VARIABLE_CHAR> Valid values are A-Z (or a-z, case-insensitive), therefore A=a, etc.
- **Empty** = report all variables that have a value other than 0 as separate replies.

Reply

\xF2<ID>\xF3RCFXVAR\xF4<VARIABLE_CHAR>:<VALUE>\xF5\xF5

- <VARIABLE_CHAR> Valid values are A-Z (or a-z, case-insensitive), therefore A=a, etc.
- <VALUE> Value of the variable. 0-65535.