

Additions to the Xchecklist dataref expressions syntax

December 10, 2018

Basic Xchecklist syntax for dataref expressions is somewhat limiting and some of the resulting constructs are a bit cumbersome. For this reason we tried to extend the syntax to make those expressions more readable and flexible, while maintaining the backward compatibility.

The old syntax used the following system for the dataref expressions:

```
sw_item:text:dataref_expression:value
```

The **dataref_expression** can be either a dataref name or an expression combining several conditions using logical operators. For example:

```
sw_item:text:(sim/dataref1:1) && ((sim/dataref2:2) || (sim/dataref3:3))
```

The **value** field could contain expressions combining numbers, datarefs and even some functions to compute the actual value. For example:

```
sw_item:text:sim/dataref1:round(1+{sim/dataref2})
```

The first addition to the syntax is the possibility to use a constant instead of the **dataref_expression**. This makes it possible to compare a dataref expression (which can be only in the value part) to a constant. For example:

```
sw_item:text:1:closer_than({sim/dataref1}, {sim/dataref2}, 10)
```

The second addition is an extension of the syntax of the **value** part; now it can contain complete "tests". For example:

```
sw_item:text:({sim/dataref1} == 1) && ({sim/dataref2} != 2)
```

Note the double colon (effectively the **dataref_expression** part is missing) - that enables the new expression syntax! Also note that to avoid confusion concerning the priorities of the arithmetic vs. logical operators, the operands of the logical operators must be enclosed in the parentheses.

You can use usual comparison operators: == for equality, != for inequality, > for greater than, < for lower than, >= for greater or equal, <= lower or equal. Only bear in mind, that testing floating point values for equality or inequality

can give you unexpected results! For comparing floating point values preferably use the **closer_than** construct.

Hopefully these new possibilities will make the dataref expressions much more readable.