v1.0

757-DEFINE_MUTUALLY_EXCLUSIVE and Variables

9-Oct-20

Symptoms

If you have a set of variables that are mutually exclusive and you set one of the variables to a non-zero value by assignment, e.g. Var1 = 1 or the Studio Debug window, then the other variables in the set stay "on" i.e. non-zero.

DEFINE_VARIABLE

INTEGER var[4]

INTEGER x

DEFINE_MUTUALLY_EXCLUSIVE

(var[1],var[2],var[3],var[4])

DEFINE_PROGRAM

WAIT 20

{

x++; IF (x > 4) x = 1;

var[x] = x // This will not invoke the mutually exclusive magic

}

In the NetLinx code example above, all elements of var will eventually be non-zero.

Note: Axcess does not support making elements of an INTEGER array mutually exclusive.

Cause

This has always worked this way, even in Axcess.

Resolution

Use ON to set variables if they are members of a mutually exclusive set:

DEFINE_VARIABLE

INTEGER var[4]

INTEGER x

DEFINE_MUTUALLY_EXCLUSIVE

(var[1],var[2],var[3],var[4])

DEFINE_PROGRAM

WAIT 20

{

x++; IF (x > 4) x = 1;

ON[var[x]] // This will work as expected – only one element of var will have a value of 1 at any time

}

This issue does not occur with DEVCHAN's. Using ON or assigning them a non-zero value will work as expected:

DEFINE_DEVICE

dvRelay = 305:1:0

DEFINE_VARIABLE

INTEGER x

DEFINE_MUTUALLY_EXCLUSIVE

([dvRelay,1]..[dvRelay,4])

([dvRelay,5]..[dvRelay,8])

DEFINE_PROGRAM

WAIT 20

{

x++; IF (x > 4) x = 1;

ON[dvRelay,x] // This works as expected: only 1 relay of relays 1 to 4 will be on at a time

[dvRelay,x + 4] = x // This works as expected: only 1 relay of relays 5 to 8 will be on at a time

}