

Name:

None

Purpose:

Move mousepointer anywhere on screen, press left/right mouse button and or release some options in advanced tab. Such as doubleclick left mousebutton usable for e.i. opening shortcuts. Visit every pixel means that the mouse will take short path from to position on screen.

Data:

Utilize `<v"Variable">` to access variable values from all mandatory or optional text input fields. Employ `<s"expression">` for straightforward calculations. For instance, combining `<s<v"Variable">+1m>` will yield the value of "Variable" increased by 1. If the variable holds a value of 100, the result will be 101. To select a variable, press `<v>`, and a list of available variables will be presented. Choose the desired variable and press enter. Retrieve a separated variable by adding `..1`, where `..` acts as the separator. For instance, if the variable "x" holds values `x=1.2;1.4;8;2`, `<vx..1v>` will return the result 1.4 to the function. Additionally, you can read thread memory by using, for example, `<r1:20r>`, which returns the value in thread memory at address 1, subaddress 20. To read separated thread memory, use as with variable `<r1:20..1r>`.

Usage:

In the "Advanced" tab for all functions, there is a checkbox labeled "Breakpoint." Enabling this checkbox will cause the execution to pause if the "Debug" option is checked. It's important to note that the "Debug" feature is exclusively supported for the main program execution and does not apply to configurations running in threads.

Exceptions:

In the "Advanced" tab for all functions, there is an option to handle faults. A fault may arise, for instance, when a function anticipates a number but receives text instead. Various reactions to faults are available:

Ignore: The program continues as if nothing happened, though variable return data may be faulty.

Halt: All program execution stops when a fault occurs.

Ignore and Goto: Redirects to a specified part of the current configuration. Error text can be extracted by reading thread memory at address 4, subaddress 2000.

The default setting is "Halt and Report," meaning faults will stop the program and be logged in the error log. Exceptions, like entering non-numeric data in a numeric position, trigger this fault-handling mechanism.

Checking the "Failsafe" box will override all other options, allowing the program to continue regardless of faults and ignoring any specified fault-handling settings.

Limitations:

Can only move pointer in screen where macroware is started no support for multiple screens.

Example: