🟗 🗄 🍓 Quick search 🧈 🌮 🎽 🚳 🖗 🖗 🦓



Topic: DMD0234

The Do-more Simulator

The Do-more Designer programming environment includes a PC-based, Do-more controller simulator. This Simulator operates as a Virtual PLC with an Ethernetequipped CPU and 4 I/O modules. The Do-more Designer software interacts with the Simulator in the same manner as it would a 'real' Do-more controller, for example, an online session is established through an automatically-created Ethernet communication link called MySim.

The Simulator has a default System Configuration and a default Memory Configuration that can be changed as required. The I/O Configuration is fixed.

Do-more Designer projects are downloaded to the Simulator just as they would be to a 'real' controller. Putting the Simulator in RUN mode will cause it to execute the ladder logic in the project. The Simulator will stay in RUN mode for a maximum of 1 hour before reverting to PROGRAM mode.

| Do-more Sim | 16 DI | | 16 DO | | 8 Ana | alog Inputs | | 8 An | alog Outputs |
|--|--|---|--|---|--|--|--|-------------------------|------------------------------|
| DM v0.12.1 | X0 X8 | YC | Y8 | WX0 | | | 2036 ÷ | WY0 | 409 |
| SIM V0.13.1 | X1 X9 | Y1 | Y9 | WX1 | | | - 0 - | WY1 | 0 |
| PWR SerRA | X2 X10 | Y2 | Y10 | WX2 | | | - 0 ÷ | WY2 | 0 |
| ROM EMPX | X3 X11 | Y3 | Y11 | WX3 | | | - 0 ÷ | WY3 | 0 |
| FRR EthTX | X4 X12 | Y4 | ¥12 | WX4 | | | - 0 - | WY4 | 0 |
| | X5 X13 | YS | Y13 | WX5 | | | - 0 - | WY5 | 0 |
| RUN - | X6 X14 | Ye | Y14 | WX6 | | | - 0 - | WY6 | 0 |
| | | | | | | | | | |
| STOP - 1 | 0 C1 C2 | C3 C4 | 4 C5 C | WX7 | C8 C9 C1 | 0 C11 C12 | - 0 <u>÷</u> C13 C14 C | WY7 15 C16 C17 C18 C | 19 C20 C21 C22 C |
| STOP - 1 C Memory C Count | 27 X15 C1 C2 ers | C3 C4 | Y15 4 C5 C | WX7 | C8 C9 C1 | 0 C11 C12 | - 0 🛨 | WY7 | 0 19 C20 C21 C22 C s |
| C Memory C C Memory C Count | 2 Done | C3 C4 | 4 C5 C | WX7 | C8 C9 C1 lemory | 0 C11 C12 R Memory R0 0.848 | - 0 ± C13 C14 C | WY7 | 0 119 C20 C21 C22 C2 s |
| C Memory C C Memory C Count TO Acc 3 TI Acc 0 | 2 Done | C3 C4 | 4 C5 C | WX7 | C8 C9 C1 Iemory 0 0 | 0 C11 C12 R Memory R0 0.848 R1 83.333 | C13 C14 C | WY7 | 0 19 C20 C21 C22 C2 s |
| C Memory C Count COUNT COUNC COUNT C | x7 x15 | C3 C4 | 4 C5 C lemory 0 0 0 | WX7 C6 C7 D M D0 D1 D2 | C8 C9 C1 1emory 0 0 0 0 0 0 0 0 | 0 C11 C12 R Memory R0 0.848 R1 83.333 R2 0.000 | C13 C14 C MSG ERR SS0 | WY7 | 0 19 C20 C21 C22 C2 s |
| C Memory C Count TO .Acc 3 T1 .Acc 0 T2 .Acc 0 | x7 x15 | C3 C4 VM V0 V1 V2 V3 | 4 C5 C lemory 0 0 0 0 | WX7 26 C7 D M D0 D1 D2 D3 | C8 C9 C1 | 0 C11 C12 R Memory R0 0.848 R1 83.333 R2 0.000 R3 49.719 | C13 C14 C MSG ERR SS0 SS1 | WY7 | 0 119 C20 C21 C22 C2 s |
| C Memory C Count TO Acc 3 TT Acc (T2 Acc (Time | X7 X15 D C1 C2 ers 2 Done 0 Done 1 Done 1 Done | C3 C4 VM V0 V1 V2 V3 V4 | 4 C5 C lemory 0 0 0 0 0 | 00 C7 C7 D M D0 D1 D2 D3 D4 | C8 C9 C1 Iemory 0 0 0 0 0 0 | 0 C11 C12 R Memory R0 0.848 R1 83.333 R2 0.000 R3 49.719 R4 0.000 | C13 C14 C MSG ERR SS0 SS1 SS2 | WY7 | 0 119 C20 C21 C22 C s |
| C Memory C Count Count CTO Acc 3 CT1 Acc () CT2 Acc () Time T0 Acc () | X7 X15 C1 C2 ers 2 Done Done 0 Done rs 0 Done | C3 C4 VM V0 V1 V1 V2 V3 V4 V5 | V15 4 C5 C 1emory 0 0 0 0 0 0 0 0 0 | WX7 56 C7 D M D0 D1 D2 D3 D4 D5 | C8 C9 C1 Iemory 0 0 0 0 0 0 0 | 0 C11 C12 R Memory R0 0.848 R1 83.333 R2 0.000 R3 49.719 R4 0.000 R5 33.614 | 0 <u>+</u> C13 C14 C MSG ERR SS0 SS1 SS2 SS3 SS3 | WY7 | 0 19 C20 C21 C22 C2 s |
| C Memory C Count Count TO Acc 3 T1 Acc 0 T2 Acc 0 Time T0 Acc 0 T1 Acc 0 | x7 x15 C1 C2 ers 2 Done 0 Done 0 Done 1 C2 1 C2 | C3 C3 VM V0 V1 V2 V3 V4 V4 V5 V6 | Y15 C5 C | WX7 26 C7 D M D0 D1 D2 D3 D4 D5 D6 C7 | C8 C9 C1 Itemory 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 C11 C12 R Memory R0 0.848 R1 83.333 R2 0.000 R3 49.719 R4 0.000 R5 33.614 R6 0.000 | C13 C14 C MSG ERR SS0 SS1 SS2 SS3 SL0 SL0 ■ | WY7 | 0 19 C20 C21 C22 C s |

LEDs and Mode Switch

The LEDs and the Mode switch mimic operation of the front panel LEDs and the Mode Switch of a Do-more Controller.

Input Section

The Input section provides a way for the programmer to change the values of the discrete and analog input modules. The ability to write to these input locations can be enabled and disabled though the **Sim-> Enable Inputs** menu selection.

16 Discrete Inputs - there are 16 discrete inputs mapped into X0 - X15. Click a button once to turn that input ON (the button stays depressed), click it again to turn the input OFF.

8 Analog Inputs - there are 8 analog inputs mapped into WX0 - WX7. Click and drag the slider from left to right, or click the up and down arrows at the right edge to generate an input value of 0 to 4095. The current value will be displayed as a bar graph that fills from the left and will be displayed in numeric form beside the up/down arrows.

Output Section

The output section displays the current status of the simulator's discrete and analog outputs.

16 Discrete Outputs - there are 16 discrete outputs mapped into Y0 - Y15. If that output is ON the LED will be bright green, if it is OFF the output will be dark green.

8 Analog Outputs - there are 8 analog inputs mapped into WY0 - WY7. The current value (0 to 4095) will be displayed as a bar graph that fills from the left and will be displayed in numeric form at the right edge.

Memory Section

Because of space limitations the memory section cannot display the full complement of memory locations in the Simulator, only the current status of the following bit, numeric, and string elements in the simulator are displayed:

C Memory - the first 16 control relays (C0 - C23). If that output is ON the LED will be bright green, if it is OFF the output will be dark green

Counters - the current count (.Acc) and the Done bit of the first 3 Counters (CT0 - CT2)

Timers - the current time (.Acc) and the Done bit of the first 3 Timers (T0 - T2)

V Memory - he first 8 locations in the V Memory block (V0 - V7)

D Memory - the first 8 locations in the D Memory block (D0 - D7)

R Memory - the first 8 locations in the R Memory block (R0 - R7)

Strings - the system MSG and ERR strings, the first 4 Short Strings (SS0 - SS3), the first Long String (SL0)

Run Do-more Simulator

Clicking the Do-more/Sim button on the Online toolbar will open the Launch and Connect to Do-more Simulator dialog according to the following:

- If there is NOT a project currently open, Do-more Designer will launch the Simulator, then open a New Online project using the communication Link named MySim.
- If there IS a project opened (either online or offline), Do-more Designer will launch the Simulator AND will launch a new instance of Do-more Designer and open a New Online project using the communication Link named MySim. The current instance of Do-more Designer and its project are left intact.
- If there is an project currently using the MySim link and it is minimized or 'behind' other applications, Do-more Designer will bring the currently running
 instance of the Simulator to the front.

| | nulator is a separate | Windows app | lication that | runs the sam | e Do-mo | re Contro | Engine as in a D | o-more PLC. | |
|--|---|---|--|--|----------------------------------|--|---|---|----------------------|
| e Simulator ha | is buttons for discre | te inputs, ligh | ts for discret | e outputs, sli | ders for | analog inp | outs, and status | bars for analo | og outpu |
| e Simulator dis stem message | splays status for sor s, and ASCII string | ne of the inte elements. | rnal element i | v <mark>alues like ti</mark> n | ner and o | counter ad | cumulators, inte | rnal registers | and bits, |
| ere's even a R | lun/Term/Stop mode | switch and L | ED indicators | for Power, P | un, and | Communi | cation activity. | | |
| | Do-more Simulator | | | | | | (m) | 87 ×) | |
| | File Sim | | | | | | | | |
| | Do-more Sim 16 D | 1500 | 1 | 8 Analog Inputs | | | 8 Analog Outputs | | |
| | DH v0.12.2 X0 | xa YO YB | wxo — | | 805 - | WYO COW | | 3000 | |
| | Simv0.13.2 X1 | X9 Y1 Y9 | WX1 | <u> </u> | 1589 🛨 | WY1 🛄 | | 450 | |
| | BUN BOTX X2) | (10 12 12 12 12 | WX2 | | 0 🛬 | WY2 | | 0 | |
| | ROM ENER | | WX3 | | 0 3 | WY3 | | 0 | |
| | | | 1004 | | 0 | WY4 | | 0 | |
| | RUN - X6 X | (14 10 114 | wxs | | 0 4 | WY6 | | 0 | |
| | STOP - | (15) | WX7 | | 0 🔄 | WY7 | | 0 | |
| | C tiamary COL CT | | 05 07 03 09 | C10 C11 C12 | 11 OIE CI | 5 C16 C17 | C18 C19 C20 C21 C2 | 3 (223) | |
| | Counters | V Memory | D Memory | R Memory | 1 | 916 - 91 - 4 | Strings | | |
| | CTO Acc 42 Don | V0 7 | 000 | R0 0.000 | MSG | | | _ | |
| | CT1 Acc 0 Don | V1 14 | 01 0 | R1 0.000 | ERR DW | ide by zero in DiV | @0000061 | | |
| | C12 ACC 0 1020 | V2 21 V3 28 | D3 0 | R3 0.000 | 890 NO | w is the time for s | Ill good men to come to the a | id of thei | |
| | Timers | V4 0 | D4 0 | R4 0.000 | 592 | | | | |
| | TO Acc 0 Done | V5 0 | 05 0 | R5 0.000 | \$\$3 | | | | |
| | | V6 0 | 0 | R5 0.000 | SLO | | | | |
| | T1 Acc 0 Done | | | 157 0.000 | | | | Ext | |
| | T1 Acc 0 Prov T2 Acc 0 Prov | 1 <u>** 0</u> | 1.00 | | | | | | |
| | T1 Acc 0 Acc | | J | | | | | | |
| u connect to t | T1 Acc 0 Boot | o-more Design | her the same | way you do a | PLC. T | 'he Simula | tor has a "built-ir | "Ethernet po | ort with |
| J connect to t | he Simulator with Do | o-more Desigr r can automat | ner the same tically resolve | way you do its address (| PLC. T | 'he Si <mark>mul</mark> a predefine | tor has a "built-ir d MySim commun | "Ethernet po ications link). | ort with |
| u connect to t ur PC's Ethern | he Simulator with Do | o-more Desigr r can automat | ner the same tically resolve | way you do its address (| PLC. T see the | he Simula predefine | tor has a "built-ir d MySim commun | "Ethernet po ications link). | ort with |
| u connect to t ur PC's Ethern you shut dowr | he Simulator with Do net port, so Designed the Do-more Simulator | o-more Desigr r can automat | her the same tically resolve on, that's the | way you do its address (same as pov | PLC. T see the vering-do | 'he Simula predefine own your l | tor has a "built-ir d MySim commun PLC,Designer w | " Ethernet po ications link). ill get commur | nt with |
| u connect to t ur PC's Ethern you shut dowr ieout errors w | the Simulator with Do the tort, so Designer the Do-more Simulator with hen your "PLC" is po | o-more Desigr r can automat ator application | her the same tically resolve on, that's the | way you do its address (same as pov | PLC. T see the vering-do | 'he Simula predefine own your l | tor has a "built-ir d MySim commun PLC,Designer w | " Ethernet po iications link). ill get commur | ort with nication |
| u connect to t ur PC's Ethern you shut dowr eout errors w | the Simulator with Do the Simulator with Do the port, so Designer the Do-more Simula then your "PLC" is po | o-more Desigr r can automat ator application | ner the same tically resolve on, that's the | way you do its address (same as pov | PLC. T see the vering-do | 'he Simula predefine own your l | tor has a "built-ir d MySim commun PLC,Designer w | " Ethernet po iications link). ill get commur | ort with nication |
| u connect to t ur PC's Ethern you shut dowr eout errors w Do you u | 11 Jacci 0 Boot | o-more Design r can automat ator applicatio owered down | ner the same tically resolve on, that's the tor application | way you do a its address (same as pov on and conne | PLC. T see the vering-do | he Simula predefine own your l with anoth | tor has a "built-ir d MySim commun PLC. Designer w er instance of Dc | " Ethernet po ications link). ill get commur o-more Design | nication |
| u connect to t ur PC's Ethern you shut dowr eout errors w Do you t | 17 Jacci 0 Boot 17 Jacci 0 Boot the Simulator with Dr het port, so Designe In the Do-more Simul hen your "PLC" is po wish to launch the D | o-more Design r can automat ator application owered down | ner the same tically resolve on, that's the ator applicatio | way you do a its address (same as pov on and conne | a PLC. T see the vering-do | he Simula predefine own your l with anoth | tor has a "built-ir d MySim commun PLC. Designer w er instance of Dc | n" Ethernet po iications link). ill get commur o-more Design | nication nication |
| u connect to t ur PC's Ethern you shut dowr eout errors w Do you t Do you t | 11 Acci 0 Boot | o-more Design r can automat ator application wered down o-more Simula | ner the same tically resolve on, that's the ator applicatio | way you do a its address (same as pov on and conne | a PLC. T see the vering-do | he Simula predefine own your I with anoth | tor has a "built-ir d MySim commun PLC. Designer w er instance of Dc | n" Ethernet po ications link). ill get commur o-more Design | nication er? |

Do not ask again - just launch and connect next time - check this option to bypass this dialog the next time the Do-more Sim button on the toolbar is clicked.

Click **OK** to launch the Simulator.

Click Cancel to close the dialog without launching the Simulator.

Do-more PID Process Simulator

The Do-more Simulator includes a PID Process simulator that can be used to demonstrate the process control abilities of the controller, or for testing potential changes to existing control solutions by working with the process control instructions in a simulation environment before deploying the control solution to a Do-more controller. The PID process simulator uses a first order filter plus a dead time calculation to provide the simulated PID loop response.

The PID Process Simulator is opened by clicking 'Sim' on the toolbar of the Simulator and selecting Setup PID Process Simulator.

| 🥐 🖸 | o-more Simulator | 1 | |
|-----|------------------|-------|-------|
| Fil | Sim | | |
| 1 | Do-more Sim | 16 DI | 16 DO |

The PID Process Simulator will use:

- the Simulator's first input (WX0) to make the loop calculations for the Process Variable (PV)
- the Simulator's second analog input (WX1) to provide a manually adjustable Set Point (SP) value.
- the Simulator's first analog output (WY0) to make the value available to the rest of the program (Output)

Do-more Designer contains a built-in structure for the PID Process Simulator (called **\$IntProcSim**) that contains the following fields used to control the simulation:

.DeadTime: - The amount of time (in seconds) before an output change begins to affect the PV. This is the dead time portion of the process simulation calculation. The range of values is 0 to 3600 seconds.

.TimeConstant: - The period (in seconds) after which 63% of the Output change has been applied to the PV. The range of values is 0 to 3600 seconds.

.Noise: - the PID process simulator can inject some random noise into the system. Change the value from 0 to a number between 0.0 and 1.0 to specify the amplitude of the random noise to be injected on the PV (0.0 is less noise than 1.0).

.Enable: - starts and stops the process simulation

This dialog will write values for these fields or they can be set by the ladder logic in the controller.

| Do-more Sim provides a simple f (FOPDT) process simulation for F | irst orde PID loop | er plus dead time demonstration. |
|---|-----------------------------------|---|
| The PID loop's output is read fron the simulated PV is calculated an WX0. | n word o d writter | output WYO and n to word input |
| The process simulator's settings PLC program through the structur Changes made below are written values in that structure. | are acc re \$IntPr to their | essible from the rocSim. respective |
| What else do I nee | ed to <mark>d</mark> o | ? |
| .DeadTime: The period in seconds before an Output change begins to affect the PV. | 2 | 0 - 3600 seconds |
| TimeConstant: The period in seconds after which 63% of the Output change has been applied to the PV. | 20 | 0 - 3600 seconds |
| Noise: The amplitude of random noise injected on the PV. | 0 | 0 - 1.0 |
| Enable: Enables and disables the process simulation. | e Er | nabled 🔽 |
| Under Original | 0 | |

Update Settings - click this button to write the above three values to the Simulator.

Exit PID Simulator - click this button to close the PID Process Simulator.

The PID Process Simulator requires two ladder logic instructions in the Simulator before it can operate properly. These two instructions (a PID and a SCALE) will use the I/O in the simulator to provide space for the PID calculation and user input.

Click the **What else do I need to do?** button to display a dialog that details the instructions that must be added to the ladder program to make the PID process simulator work. An example project named PID1.Dmd is shipped with Do-more Designer that already contains the required ladder logic components that are described below.

| Step 1: | Add a PID instruction to create the loop you wish to simulate. Assign it a valid name. For this example we'll use 'MyLoop'. |
|---------|---|
| Step 2: | Check the box to enable 'Scale to .PV'. The default settings will scale the simulator's process output (which is fed into WX0) from 0-4095 to 0.0-100.0, and assign it to 'MyLoop.PV'. |
| Step 3: | Check the box to enable 'Scale from .Output %'. The default settings will scale 'MyLoop.Output' from 0.0-100.0 to 0-4095, and assign it to WY0, which is the simulator's process input. |
| Step 4: | Add a SCALE instruction to scale WX1 from 0-4095 to 0.0-100.0, and assign it to $^{\rm M}\text{yLoop.SP}^{\rm i}.$ |
| NOTES: | In the PID scaling function and the SCALE instruction, 0.0-100.0 denotes floating point values which are NOT the same as 0-100, which are integer. Specify floating point values to ensure the correct result. |
| | The PID and SCALE instructions must be enabled to function. Don't forget to turn them on with input logic. |

Add a <u>Closed Loop Controller (PID)</u> instruction with the following parameters:

- 1. give it the name MyLoop
- 2. set Position, Forward Acting, Set SP equal to PV, and do NOT select any other options
- 3. check the 'Scale to .PV' box to use WX0 as the Process Variable

make sure it's raw scaled range is 0 to 4095, and it's Eng scaled range is 0.0 to 100.0

Make sure to use floating point representations of the 0.0 and 100.0 numbers (not 0 and 100) in the SCALE portions of the instruction.

4. check the 'Scale from .Output %' box to use WY0 as .Output make sure it's scaled range is 0 to 4095 and To Scaled is set to WY0

| D | | Closed Loop Controlle |
|--|-----------------|---|
| PID Struct - Loop Algorithm (* Position (* Velocity | MyLoop | Control Loop Forward Acting C Reverse Acting |
| − Initialization Mo | de ual to PV | Other Options Use Error-Squared Enable Error Deadband Disable Bias Freeze |
| | WX0 | Scale from .Output % Scaled Min 0 0% |
| Raw Min | 0 | Scaled Max 4095 100% |
| Raw Max | 4095 | To Scaled WY0 |
| Eng. Min | 0.0 | • |
| Eng. Max | 100.0 | 0 |
| Eng Lipite | | 0 |

Next, add a Scale Value (SCALE) instruction to use WX1 as a user-controllable Set Point (SP).

- 1. set the Input to WX1
- 2. make sure the In range is 0 to 4095
- 3. make sure the Out range is 0.0 to 100.0
 - Make sure to use floating point representations of the 0.0 and 100.0 numbers (not 0 and 100) in the SCALE instruction.
- 4. make sure the output is set to MyLoop.SP

| /X % ? | | 3 |
|---------|-----------|-----------|
| SCALE | Sca | ale Value |
| Input | WX1 | 0 |
| In Min | 0 | 0 |
| In Max | 4095 | 0 |
| Out Min | 0.0 | 0 |
| Out Max | 100.0 | 0 |
| Output | MyLoop.SP | 0 |

Select Comm Port

| Available Ports - | Select the PC's serial port for Do-more Sim to use as the 'Internal Serial Port'. Please note that MP/SIM must be restarted for the new port to take effect. |
|-------------------|---|
| ОК | Cancel |

The Do-more Simulator can use one of the computer's onboard serial ports as the Internal Serial Port (@IntSerial). The Sim-> Select Comm Port menu selection will open this dialog which will list the available serial ports on the computer. Highlight the serial port in the list to use and click OK.

Note: If the port selection is changed the Simulator must be restarted for the change to take effect.