

MAD 4 Mach

MAD (Maintenance And Diagnostic) is a module that can be easily integrated into a screen set (into a “lua” panel in Mach 4’s native screen designer). The “MadLauncher.mcs” can be used to “Launch” MAD from the Wizard folder, or from being embedded in a Lua Panel. Or you can individually load functions via embedding those functions into buttons, see CUSTOMIZING end of document for details on that.

The function buttons launch the following Dialogs:

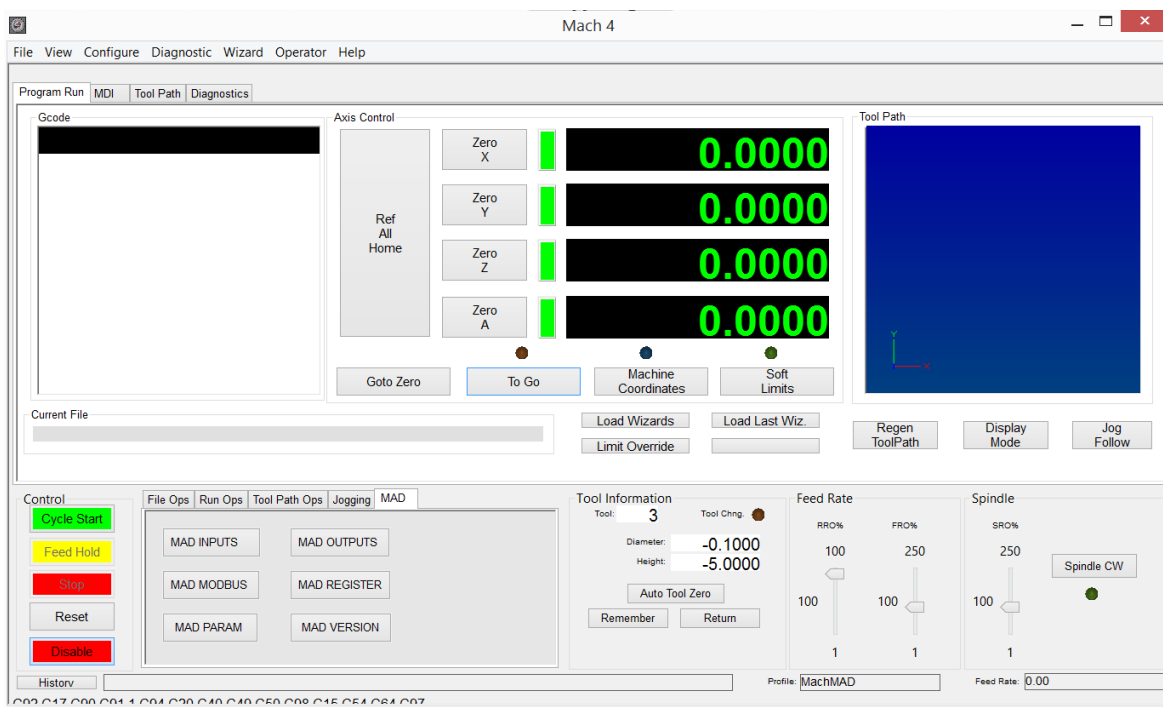
- MAD INPUTS (read only), reads the status of ALL OEM inputs real time.
- MAD OUTPUTS (read and write) OEM Outputs.
- MAD MODBUS (read and write) from/to Modbus device(s) up to 16 user defined registers.
- MAD REGISTER (read and write) from/to the Registers plugin, up to 16 user defined registers.
- MAD PARAM (Get and Set Pound Vars) up to 16 user defined pound vars.
- MAD VERSION (pops up a wx message window that shows the version).

The installer will install a MachMAD screen set (with a “MAD” tab in the Ops notebook and MachMAD profile. You can look at the MadLauncher code in the lua panel from the screen designer under the events button. You can use this screen as an example for how to integrate the launcher into your screen.

You can also launch the launcher from the “wizards” menu.

The installer installs not only the above screen set, profile and wizard, but also the compiled MAD module into the wizard folder, further it installs a MAD Resource folder that will have the DirectSoft PLC program project IF you have a DL 06 PLC with an Ethernet Module in it. If you’re running other PLCs, or want to use the Serial Port then you would need to change the parameters of the OEM Modbus plugin setup.

Screen set with Launcher tab:



Inputs Dialog:

MAD 4 MACH INPUTS

File Help

MACH OEM INPUTS

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		V		V
INPUTS 0-15:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Digitize:	0	Jog Z+:	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Index:	0	Jog Z-:	0
INPUTS 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Limit OVR:	0	Jog A+:	0
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	E-Stop:	0	Jog A-:	0
INPUTS 32-47:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	THC On:	0	Jog B+:	0
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	THC Up:	0	Jog B-:	0
INPUTS 48-63:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	THC Down:	0	Jog C+:	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Timing:	0	Jog C-:	0
Motor Homed 0-15:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Jog X+:	0	Spin@Spd:	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Jog X-:	0	Spin@0:	0
Motor Homed 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Jog Y+:	0		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Jog Y-:	0		
Motor Lim++ 0-15:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Motor Lim++ 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
Motor Lim-- 0-15:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Motor Lim-- 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Pause Updates Exit

Welcome To MAD 4 Mach INPUTS!

The “Pause Updates” button when pressed (turns RED when paused), will disable the live time updating. A green grid square with a “1” in it is “On”; white with “0” is off.

Outputs Dialog:

MAD 4 MACH OUTPUTS

File Help

MACH OUTPUTS

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		V		V		V
OUTPUT 0-15:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X++:	0	C HOME:	0	Y HOME'D:	1
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	X--:	0	GCODE RNG:	0	Z HOME'D:	1
OUTPUT 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X HOME:	0	FEED HOLD:	0	A HOME'D:	1
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	Y++:	0	BLOCK DEL:	0	B HOME'D:	0
OUTPUT 32-47:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y--:	0	SINGLE BLK:	0	C HOME'D:	0
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	Y HOME:	0	REV. RUN:	0	DWELL:	0
OUTPUT 48-63:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Z++:	0	OPT. STOP:	0	P MOUSE D:	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Z--:	0	MACH. ENA'D:	1	LIM OVR:	0
ENABLE 0-15:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Z HOME:	0	TOOL CHG:	0	CHR PUMP1:	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	A++:	0	DIS TO GO:	0	CHR PUMP2:	0
ENABLE 16-31:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A--:	0	MACH. COR:	0	AMP HI/LOW:	0
																	A HOME:	0	SOFT LIM:	0	SPIN ON:	0
																	B++:	0	JOG INC:	0	SPIN FWD:	0
																	B--:	0	JOG CONT:	0	SPIN REV:	0
																	B HOME:	0	JOG ENA:	0	COOL ON:	0
																	C++:	0	JOG MPG:	0	MIST ON:	0
																	C--:	0	X HOME'D:	1	DIG TRIG:	0

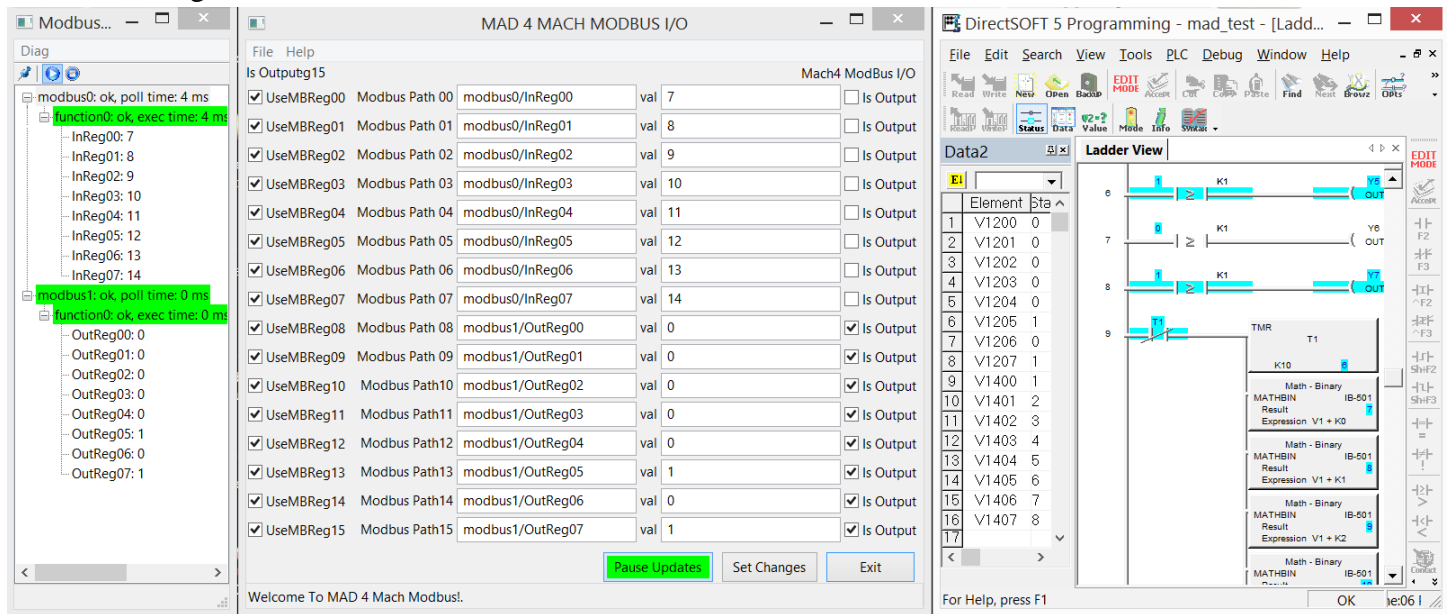
Pause Updates Set Changes Exit

Welcome To MAD 4 Mach OUTPUTS!

You will need to push the Pause updates button (turns RED when paused), if you want to change the state of an output. Once it is paused click in the grid square you want to change, change its value (0 or 1), and then push the Set Changes button. Then push the Pause update button again, this will set that output to the value you choose (1 is “On”, 0 is “Off”).

NOTE: You have to have an output enabled in mach config for you to control it, also you have to have Mach4 “Enabled” for outputs to become active. Some outputs like the “ENABLE 0-31”, “MACH ENA'D” and any other OUTPUT that its state is controlled by the core, or you have some other code running that talks to that output, then it will OVER-WRIGHT any change of state you make.

Modbus Dialog:



In the above screen shot, on the left is Mach4 OEM diagnostic utility for looking at the Modbus, the middle panels is the MAD Modbus panel, if the “useMReg##” check box is checked it will turn on or off the reading or writing to that Modbus register. This dialog comes up in an “Updates paused” state so that “Outputs” (if the “Is Output” is checked) will not write out. The right side screen shot is the DirectSoft5 PLC software, showing a simple ladder program that is reading from and writing to Mach4. In the overall screen shot above, Outputs on “UseMReg13” (and 15) have been set to “1” (it can be any numerical value that is a 16 bit integer. You can see that rungs: 6 and 7, and elements V1205, and V1207 (that turn on outputs in the PLC, are on in DirectSoft). There is timer that is running and pushing numbers into elements V1400-07 that is coming into Mach4 (you can see these in the OEM MB diagnostic, and on the MAD MB registers 0-7).

NOTE: Once you push the “Unpause Updates” it will start updating what the input registers are getting, and setting the output registers to whatever value you have in there.


NOTE2!: Once you set it to Unpause (it is running), at that point you can type into your output val text control some number and it will put that out on the fly!!!! You do NOT have to pause it to change Output values. The only time you need to pause it is for is changing your control selections.

TROUBLE SHOOTING: I noticed sometimes when you first make a change to the Outputs (only on the first change and from un-pause to pause, that the Modbus outputs throw a fault in the OEM MB diagnostics window. The fix is to go to the OEM MB diagnostic window and “click” on the “Stop Modbus” button icon(the little square/stop), then “click” on the “Run Modbus” button icon(the little right-pointing-arrow/play). Once you do that, there are no further problems. (I have asked Steve if he would add a way in mc.mcxxxxx to monitor and reset MB, and he is working on it).

Here is how I have set my Modbus Config, in Mach4 to talk to a DL06, its input addresses (outputs from Mach) are V1200-07, and its output addresses are V1400-07 (inputs to Mach). (Under the “Configure” menu, “Plugins...” pick the “Modbus” and press the configure button. For “modbus0” and “modbus1”, I have set to: IP addy 192.168.0.2, port 502, poll interval 50ms (not 25).

For the 7 input registers (holding type), I have this:

ModbusSetup



modbus0

RegIn

modbus1

RegOut

Modbus Function: Read Holding Registers 16bit (0x3)

Slave Address: 1

Modbus Register: 769

Nbr Regs: 8

	Type	Name	Description
Reg 1	Mach Input Register	InReg00	InReg00
Reg 2	Mach Input Register	InReg01	InReg01
Reg 3	Mach Input Register	InReg02	InReg02
Reg 4	Mach Input Register	InReg03	InReg03
Reg 5	Mach Input Register	InReg04	InReg04
Reg 6	Mach Input Register	InReg05	InReg05
Reg 7	Mach Input Register	InReg06	InReg06
Reg 8	Mach Input Register	InReg07	InReg07


OK

Cancel

RegIn

For the 7 output registers (holding type), I have this:

ModbusSetup



modbus0

RegIn

modbus1

RegOut

Modbus Function: Write Multiple Registers 16bit (0x10)

Slave Address: 1

Modbus Register: 641

Nbr Regs: 8

	Type	Name	Description
Reg 1	Mach Output Register	OutReg00	OutReg00
Reg 2	Mach Output Register	OutReg01	OutReg01
Reg 3	Mach Output Register	OutReg02	OutReg02
Reg 4	Mach Output Register	OutReg03	OutReg03
Reg 5	Mach Output Register	OutReg04	OutReg04
Reg 6	Mach Output Register	OutReg05	OutReg05
Reg 7	Mach Output Register	OutReg06	OutReg06
Reg 8	Mach Output Register	OutReg07	OutReg07

OK

Cancel

RegOut

Registers Dialog:

Mach4 REGISTERS					
<input checked="" type="checkbox"/> UseReg00	Reg Path 00	<input type="text" value="gRegs0/Reg00"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg01	Reg Path 01	<input type="text" value="gRegs0/Reg01"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg02	Reg Path 02	<input type="text" value="gRegs0/Reg02"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg03	Reg Path 03	<input type="text" value="gRegs0/Reg03"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg04	Reg Path 04	<input type="text" value="gRegs0/Reg04"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg05	Reg Path 05	<input type="text" value="gRegs0/Reg05"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg06	Reg Path 06	<input type="text" value="gRegs0/Reg06"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg07	Reg Path 07	<input type="text" value="gRegs0/Reg07"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg08	Reg Path 08	<input type="text" value="iRegs0/Reg00"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg09	Reg Path 09	<input type="text" value="iRegs0/Reg01"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg10	Reg Path10	<input type="text" value="iRegs0/Reg02"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg11	Reg Path11	<input type="text" value="iRegs0/Reg03"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg12	Reg Path12	<input type="text" value="iRegs0/Reg04"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg13	Reg Path13	<input type="text" value="iRegs0/Reg05"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg14	Reg Path14	<input type="text" value="iRegs0/Reg06"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg
<input checked="" type="checkbox"/> UseReg15	Reg Path15	<input type="text" value="iRegs0/Reg07"/>	val	<input type="text"/>	<input type="checkbox"/> Write Reg

Welcome To MAD 4 Mach Registers!.

Registers reads/writes to the plugin called registers, to the global and/or instance vars. Write Reg check box works the same way as Modbus above, if checked it will write whatever numerical value to that Register.

Params Dialog:

MAD 4 MACH PARAM

File Help

Write Param

☒ UseParam01

Param Num: 01

1

val

☐ Write Param

☒ UseParam02

Param Num: 02

2

val

☐ Write Param

☒ UseParam03

Param Num: 03

3

val

☐ Write Param

☒ UseParam04

Param Num: 04

4

val

☐ Write Param

☒ UseParam05

Param Num: 05

5

val

☐ Write Param

☒ UseParam06

Param Num: 06

6

val

☐ Write Param

☒ UseParam07

Param Num: 07

7

val

☐ Write Param

☒ UseParam08

Param Num: 08

8

val

☐ Write Param☒ UseParam09

Param Num: 09

9

val

☐ Write Param☒ UseParam10

Param Num:10

10

val

☐ Write Param☒ UseParam11

Param Num:11

11

val

☐ Write Param☒ UseParam12

Param Num:12

12

val

☐ Write Param☒ UseParam13

Param Num:13

13

val

☐ Write Param☒ UseParam14

Param Num:14

14

val

☐ Write Param☒ UseParam15

Param Num:15

15

val

☐ Write Param☒ UseParam16

Param Num:16

16

val

☐ Write Param

Updates Paused

Set Changes

Exit

Welcome To MAD 4 Mach Param!.

Params, will get/set Pound vars works in function like MB above .

Versions, just shows what version of MAD 4 Mach your running.

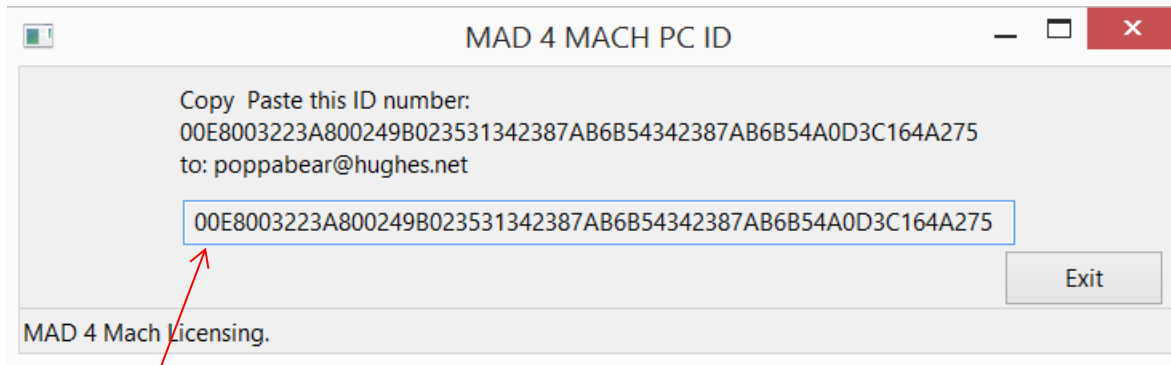
Message

i

MAD 4 MACH vs. 1.0

OK

NOTE: MAD is a licensed module, once you run it for the first time on the computer you plan on running it on. It will pop up a dialog that will have your computer number in it.



Copy and paste that out, and email it to me, and I will generate a lic., and send it back to you. The lic. lives in Mach4's main directory, i.e. C:\Mach4\MADLic.dat

The cost is \$10.00 US funds per lic. payable via paypal at, my email address, please make sure you put an email that you want the lic. coming back to and the long number above.

CUSTOMIZING:

If you would prefer to not use the launcher, and would like to embed the individual functions into a button here is a way to do it. In the screen designer, open the event script for the button you want to call a MAD function from. In that button, at the top of the script put the following:

STEP 1). --this block of code, goes at the top of the button script, it gives access to the MAD module.

```
local mInst = 0; --Mach4 instance number
local inst = mc.mcGetInstance(mInst);
local Path = mc.mcCntlGetCwd(inst);
local revpath = "";
revpath = string.gsub(Path, "\\", "/");
package.path = "./?.lua;" .. revpath .. "/Wizards/?.mcc;" --c:/Mach4Hobby
local mad = require "MAD";
```

STEP 2).

Then put **one** of these lines of code below the above block of code in the scripeter (don't include the bullet), for which ever MAD function that button is calling:

- mad.MadINPUTS(); -- for MAD Inputs
- mad.MadOUTPUTS(); -- for MAD Outputs
- mad.MadMBIO(); -- for MAD ModBus
- mad.MadREG(); -- for MAD Register (global, and instance registers)
- mad.MadPARAM(); -- for MAD Params (get and set pound vars)
- mad.Version(); -- for MAD Version (to show what version your running)

Save the script for that button, close the scripeter for that button, then save the screen set, close the screen designer. From that point your new button(s), will launch the function dialog that you wanted it to launch.

Scott "Poppa Bear" Shafer