

Digital I/O Module

Getting Started Guide

Digital IO allows you to set up automated actions to make the test process safer and more efficient. Unlike the digital IO port built into most Cirris tester models, this enhanced digital IO module comes with more voltage options, more inputs and outputs, and more current giving you additional control of your test process.

Digital IO Hardware

The Digital IO connector has a 44-pin D-Sub connector with the following functions:

20 Input/Output Dual Pins

The digital IO module has 20 input/output dual pins. The pins can be used as either an input or output depending on your needs. When used for output, the pins function similar to the dedicated output pins. However, for light loads (typically less than 4 mA) it is advised you use the dedicated outputs.

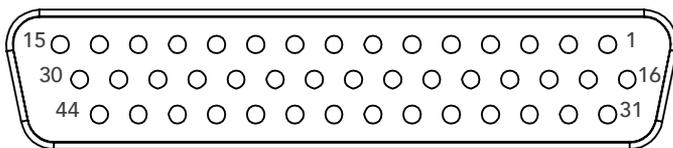
8 Dedicated Output Pins

The module's 8 dedicated output pins are open collector or "sink" outputs. When activated they will connect (or sink) a voltage to ground, in effect turning ON the output circuit. The outputs can sink up to 500 milliamps total. To limit the output current, always ensure adequate resistance between power supply and the output.

4 Voltage Rails

The digital IO module has 4 voltage rails capable of drawing a max of 500 mA combined. They are accompanied by 4 ground pins. The rails have the following voltages:

- 3.3 VDC
- 5 VDC
- 12 VDC
- 24 VDC



Connector Pin-out

Pin	Purpose	Pin	Purpose
1	24 VDC	23	Input/Output 15
2	Ground	24	Input/Output 16
3	12 VDC	25	Input/Output 17
4	Ground	26	Input/Output 18
5	5 VDC	27	Input/Output 19
6	Ground	28	Input/Output 20
7	3.3 VDC	29	Output 21
8	Ground	30	Output 22
9	Input/Output 1	31	Output 23
10	Input/Output 2	32	Output 24
11	Input/Output 3	33	Output 25
12	Input/Output 4	34	Output 26
13	Input/Output 5	35	Output 27
14	Input/Output 6	36	Output 28
15	Input/Output 7	37	
16	Input/Output 8	38	
17	Input/Output 9	39	
18	Input/Output 10	40	
19	Input/Output 11	41	
20	Input/Output 12	42	
21	Input/Output 13	43	
22	Input/Output 14	44	

Connecting the Module to the Tester

1. Connect your 1100 or Easy Touch tester to your Digital IO module.
2. Connect the power supply to an outlet.
3. Press the switch to the On position. A light will indicate the module is receiving power.

Programing Digital IO Software

The digital IO functions are written in the free LUA programming language. For more information on LUA and scripting, see the easy-wire Scripting manual.

To control the digital IO module, Cirris has created a library of LUA functions called diomod.lua. The diomod.lua file must be copied to your working directory in order for the LUA script to have access to the diomod library. This file can be found on cirris.com or on the flash drive.

diomod.ReadUserInputStates Example

The following example illustrates commands used with lua software to control the digital IO module.

```
state1, state2, state3, state4 =  
diomod.ReadUserInputStates(1, 2, 3, 4)
```

This function returns the state (on or off) of the digital inputs on the digital IO port. Each returned input state corresponds respectively to the specified input. The function returns a 1 if the input is on (high) or a 0 if the input is off (low).

diomod.SetUserOutputStates Example

The full format of this function is:

```
diomod.SetUserOutputStates(Pin1, State1, Pin2, State2, ...  
Pin8, State8)
```

The outputs are "sink" outputs. This function sets the digital outputs on (sinking) or off (floating). This function takes up to eight parameter pairs. In each pair, the first parameter, specifies which output to be set. The second parameter, specifies whether the output is set on or off. If this parameter value is 0 the output is set on (sinking low); if 1, the output is set off (not conducting).

Digital IO Full Script Example

```
cmpDIO = {}
cmpDIO.description = "Digital IO Module Example 1"

function cmpDIO.test()

    LoadLibrary("diomod") -- Loads the diomod library
    diomod.SetUserOutputStates(11,0) -- Turns on output 11
    diomod.SetUserOutputStates(12,1) -- Turns off output 12

    return MessageBox("Test Finished")

end

-- *****
-- LoadLibrary FUNCTION MUST BE PRESENT IN THE SCRIPT
-- *****

function LoadLibrary(sLibrary)

    if Cirris then

        package.path = string.format("%s;%s?.lua",
            package.path,string.match(debug.getinfo(1).
            source:match("@(.*)$"), "(.-)([^\|/]-
            %.?([^\%.\|/]*))$"))

        _G[sLibrary] = require(sLibrary)

    else

        dofile(sLibrary..".lua")

    end

end
```

If you are unfamiliar with Lua programming, Cirris can help create the scripts needed for your digital IO needs.

For more information on Digital IO, see the easy-wire Scripting manual. If you have questions, contact Cirris Tech Support.



401 North 5600 West, Salt Lake City, UT 84116

801-973-4600 | www.cirris.com