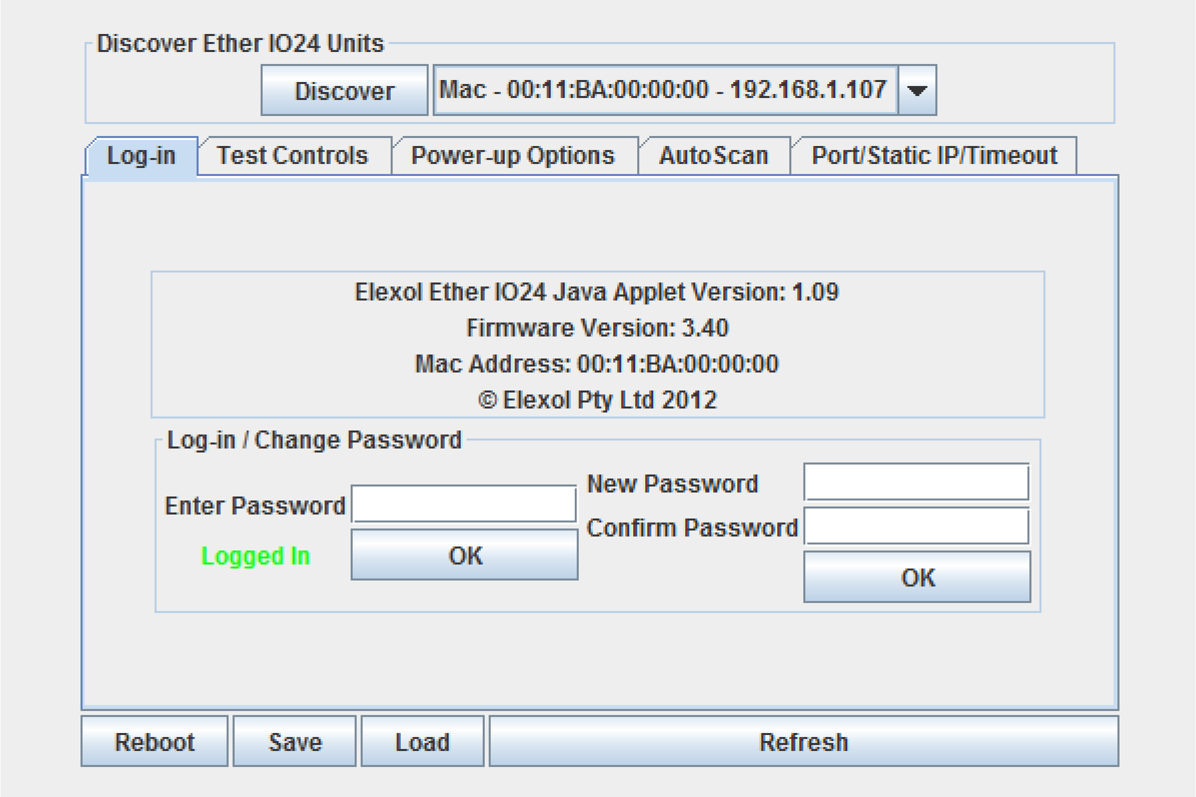
Configuring Autoscan on the ETHER IO24 TCP RANGE

The following document outlines the steps involved in setting up Autoscan on the Ether IO24 TCP as well as testing Autoscan.

Autoscan setup on Ether IO24 TCP

1. Run Test utility and search for any Ether I/O 24’s on the network.



Discover Ether IO24 Units

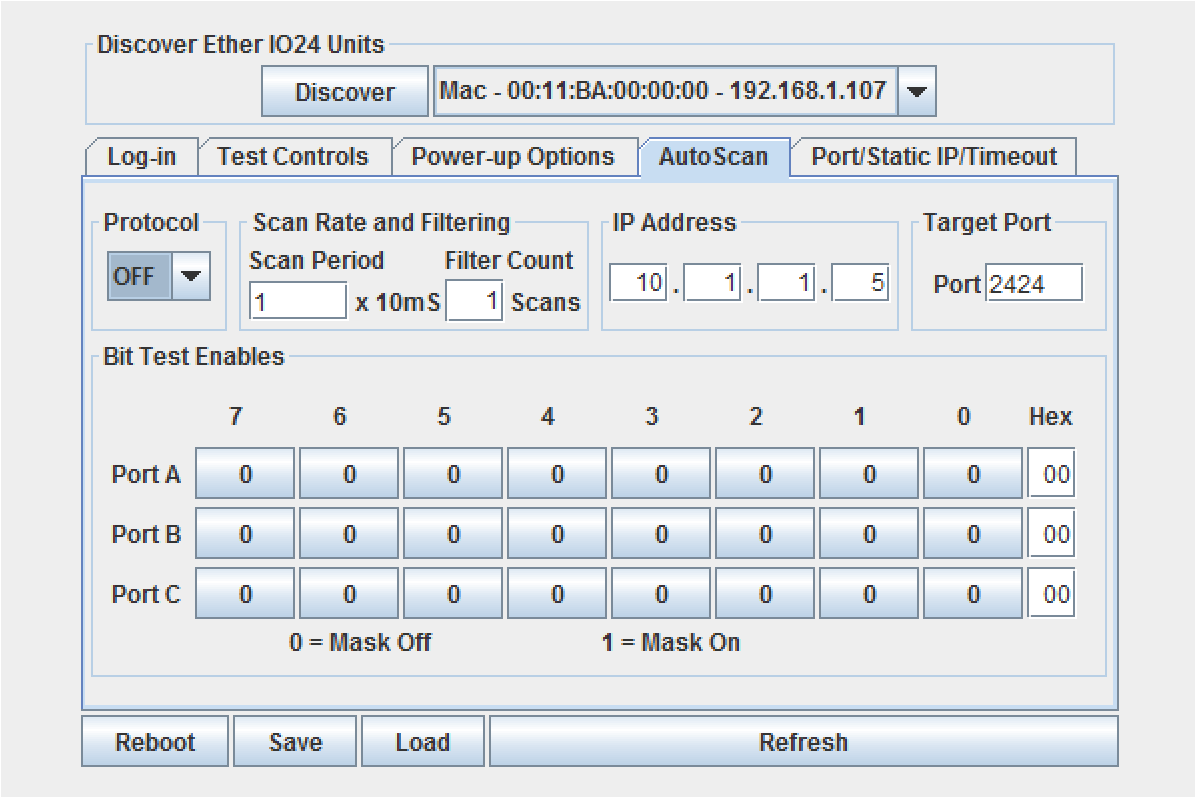
Mac - 00:11:BA:00:00:00 - 192.168.1.107

Elexol Ether IO24 Java Applet Version: 1.09
Firmware Version: 3.40
Mac Address: 00:11:BA:00:00:00
© Elexol Pty Ltd 2012

Log-in / Change Password

Logged In

2. Log into unit and Click the Auto scan Setup tab to fill in the autoscan setup values



Discover Ether IO24 Units: Mac - 00:11:BA:00:00:00 - 192.168.1.107

Protocol: OFF

Scan Rate and Filtering: Scan Period 1 x 10mS; Filter Count 1 Scans

IP Address: 10 . 1 . 1 . 5

Target Port: Port 2424

Bit Test Enables

	7	6	5	4	3	2	1	0	Hex
Port A	0	0	0	0	0	0	0	0	00
Port B	0	0	0	0	0	0	0	0	00
Port C	0	0	0	0	0	0	0	0	00

0 = Mask Off 1 = Mask On

Set up the IP address and Port number for the target address that will be used. This may be a PC or another Ether IO24 unit.

Set the scan rate and filtering that is required for the application. The above picture has a scan period of 1 x 10ms and a scan count of 1, these settings will be adequate for most applications. The unit will scan the inputs every 10 milliseconds and if there is any change, the change will be sent. These values can be adjusted to suit application needs. .

Once the address and scan rate have been set, the ports that require auto scan to be enabled can bet set. Toggle the pin state to enable auto scan on any pin and then select the protocol that enable auto scan check box.

Reboot the unit and when the enabled auto scan pin changes state, the new changed port value will be sent to the Target IP address on the port programmed in.

3. The unit can then be tested. The testing procedure is outlined in the following section.

Testing Autoscan on the Ether I/O 24 module

When implementing autoscan with a PC, the easiest way to test autoscan is working is by using a Network packet sniffer. There are a number of packet sniffers available for download, Elexol uses wireshark to test the packet sent to and from the Ether IO24 TCP.
Wireshark can be downloaded from the the following link
http://www.wireshark.org/

If you are testing autoscan with another device and it is not working a computer running the packet sniffer can be connected to a hub along with the Ether I/O device so that all packets can be captured and the problem can be resolved. If you were to place the packet sniffer onto a switch or a router then you will not see the packets being sent from the Ether I/O module to the target device.

Once you have the packet sniffer running you will see the autoscan packets being sent back to the device that you are autoscanning with. The contents of the autoscan packet will depend on which port has changed state.

The data in the packet will contain "*" followed by the port designator that changed and the new value of that port.

For example

UDP data HEX 2a 41 08 43 40

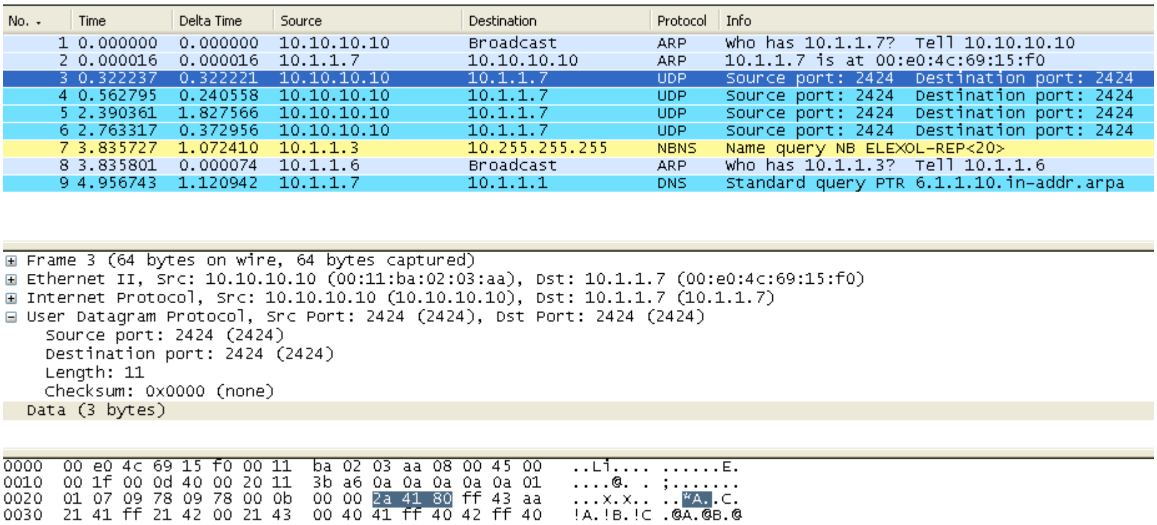
which corresponds to *A port change C port change.

The data which is sent back in autoscan mode is set out similar to the response when a read is performed, with a “*” followed by the Port designator (A,B,C) followed by the data byte (which is the value of the entire port). If bits on two or more autoscan pins change simultaneously, the module will place the changes in a single packet before being sent. If the pins are on different ports the packet data will always be in the order of A,B and C.

Below is an of the packet capture using UDP. The unit was set for the following parameters

Ether IO24 TCP IP – 10.10.10.10 (loaded J3)
Destination IP – 10.1.1.7 (PC)
Destination Port – 2424
Scan period – 1
Scan Count – 1
Port A - Enabled for autoscan
Port A power ups -Value 0xFF Direction 0xFF Pullup 0x00

UDP Autoscan Packet Capture



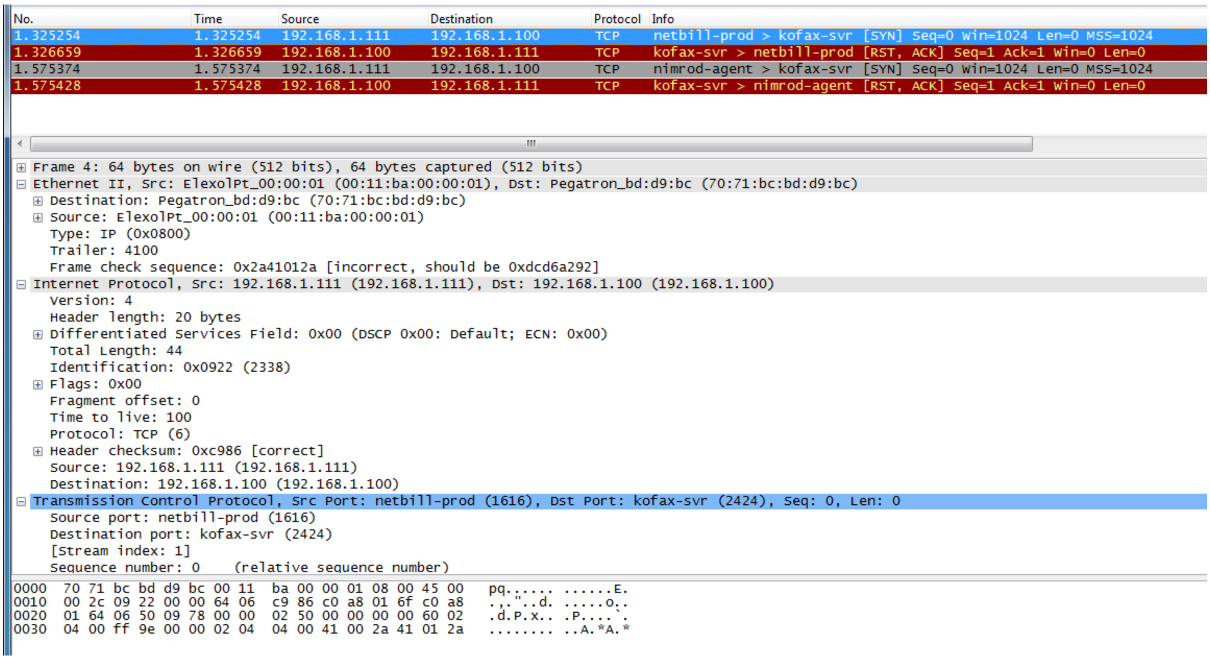
```
No. .  Time      Delta Time  Source       Destination      Protocol  Info
     1 0.000000  0.000000    10.10.10.10  Broadcast        ARP       Who has 10.1.1.7?  Tell 10.10.10.10
     2 0.000016  0.000016    10.1.1.7     10.10.10.10      ARP       10.1.1.7 is at 00:e0:4c:69:15:f0
     3 0.322237  0.322221    10.10.10.10  10.1.1.7         UDP       Source port: 2424  Destination port: 2424
     4 0.562795  0.240558    10.10.10.10  10.1.1.7         UDP       Source port: 2424  Destination port: 2424
     5 2.390361  1.827566    10.10.10.10  10.1.1.7         UDP       Source port: 2424  Destination port: 2424
     6 2.763317  0.372956    10.10.10.10  10.1.1.7         UDP       Source port: 2424  Destination port: 2424
     7 3.835727  1.072410    10.1.1.3     10.255.255.255   NBNS      Name query NB ELEXOL-REP<20>
     8 3.835801  0.000074    10.1.1.6     Broadcast        ARP       Who has 10.1.1.3?  Tell 10.1.1.6
     9 4.956743  1.120942    10.1.1.7     10.1.1.1         DNS       Standard query PTR 6.1.1.10.in-addr.arpa

⊞ Frame 3 (64 bytes on wire, 64 bytes captured)
⊞ Ethernet II, Src: 10.10.10.10 (00:11:ba:02:03:aa), Dst: 10.1.1.7 (00:e0:4c:69:15:f0)
⊞ Internet Protocol, Src: 10.10.10.10 (10.10.10.10), Dst: 10.1.1.7 (10.1.1.7)
⊟ User Datagram Protocol, Src Port: 2424 (2424), Dst Port: 2424 (2424)
    Source port: 2424 (2424)
    Destination port: 2424 (2424)
    Length: 11
    Checksum: 0x0000 (none)
  Data (3 bytes)

0000  00 e0 4c 69 15 f0 00 11  ba 02 03 aa 08 00 45 00   ..Li.... ......E.
0010  00 1f 00 0d 40 00 20 11  3b a6 0a 0a 0a 0a 0a 01   ....@. . ;.......
0020  01 07 09 78 09 78 00 0b  00 00 2a 41 80 ff 43 aa   ...x.x.. ..*A..C.
0030  21 41 ff 21 42 00 21 43  00 40 41 ff 40 42 ff 40   !A.!B.!C .@A.@B.@
```

The data being sent back is from the switch being activated on Port A bit

Below is an of the packet capture using TCP. The unit was set for the following parameters

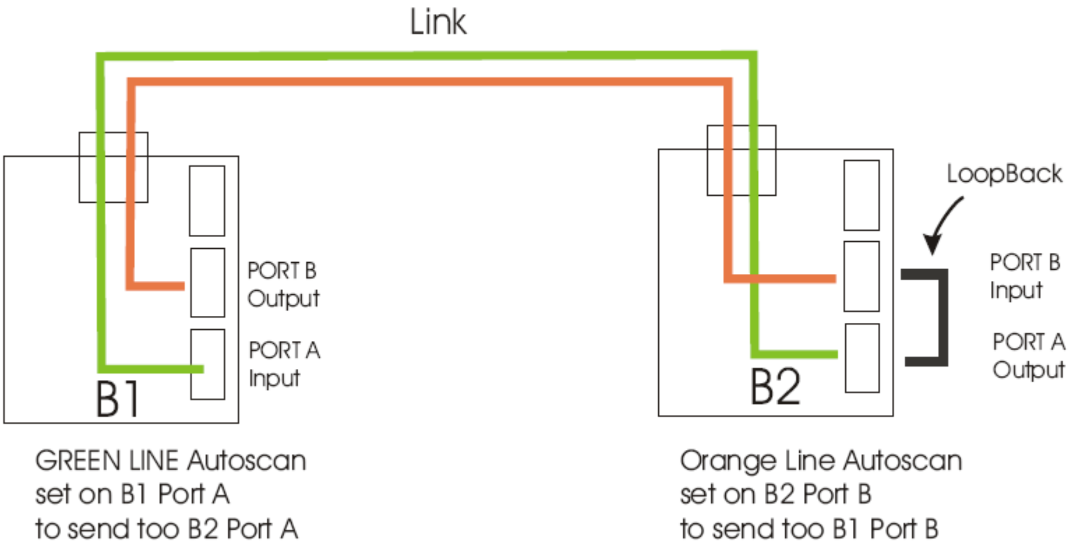
Ether IO24 TCP IP – 192.168.1.111 (DHCP)
Destination IP – 192.168.1.100 (PC)
Destination Port – 2424
Scan period – 1
Scan Count – 1
Port A - Enabled for autoscan
Port A power ups -Value 0xFF Direction 0xFF Pullup 0x00

TCP Autoscan Packet Capture



```
No.        Time      Source         Destination    Protocol Info
1.325254   1.325254  192.168.1.111  192.168.1.100  TCP      netbill-prod > kofax-svr [SYN] Seq=0 Win=1024 Len=0 MSS=1024
1.326659   1.326659  192.168.1.100  192.168.1.111  TCP      kofax-svr > netbill-prod [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1.575374   1.575374  192.168.1.111  192.168.1.100  TCP      nimrod-agent > kofax-svr [SYN] Seq=0 Win=1024 Len=0 MSS=1024
1.575428   1.575428  192.168.1.100  192.168.1.111  TCP      kofax-svr > nimrod-agent [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

⊞ Frame 4: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
⊟ Ethernet II, Src: ElexolPt_00:00:01 (00:11:ba:00:00:01), Dst: Pegatron_bd:d9:bc (70:71:bc:bd:d9:bc)
  ⊞ Destination: Pegatron_bd:d9:bc (70:71:bc:bd:d9:bc)
  ⊞ Source: ElexolPt_00:00:01 (00:11:ba:00:00:01)
    Type: IP (0x0800)
    Trailer: 4100
    Frame check sequence: 0x2a41012a [incorrect, should be 0xdcd6a292]
⊟ Internet Protocol, Src: 192.168.1.111 (192.168.1.111), Dst: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 44
    Identification: 0x0922 (2338)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 100
    Protocol: TCP (6)
  ⊞ Header checksum: 0xc986 [correct]
    Source: 192.168.1.111 (192.168.1.111)
    Destination: 192.168.1.100 (192.168.1.100)
⊟ Transmission Control Protocol, Src Port: netbill-prod (1616), Dst Port: kofax-svr (2424), Seq: 0, Len: 0
    Source port: netbill-prod (1616)
    Destination port: kofax-svr (2424)
    [Stream index: 1]
    Sequence number: 0    (relative sequence number)

0000  70 71 bc bd d9 bc 00 11  ba 00 00 01 08 00 45 00   pq...... ......E.
0010  00 2c 09 22 00 00 64 06  c9 86 c0 a8 01 6f c0 a8   .,."..d. .....o..
0020  01 64 06 50 09 78 00 00  02 50 00 00 00 00 60 02   .d.P.x.. .P....`.
0030  04 00 ff 9e 00 00 02 04  04 00 41 00 2a 41 01 2a   ........ ..A.*A.*
```

Autoscan with two Ether IO 24's without the need for a PC interface



The above picture shows an example of two Ether IO24's connected together without a need for a PC interface. It also shows how the system can be used to provide feedback between the two units.

The units are set up as followed:

PORTA on B1 is set for input with Autoscan setup to point to B2, when the pin changes state on PORTA B1, the unit will send a TCP/UDP packet to B2 notifying the change. The unit B2 PORTA is configured for output so any changes that are reflected on B1 will be seen on B2. For example if channel 2 and 3 are high on PORT A B1 and 1 and 4 are off, B2 will reflect the same state on PORTA. If any of the pins change state on PORTA B1 a new TCP/UDP packet will be sent from B1 to B2 PORT A will change on B2.

In order to provide feedback to unit B1, B2 can also be set up for autoscan. PORT A can be loop backed to PORT B. This port can then be set for input with Autoscan setup to point back to B1. PORT B on B1 can be set for output to reflect the changes. With this setup any changes that happen to PORTA B1 will be seen on PORTA B2 and PORTB B1. This is a great setup for switching remote devices and getting feedback to say that it has been activated.

Tips and tricks

- Auto scan will only work with values that are set to inputs
- Don't leave floating input pins when using auto scan as the pins will change state and constantly send data. Tie them to a known state either via external hardware or setting the pull-ups. If they are not being used it is recommended not to set the autoscan enable pin for that pin.
- The Auto scan packet sent back from the Ether IO24 contains 3 bytes '*' 'Port Designator' 'Hex Port Value' e.g. *A0 data in packet either TCP or UDP will be 2a 41 00
- Use a program called Wire shark to debug the packets being sent back and forth from the PC to the Ether I/O 24 and vice versa. www.wireshark.org
- Know what data is being sent back with the various commands sent. E.g. Sending IO24 will return 12 bytes, Read command 2 bytes, etc this will help with programming.
- Set the scan period and scan count correctly
- If Error writing value occurs, make sure that you are logged into the unit. Try rewrite the value until it is successful.
- When restarting the test utility, the port value used by the utility may change value.
- In Wireshark use MAC address when using filters e.g eth.addr eq 00:11:ba:00:00:00
- I am seeing malformed packets from the unit within the capture, the packets from the unit are not actually malformed. Wireshark is converting the packets to the Kofax format due to the port number 2424.

  What you will need to do is change a setting within wireshark so that the packet is not recognised as a Kofax packet. This can be done by going into EDIT->preferences. Then select Protocols and goto TPNCP, what you will need to so is changed the TPNCP "well-known" TrunkPack UDP port to 24242 or something different to 2424.