NETIOM-xAP

User Manual

About This Manual

This manual provides a detailed description of the operation and features of Netiom-xAP. It assumes that the reader is familiar with network terminology and therefore does not give any detailed descriptions of networks or their operation.

If you are not familiar with networking then it may help to read the “Network Terminology” document.

PC based software is provided which provides a graphical interface to the programming features of Netiom-xAP. This manual makes reference to this software in the explanations for the set up procedure.

It is not necessary to read the whole manual before setting up Netiom-xAP as it is unlikely that you will use all of its features. For small networks the Quick Start section should get you up and running so that you can view web pages on a browser.

If you encounter problems with the Quick Start then it likely that you have some sort of problem with network addressing. If so refer to that section. The Introduction and Operation section may also help with problem solving.

Introduction

Netiom-xAP is a network enabled Input / Output module with support for the Basic Status and Control (BSC) xAP Schema.

It consists of:

16 Digital Inputs
16 Digital Outputs
4 Analogue Inputs
1 Serial Port
1 Ethernet Port

Three LEDs are provided which show the status of the Ethernet connection. A link is provided to switch between active mode and programming mode.

The module can act as an HTTP server with up to 32K of web space and can generate e-mails.

Netiom-xAP will work on both local networks and wide area networks such as the Internet. A xAP software router will be required if xAP is to be used over a wide area network. This is available as a free download from www.mi4.biz.

To run Netiom-xAP you will need:

A 12 Volt d.c. power supply.
A null modem serial lead (for programming).
An Ethernet cable to connect to the network.

xAP Background

xAP is a new protocol ideally suited for controlling many different devices typically found in an automation environment, particularly in the home. It encompasses very simple On/Off devices to very complex audio/video servers, databases and internet data sources. It is the 'glue' that allows lots of different devices to work together. xAP is a broadcast protocol which ensures that all devices hear information from all other devices, on Ethernet this is implemented by using UDP. xAP messages are sent in readable text so are easy to interpret.

The Netiom-xAP fully supports the xAP v1.2 protocol and in particular the BSC v1.3 schema (Basic Status and Control). This particular xAP standard is aimed at allowing simple devices to monitor and control each other in an almost 'plug and play' fashion. Additionally the Netiom-xAP implements an extra schema specific to itself called 'Netiom'.

BSC is a very standard way for simple devices (ON|OFF, LEVEL or TEXT based) to interact. There are only these three types of nodes that BSC can support and conveniently all the input/output capabilities of the Netiom xAP can be represented and this allows for the Netiom xAP to have a standard way to work harmoniously with other xAP BSC devices. Several commercial software applications have inbuilt support for BSC using xAP plugins, for example HomeSeer and XLobby (or MainLobby via Homeseer). HomeSeer which is a sophisticated automation engine can even automatically recognise the Netiom-xAP and is able to monitor and control it in a 'plug and play' fashion using the xAP plugin. Netiom-xAP is easily integrated into these applications using the BSC schema. Other 'free' applications like xAP Desktop can be used too.

Each 'node' on the Netiom xAP can be addressed using BSC and there are 102 such nodes in 8 categories (eg 16 x inputs, 16 x outputs, 4 x analogue inputs etc). Each of these nodes has an associated number or **ID** but can also be named using the configuration application so output 4 could be named "SecurityLights" for example which makes things much easier. The ID is fixed and is a two digit hex number as listed in appendix 3, in this example it is 14.

BSC defines four message types to implement full status reporting and control. Two messages '**info**' and '**event'** are used to periodically provide information on the state of an input or output node. An 'event' is sent instantly something changes to inform other interested listeners , and an 'info' is sent periodically just as summary information. The time between the 'info' messages can be configured , or they can be turned off completely. There are also two action messages **'query'** and **'cmd'** . 'Query' is used to request the current state of an input or output and 'cmd' is a command to instruct an output to change state. Obviously 'cmd' can't be issued to an input. Here are two example xAP messages for our 'security light', the first xAP message is an 'info' telling us the light is currently ON and the second is a 'cmd' from our ACME automation controller that turns the SecurityLight off by 'targeting' it. You can skip over the other parts of the message for the time being. The key values in the message are in bold.

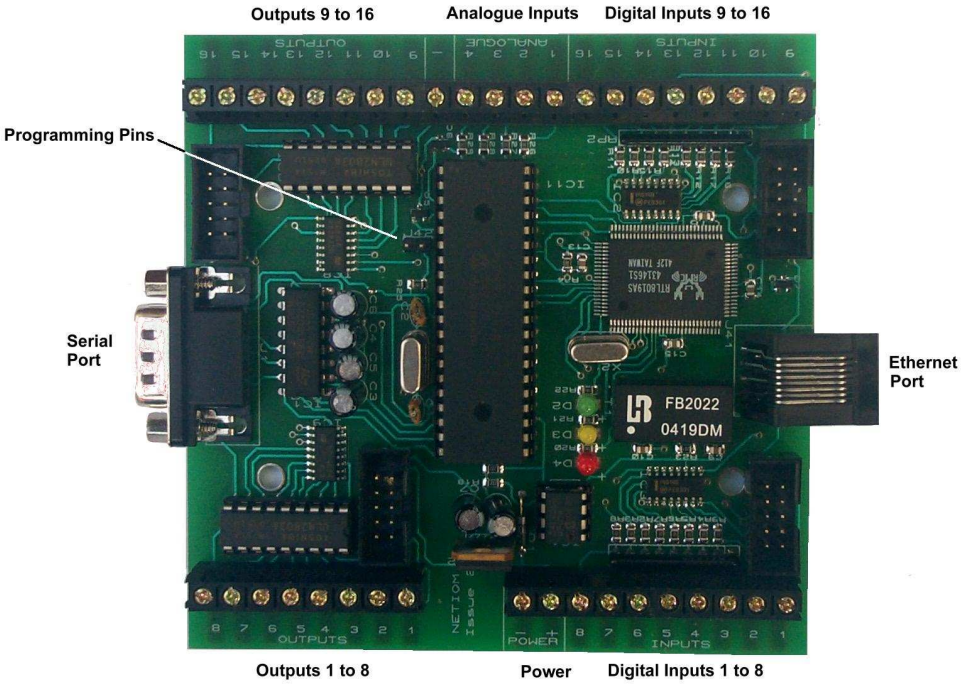
```
xap-header
{
    v=12
    hop=1
    uid=FF2234 14
    class=xapbsc.info
    source=phaedrus.netiom.MyPC:SecurityLight
}
output.state
{
    state=on
}
```

```
xap-header
{
```

```
    v=12
    hop=1
    uid=FF223300
    class=xapbsc.cmd
   source=ACME.Controller.MyOtherPC
    target=phaedrus.netiom.MyPC:SecurityLight
}
output.state
{
    ID=14
    state=off
}
```

There are many advanced features of xAP , inparticular wildcarding which allows you to address many devices at once. The Netiom-xAP supports all these advanced features.

Further information is available from the main xAP website at http://www.xapautomation.org and there are several introductory articles there. Also there is the full xAP v1.2 specification and most importantly the BSC v1.3 schema specification at http://downloads.xapautomation.org/documents/BSCv13.pdf . There are also links to many other interesting xAP websites that offer freely downloadable applications that can be used with Netiom xAP. xAP Desktop is downloadable from http://www.mi4.biz, along with many other interesting applications including a xAP router to pass xAP information across the internet.



Operation

Netiom-xAP has 3 LEDs which indicate the status of the Ethernet connection.

Green indicates that Netiom-xAP is connected to the network. When it is on it indicates that the network connection is present but not necessarily that Netiom-xAP is communicating with the network.

Amber is the receive indication. This is normally on and switches off when data is received. In normal operation this will flicker off every few seconds as it detects normal network polling.

Red is the transmit indication. This will switch off when data is transmitted from Netiom-xAP. It is normally on.

The programming link is used to switch Netiom-xAP between active mode and programming mode.

In active mode Netiom-xAP will connect to the network and respond to Ethernet messages. It will not respond to any commands issued on the serial port.

In programming mode Netiom-xAP will not access the network but it will accept commands from the serial port. When the link is removed all network connections will be reset.

Quick Start

1. Place the programming link in position.

2. Connect the network to Netiom-xAP with an Ethernet cable.

3. Connect a 12 Volt d.c. power supply to the power terminals.

4. The LED's should now be illuminated. The amber LED may flicker occasionally.

5. You may now want to verify the settings using the Netiom-xAP Interface software.

6. Install and run the Netiom-xAP Interface Software.

7. Select the PC comms port you will be using. This is on the General page.

8. Connect a null modem lead between the PC comms port and the Netiom-xAP serial port.

9. Press the "Read From NETIOM" button. You can now review the settings.

10. The next step will depend on your network settings. If you have a DHCP server (typically an ADSL router) proceed from section "With DHCP Server" otherwise go the section "Without DHCP Server".

With DHCP Server

11. Ensure that the "Get Address Automatically" box is checked on the server page of the Netiom-xAP Interface software. If it is not checked, then check it and download the new settings to Netiom-xAP using the "Write To NETIOM" button on the general page.

12. Remove the programming link. The red LED should now flicker once or twice.

13. Wait for about 5 seconds and replace the programming link.

14. Read the data back from Netiom-xAP as before.

15. The assigned IP address will now be displayed on the Server page. If this reads 0.0.0.0 then the DHCP process has failed and it will be necessary to set up the IP address manually. Follow the "Without DHCP Server" section instructions.

16. Make a note of the IP address and go to step 22.

Without DHCP Server

17. You will need to set up the addressing manually. Uncheck the “Get Address Automatically” box on the server page of the Netiom-xAP Interface software.

18. If there are a large number of devices on your network you will need to be assigned an IP address from your network administrator. If there are only a few PCs you can determine their address from a command prompt. Type in “ipconfig” and the settings will be returned. Make a note of each one and choose an address close to the ones returned changing only the fourth number. For example if your network is using 192.168.0.2 and 192.168.0.5 try using 192.168.0.6.

19. Enter the obtained number into the IP address boxes. You should be safe using 192.168.0.1 for the gateway address and 255.255.255.0 for the Submask.

20. Download the new data to Netiom-xAP using the “Write To NETIOM” button on the general page.

21 When the download is complete remove the programming link.

22. You should now be able to connect to Netiom-xAP on a web browser. Enter the IP address in the address field of the browser and the web pages should appear.

Interface Software

PC based software is provided to provide a graphical interface to programming Netiom-xAP.

On the opening page there are controls for loading and sending data and a logging area which will show a log of activity as well as any problems encountered.

Sections are provided which cover:

Server and General parameters.
e-mail options.
Web page selection and downloading.
xAP related parameters.

Data

Data can be written to Netiom-xAP or read from it via the serial port at 19200 baud. Note that the programming link must be in place before communications can be established with Netiom-xAP. The best way to use the program is to read data from Netiom-xAP, modify it and then write it back.

In general only data that is needed will be sent to Netiom-xAP. For example if e-mail is disabled then none of the E-mail data will be downloaded. This means that you can temporarily disable a function without affecting data held within Netiom-xAP.

At start up the program will load a set of defaults from the file “DefaultConfig.txt”. If you need a different set of defaults you can overwrite this file with your own values. If you have several different configurations then these can be saved as separate files.

Before sending data to Netiom-xAP the data is checked for any inconsistencies and will not proceed if any are found. For example, if e-mail has been enabled but no IP address has been specified then data will not be sent to Netiom-xAP as this might cause unpredictable operation. If problems are encountered a list is written to the logging window.

During the write operation to Netiom-xAP all data is read back and compared with the data sent. If there are any problems then they will be reported in the log window.

Web Pages

All web page selection and downloading is done from the “Web Page” section.

You can specify a list of pages which will be used to create an image to download to Netiom-xAP. These do not need to be in a single folder but it may be more convenient if they are. When you download the web pages they are copied to a sub folder called “TempWebPges”. They are then combined into an image file and sent to Netiom-xAP. The copied files are then deleted.

Do not place any other files in the “TempWebPges” as they will be included in the image and then deleted.

It is not possible to read the web pages back from Netiom-xAP through the serial port and so no checking is performed. You will need to check that they have been successfully downloaded using the network.

Setting The Address

In order to function properly on a network Netiom-xAP needs know the following information:

IP Address
Gateway Address
Subnet Mask

Netiom-xAP is able to determine this information automatically if the network has a DHCP server. However if these addresses are obtained automatically they will be dynamic and can change with time. If Netiom-xAP is used as a server this can cause problems as clients will not know the current address. In these situations assigning a static IP address may be preferable.

To obtain an address automatically check the “Get Address Automatically” box. After you download the data to Netiom-xAP and remove the link Netiom-xAP will attempt to obtain the relevant information.

To set up the addresses manually uncheck the” Get Address Automatically” box and enter the addresses in the appropriate edit boxes. Each box must contain a number in the range 0 to 255. The default values are:

IP Address: 192.168.0.6
Gateway Address: 192.168.0.1
Submask: 255.255.255.0

These values should be OK for small networks with only one or two computers and a gateway router.

If you need to change them you will need to ascertain the values from the network. For local networks there are three address ranges:

10.0.0.0 to 10.255.255.255
172.16.0.0 to 172.31.255.255
192.168.0.0 to 192.168.255.255

The best way to find free addresses is to log on to the gateway router as this will keep track of all the attached devices. It will also show its own IP address which should be used as the Gateway Address.

You can find the settings on any local PC running Windows by going to the command prompt and typing “ipconfig”. This will return the IP Address of the PC, the Gateway Address and the

Submask. If you only have one PC and gateway router then use the last two values for the Gateway Address and Submask and increment the last digit of the IP Address by 1.

Device Name

This is an optional field and need not be filled in. It is used as a tag field in dynamic web pages and is useful as a way of identifying individual Netioms if you plan to use more than one. The maximum size is 8 characters.

Server Passwords

With the “Enable Server Password” box unchecked web pages will be available to anyone who accesses the Netiom-xAP from a browser.

With the box checked a user name and password will be required when the user first accesses the Netiom-xAP. Once the user name and password has been entered it will be valid for the remainder of the session.

Both user name and password fields will need to be filled in if this option is selected. Maximum size for both fields is 8 characters. Note that these fields are case sensitive.

Serial Port

With the programming link in place the serial port always runs at 19200 baud.

In active mode you can specify the baud rate so that Netiom-xAP can communicate with external equipment. Netiom-xAP can support baud rates in the range 2400 to 38400 baud.

Netiom-xAP will store incoming serial data in a working buffer until it deems that the data string is complete. Once complete the data is made available for dynamic web pages or is used to trigger an e-mail.

You can specify either a delimiter indicates that the data is complete or use a timed methology i.e. no data for a fixed period. If the latter is chosen then the timeout value can be specified in the range 20mS to 20 seconds.

If a delimiter is used it can be either a carriage return/line feed pair (CR/LF) or a single character. To use carriage return/line feed pair as the delimiter check the “Use CR/LF” checkbox. To use a single character check the “Use Delimiter” checkbox and specify the character in hex format (for a list of ASCII codes see appendix 4).

HTTP Settings

HTTP Port

Normally web browsers use port 80 to initiate a connection to a web server. However you can program Netiom-xAP to listen on any port in the range 1 to 65535. This would normally only be necessary if Netiom-xAP is connected to an external network such as the Internet. If you have another HTTP server or you are running more that one Netiom-xAP then separate ports may be necessary for port forwarding. It may also be useful as an additional security method to help prevent unauthorised users stumbling across your site.

If you do change your HTTP port number Netiom-xAP will still be accessible from browsers by specifying the port number with the address:

http://192.168.0.6:8080/

will connect to IP Address 192.168.0.6 using port 8080.

MAC Address

This field shows the MAC address of the Netiom-xAP. It is not possible to change this number and is provided for information only.

XAP Messaging

Netiom-xAP supports the Basic Status and Control (BSC) set of xAP Schema. It will respond to query and cmd messages with event and info messages. Netiom-xAP will not issue any cmd or query messages. The xAP specification and the BSC schema details may be downloaded from www.xapautomation.org.

Address

The target address of a xAP enabled device will take the following form:

AVendor.ADevice.ALocation:ANode

The first two elements (Vendor and Device) are fixed as “Phaedrus.Netiom” and are not programmable.

The third element (ALocation) can be programmed with up to 32 characters. Sub-addressing and wildcarding is fully supported. This would typically be used to identify individual Netiom-xAPs.

The fourth element, the node address, can be programmed with up to 32 characters. Each input and output (as well as counters, latches etc) on the Netiom-xAP is a distinct node. Sub-addressing and wildcarding is fully supported. When using ‘*’ wildcarding it is important each node address has the same number of sub-addresses. For example if two node addresses have the designators:

Output.1 (2 sub-addresses)
Output2 (1 sub address)

Then using *.* in the target field will only address Output.1, while using a single ‘*’ will only address Output2.

Individual nodes can be excluded from the xAP processing by leaving the node designator blank. It is recommended that any unused node be excluded, as this will reduce the time associated with processing xAP messages. Any nodes thus excluded can still be used with other processes such as e-mail generation and web pages.

The address is used in conjunction with the ID element and both must match when issuing a command before that command is implemented. While it is possible to use a wildcard (*) for the ID it is better to use the node number directly. This can significantly reduce the response time as Netiom-xAP will not need to search through all the addresses for a match.

Heartbeat

Netiom-xAP will generate a heartbeat message at a fixed period of 60 seconds.

Updates

Netiom-xAP can send regular info messages even when conditions have not changed. The period between messages can be set on the Control screen of the interface application by adjusting the value in the “xAP Update Period” box. The value can be set between 1 and 255 seconds. A value of zero will disable this feature.

A single message is sent at the end of each period and Netiom-xAP will cycle through all enabled nodes. Normally each message will be of info type. However if the value of the node has changed since the last message for a particular node it will be of type event. Event messages generated by this mechanism will only be generated by counters and analogue inputs if the change is outside their hysteresis level.

Nodes

Netiom-xAP has a total of 102 nodes split into 8 groups. Each node can be named with up to 32 characters as required.

16 Digital Inputs
16 Digital Outputs
4 Analogue Inputs
2 Serial Ports (input and output)
16 Virtuals
16 Counters
16 Low latches
16 High Latches

Each node can be individually addressed and fully supports wildcarding and sub addressing. Although there is only one serial port it has been divided into separate nodes for input and output purposes.

XAP BSC Class Messaging

Digital Inputs

A typical xAP message from an input would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF12340F
   class=xAPBSC.event
   source=Phaedrus.Netiom.location:input.1
}
input.state
{
   state=on
}
```

A single element is reported:

“state” is the input status and will be either “on” (input is high or contacts open) or “off” (input is low or contacts closed).

Digital inputs will not respond to commands.

Digital Outputs

A typical xAP message from an output would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123411
   class=xAPBSC.info
   source=Phaedrus.Netiom.location:output.1
}
output.state
{
   state=off
}
```

Only a single element, “state”, is reported. This may be either “on” or “off”.

Digital outputs will respond to on, off or toggle commands. Only “state” is supported. A digital output is considered to be on when it is a low condition. This is the equivalent to a relay or LED being on if it is connected from an output to a positive supply.

Analogue Inputs

A typical xAP message from an analogue input would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123461
   class=xAPBSC.info
   source=Phaedrus.Netiom.location:Analogue.1
}
output.state
{
   state=on
   level=358/1023
}
```

Two elements, “state” and “level”, are reported. Level will be in the range 0 to 1023. State will be defined by the “Analogue On State Level” which can be defined in the interface software.

Commands are not supported.

Analogue inputs have a Hysteresis value which is used to determine the minimum change required before a xAP event message is generated. This can be set anywhere within the range 1 to 1023. If xAP events are not required for a particular analogue input the Hysteresis level should be set to 1024. This will still allow interrogation of the value with a xAP query.

Serial Port

A typical xAP message from the serial port would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF1234654
   class=xAPBSC.event
   source=Phaedrus.Netiom.location:SerialPort.In
}
input.state
{
   state=on
   text=Hello World
}
```

Only a single element, “text”, is reported. If this message is in response to a query the last recorded serial input will reported.

The “text” element is supported as part of a command to the serial output port. In this case the associated text will be sent out of the serial port.

Some equipment requires a carriage return / line feed pair to delimit the end of a text string. This cannot be accommodated within the xAP message structure using the “text=” format. However if this is required check the “Append CR/LF” check box in the User Interface and each message will be terminated with the carriage return / line feed.

Other single character delimiters can be specified by checking the “Append Delimiter” check box and specifying the character in hex format (for a list of ASCII codes see appendix 4).

Serial port data can also be sent or reported in hex using the pling format (“text!”). For example the equivalent of the above event in hex would be:

```
xap-header
{
   v=12
   hop=1
   uid=FF1234654
   class=xAPBSC.event
   source=Phaedrus.Netiom.location:SerialPort.In
}
input.state
{
   state=on
   text!48656C6C6F20576F726C64
}
```

For outgoing messages Netiom-xAP will automatically detect the two command types and act accordingly.

For incoming messages three options are provided:

“Use ASCII Only” will only report in ASCII characters (i.e. text=). Any Line Feed characters will be removed as these will corrupt the xAP message. All other characters will be reported even if they are non-printable characters.

“Use Hex Only” will always report in hex format (i.e. text!).

“Auto ASCII / Hex” will examine the incoming data and if there are any non-printable characters it will report in hex. Otherwise it will report in ASCII.

Virtuals

Virtuals are the equivalent of an input / output but with no physical connection. Their only function is to generate xAP message when they change. Thus they can be used as part of a web page to trigger xAP events.

A typical xAP message from a virtual would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123427
   class=xAPBSC.info
   source=Phaedrus.Netiom.location:Virtual.2
}
output.state
{
   state=off
}
```

Only a single element, “state”, is reported. This may be either “on” or “off”.

Virtuals will respond to on, off or toggle commands. Again only “state” is supported.

Counters

The 16 counters are incremented each time the corresponding input goes from a low to a high condition.

A typical xAP message from a counter would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123431
   class=xAPBSC.info
   source=Phaedrus.Netiom.location:Counter.1
}
output.state
{
   state=on
   level=123/65535
}
```

Two elements, “state” and “level”, are reported. Level will be in the range 0 to 65535. State will be defined by the “Counter On State Level” which can be defined in the interface software.

Commands can be used to set the counter value anywhere in the valid range.

Counters have a Hysteresis value which is used to determine the minimum change required before a xAP event message is generated. This can be set anywhere within the range 1 to 65535. If xAP events are not required for a particular counter the Hysteresis level should be set to 65535. This will still allow interrogation of the value with a xAP query.

Latches

Two sets of latches are provided. High latches will be set when the corresponding input changes to a high condition and Low latches will be set when the input goes to a low condition.

A typical xAP message from a latch would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123451
   class=xAPBSC.info
   source=Phaedrus.Netiom.location:Latch.1
}
output.state
{
   state=off
}
```

Only a single element, “state”, is reported. This may be either “on” or “off”.

Latches will respond to on, off or toggle commands. Only “state” is supported. While latches can have their state set to either “on” or “off” it would be normal practice to only set them to “off”. If an input is high and the corresponding latch is set to “off” a xAP message will be generated showing the state as “off”. However the latch will immediately be set thus generating another message.

xAP Netiom Class Messaging

Netiom class messages address blocks of nodes. The target field should only address the base address i.e. Target=Phaedrus.Netiom.location.

Two families of messages are provided: cmd and query. The query message can only request data from whole blocks of nodes (i.e. all inputs, all outputs etc.). The command message can address multiple nodes or whole blocks of nodes.

Netiom Class Commands

A typical Netiom class command header would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123451
   class=netiom.cmd
   source=Acme.Device.Name
   target=Phaedrus.Netiom.location
}
```

Multiple message blocks are supported within the schema. Each message block header defines the block of nodes being addressed.

Digital Outputs	outputs.state
Virtuals	virtuals.state
Counters	counters.state
Low Latches	lowlatches.state
High Latches	highlatches.state

The body of the message will contain a list of nodes which are being addressed. Thus to set a single node Output.1 on the message body would look like:

```
outputs.state
{
Output.1=On
}
```

Any number of nodes may be included in a single body for example:

```
outputs.state
{
   Output.1=On
   Output.2=On
   Output.3=Off
   Output.4=Off
}
```

As stated above multiple message blocks are supported so it is possible to address several blocks in a single message:

```
outputs.state
{
   Output.1=On
   Output.2=On
   Output.3=Off
   Output.4=Off
}
counters.state
{
   Counter.1=123
   Counter.2=456
   Counter.3=789
   Counter.4=0
}
```

All nodes in a particular block can be set to a defined state with the “<group name>=” identifier. For example:

```
outputs.state
{
   outputs=On
}
```

will turn all outputs on.

As well as addressing different node blocks it is possible to address the same node block within the same message:

```
global.state
{
   outputs =On
}
outputs.state
{
   Output.4=Off
}
```

In this example all outputs are turned on except output 4 which will be turned off. This provides a simple method for setting all outputs glitch free to a particular pattern as the outputs are not updated until the whole message has been processed.

All nodes can be addressed in a single block using the “global.state” message block:

```
global.state
{
  outputs=on
  virtuals=off
  counters=0
  lowlatches=on
  highlatches=off
}
```

It is not necessary to address all node blocks within the global message. For example:

```
global.state
{
   outputs=on
}
```

is equally valid.

Any netiom command will generate a netiom info response for the blocks addressed (see Netiom Class Queries). In addition any states which are changed will generate the appropriate BSC event messages.

Netiom Class Queries

Queries can address individual blocks of nodes. A typical header would look like:

```
xap-header
{
   v=12
   hop=1
   uid=FF123451
   class=netiom.query
   source=Acme.Device.Name
   target=Phaedrus.Netiom.location
}
```

The message block should contain a list of node blocks to be addressed. For example:

```
state
{
   node=inputs
}
```

It is not necessary to include all blocks within the message body any combination is possible.

Netiom-xAP will respond to a query with a message with a single message block as follows:

```
xap-header
{
   v=12
   hop=1
   uid=FF123400
   class=NETIOM.info
   source=Phaedrus.Netiom.location
}
inputs.state
{
   input.16=on
   input.15=on
   input.13=on
   input.8=on
   input.9=on
   input.14=on
   input.5=on
   input.6=on
   input.7=on
   input.2=on
   input.3=on
   input.1=on
   input.11=on
   input.10=on
   input.4=on
   input.12=on
}
```

Where several blocks are requested Netiom-xAP will respond with several messages each with a single message block.

E-mail

General

Netiom-xAP can generate e-mails in response to events on its inputs. It uses the SMTP protocol with “plain auth login”. This means that Netiom-xAP will supply a user name and password to the e-mail server.

In order to enable Netiom-xAP to send e-mails you will need to:

Check the “Enable e-mail” box.
Specify the IP address of the e-mail server (not the name).
Specify Account Name and Password
Specify the e-mail Address of the sender
Specify the e-mail Address of the receiver

Account

Most of this information is the same as you would use to set up an account on a PC. The main difference is that the e-mail server will use a URL. Netiom-xAP does not have the ability to resolve URLs so it will be necessary to look it up. There are several sites on the Internet, which provide a look up service. If you have not done this before try:

http://www.kloth.net/services/dig.php.

The e-mail Address of the sender should be one that you use with this account. The e-mail Address of the receiver should be (obviously) the address to which the e-mail should be sent.

The maximum size of the fields are:

Account Name 24 characters
Password 8 characters
e-mail To 32 characters
e-mail From 32 characters

If you are having difficulty connecting to the e-mail server you can record the server responses via the serial port in a program such as HyperTerminal. Check the “View Server Messages” box and Netiom-xAP will repeat all incoming messages from the e-mail server out to the serial port. This may give an indication as to what is going wrong. Note that Netiom-xAP will use the baud rate setting (Serial Port section) and that the programming link will need to be removed before e-mails are generated.

Note that e-mails will not be generated while the programming link is in place. The baud rate of the serial port will be that selected under the Serial Port section.

Triggers

E-mails can be triggered from any of the digital or analogue inputs or from serial port messages.

Digital inputs can trigger e-mails when they go high or when they go low or both. These are selected by checking the associated boxes in the “Trigger Low on Inputs” and “Trigger High on Inputs” sections. There is no limit on the combinations of high or low triggers.

Analogue inputs can trigger e-mails when their values exceeds a preset level and / or falls below a preset level. Each input can have different values for above and below levels. So for example if the above level was set to 600 and the below level was set to 500 an e-mail would be generated when the input level reached 600 or when it fell below 500.

In order to prevent multiple e-mails being generated from analogue inputs a hysteresis value can be specified. For “above” values, after an e-mail is generated, the input level would need to fall below the “above” level minus the hysteresis value and then back to the “above” level again before a second e-mail is sent.

Thus taking the value of 600 (above) with a hysteresis value of 10 an e-mail would be generated when the input value reached 600. If the input value fell to say 595 and then returned to 600 a second e-mail would not be generated. The level would have to reach 590 before returning to 600 before another e-mail is generated.

Similarly with “below” levels the hysteresis value would be added to the “below” value.

If you do not want to generate e-mails from these inputs then the “above” level should be set to 1024 and the “below” level should be set to 0.

e-mails can also be triggered from the serial port. Each time a serial port message is received e-mail is generated.

E-mail Content

The subject and message body of the e-mail can be customised using dynamic web page files. These have fixed names: for the subject “esub.cgi” and for the body “email.cgi”. These files can use any of the scripts used in the standard web page files.

If file “esub.cgi” is not present Netiom-xAP will use “NETIOM Message” as the subject. If “email.cgi” is not present Netiom-xAP will use “Test Message” as the body of the e-mail.

Netiom-xAP will process the dynamic web page files after connection to the e-mail server has been established. Normally this will happen within 1 or 2 seconds but there are times when this period may be extended to several seconds if there are connection problems. Therefore if the triggering mechanism is transitory it may have reverted to its normal condition before the e-mail is actually sent.

In the event of a problem in sending an e-mail, Netiom-xAP will wait approximately 10 seconds and re-try. If there are further problems Netiom-xAP will continue re-trying for a total of 10 attempts. If there has been no success at this point no further attempts will be made until another e-mail is generated.

Pings

Netiom-xAP will respond to network pings. To ping Netiom-xAP from a PC use the command line ping followed by its IP address. For example:

```
ping 192.168.0.6
```

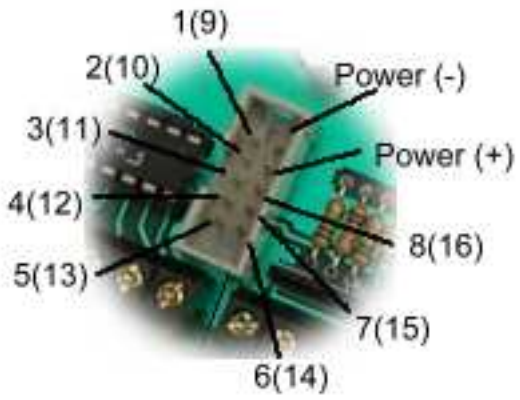
Hardware

Connections

The digital inputs and outputs are available via screw terminals or a 10 way headers. The headers also have power pins so that auxiliary modules can be powered directly from Netiom-xAP. Each of the headers has the same pin assignment regardless of whether they are inputs or outputs.

Note that the input connectors Power + pin is connected to 5V while the output connectors Power + pin is connected to 12V.

The following diagram shows the pin arrangement:



Digital Inputs

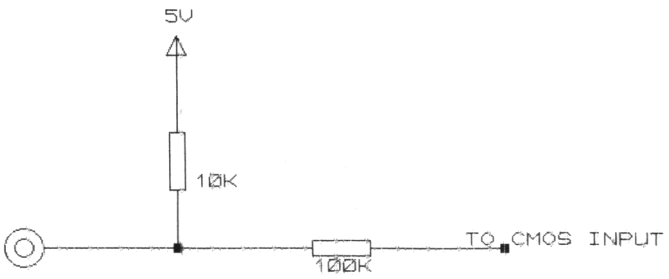
Each digital input is connected to a CMOS gate via a 100K resistor. It is pulled up to the internal 5 Volt supply rail via a 10K resistor.

The input can be directly interfaced to TTL and CMOS outputs. The input voltage should not exceed 5 Volts. In addition the inputs can use clean contacts referenced to 0 Volts or open collector outputs again referenced to 0 Volts.

If the driving modules do not share the same power supply as Netiom-xAP then it will be necessary to common their 0 Volt lines.

The inputs are scanned with a timebase of 10mS. Note that during periods when Netiom-xAP is servicing network activity input scanning may be delayed by approximately 50mS. Therefore it is recommended that inputs be present for at least 100mS in order to guarantee recognition.

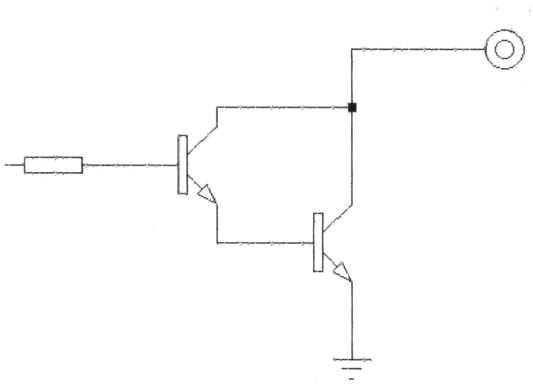
With the input high or open circuit Netiom-xAP will report it as a “1” or “on”, otherwise it will be reported as a “0” or “off”.



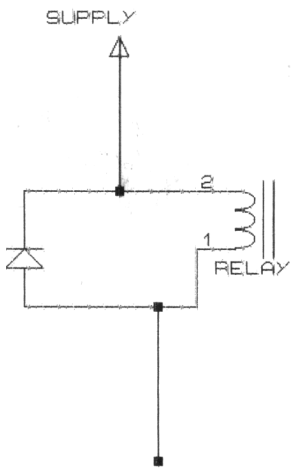
Each input has an anti-bounce or recognition period associated with it. This can be set to a value of 0 to 2.55 seconds.

Outputs

Outputs are open collector and can be used to drive relays solenoid LEDs etc. Optional LED and relay modules are available which plug directly into the 10 way header.



Note that if these outputs are to drive inductive loads such as relays and motors then the load must have some form of back EMF suppression fitted. IF SUPPRESSION IS NOT FITTED THEN THE OUTPUT DEVICE WILL BE DAMAGED. The following diagram shows how to interface the VIOM output to a relay.



Analogue Inputs

The analogue inputs cover the range 0 to 5 Volts. If a larger range is necessary then it will be necessary to use a potential divider on the input.

Measurements are referenced to the internal 5 Volt supply rail which has an accuracy of 5%. The resolution of the A/D converter is 10 bits. All reported values for the inputs will be in the range 0 to 1023.

For the highest accuracy the input impedance should be less than 2K ohms. Higher values are possible but the accuracy will be reduced and there will be increased noise levels.

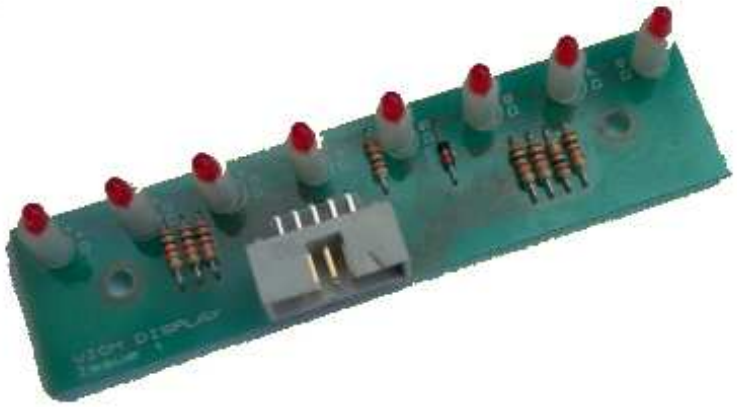
The voltage on the input should not exceed 5 Volts. If higher voltages are possible (e.g. if you are using a potential divider) then you must ensure that voltages higher than 5 Volts cannot be present under start up or fault conditions.

If the driving modules do not share the same power supply as Netiom-xAP then it will be necessary to common their 0 Volt lines.

Hardware Accessories

Display Module

The display module consists of 8 LEDs which can be connected directly on to the Netiom-xAP controller. Connection is via a ribbon cable to the output header. If all 16 outputs are to be displayed then two VIOM Display modules will be required.



Relay Module

The Relay module consists of 8 single pole change over relays which can be connected directly on to the Netiom-xAP controller. Connection is via a ribbon cable to the output header. If all 16 outputs are to be used then two Relay modules will be required.

Internal Elements

Counters

Each input has associated with it a 16 bit counter. Each time the input goes high the counter is incremented. At power up all counters are set to zero. The counters can be reset individually or collectively using a command from a browser.

Latches

Each input has two latches, a high latch and a low latch. The high latch is set when the input goes high and the low latch is set when the input goes low. The latches can be reset individually or collectively using a command from a browser.

Static Web Pages and Elements

Static web pages and elements are those that are not dynamically modified by Netiom-xAP. That is they cannot contain any information about the status of inputs and outputs.

Netiom-xAP supports the following static file types:

Static HTLM (.htm, .txt)
Graphic (.jpg, .gif)
Audio (.wav)
Java (.cla)

If a browser requests a specific file Netiom-xAP will return that file if it exists. If it does not exist it will return a “404 Not found” error message. For example

http://192.168.0.6/XYZ.htm

will return the file XYZ.htm if it exists.

If a file is not specified then Netiom-xAP will return Index.htm. Thus

http://192.168.0.6/

will be interpreted as:

http://192.168.0.6/Index.htm

If Index.htm is not included in you web pages then a “404 Not found” error message will be returned.

One web page may reference several other pages and elements. For example the index page may contain some images and, if frames are used, other dynamic and static pages. When the browser encounters these references it will request all the references at the same time. Netiom-xAP can only service 4 requests at a time and if more are requested they may not be serviced. Therefore it is better to design your pages so that a browser will request no more than 4 elements at a time.

Netiom-xAP will only support short name files in 8.3 format. So a file named MyLongNameFile.htm will not be recognised.

Dynamic Web Pages

Dynamic web pages are those that Netiom-xAP will modify at the time of request. All dynamic web pages should use the .cgi extension otherwise Netiom-xAP will assume that they are static pages and will not modify them.

Tags are 3 character elements within the web page. The first character is always a “%” and is followed by a 2 digit numerical code. When Netiom-xAP encounters a tag it will replace the three characters with the processed data. For example:

Analogue Input 1 %01

Will appear as

Analogue Input 1 123

on a web page (where 123 is the current value of analogue input 1).

All tags must have a two digit code. %1 would be invalid and would be replaced with an “E”.

Dynamic web pages are also used in sending e-mails (see the relevant section). These special pages use the same tags as the normal pages.

A full list of tags is given in Appendix 1

Commands

Commands can be sent either from a web browser.

Each command consists of a single upper case alpha character and two numerals.

When sending data from a browser it would be normal to place buttons on a form. Netiom-xAP only supports the HTTP GET method so the form must specify this method. A single button on a web page might look something like:

```
<FORM METHOD=GET action="commands.cgi">

    <form>

    <input type=submit name=T01 value=" Toggle 1 ">

    </form>
```

The command uses the name field of the button, in this case T01 (toggle output 1). Note that only one command can be sent at a time.

A full list of commands is provided in Appendix 2.

Using Netiom-xAP on the Internet

Netiom-xAP can work on wide area networks such as the Internet as well as local networks. Dial up networks using modems are not suitable for accessing Netiom-xAP.

When using xAP over the internet a xAP router is used to convert the UDP protocol to TCP for assured delivery. A free xAP router can be downloaded from www.mi4.biz.

The local network will be connected to the Internet via a gateway. This may be a computer or an ADSL router or some similar device. The gateway will have two IP addresses, one which it will present to the local network and one which it will present to the Internet. The Internet IP address will be supplied by your ISP.

If you are connecting to Netiom-xAP via the Internet then you will need to use the Internet IP address. You will also need to tell the gateway to route incoming traffic to Netiom-xAP. This is called “Port Forwarding”. Thus if you are using Netiom-xAP as an HTTP server using port 80 you will need to set the gateway “Port Forwarding” on port 80 to the Netiom-xAP IP address. Then any port 80 requests the gateway receives from the Internet will be forwarded to Netiom-xAP.

Note that if you are using only e-mail “Port Forwarding” will not be required. Port forwarding details when using a xAP router are provided with the appropriate download.

Troubleshooting

Problem: No LEDs are illuminated

Solution: This is likely to be a power problem. Check that power is connected and that it is the correct polarity.

Problem: The green LED is off but the red and green are on.

Solution: This is an Ethernet connection problem. Make sure that the Ethernet cable is firmly in position at both ends. Is the router or hub is switch on?

Is the cable good and is it the correct type? There are two types of Ethernet cables, straight through and cross over. If you are connecting into a hub or router you should use a straight through type. If you are connecting directly to a PC you will need to use a cross over type.

Problem: I get a “404 Not found” message when I connect through a browser.

Solution: the file you have requested is not present. If you have just entered the IP address then check that you have included a file called “Index.htm” in the file list.

Netiom-xAP will only support short name files in 8.3 format. Make sure that all your files conform to this format.

Problem: The tags do not seem to work in my web pages.

Solution: Tags will only work in files with a .cgi extension. Make sure that you are only using tags in files with that extension. Make sure that you are using valid tags. Remember %1 is not valid, it must be %01.

Problem: Netiom-xAP does not send e-mails.

Solution: The first step is to see if Netiom-xAP is attempting to send the e-mail. Attempt to send an e-mail and observe the red LED. Make sure that there are no HTTP accesses with you are doing this test.

If the red LED does not flicker then Netiom-xAP is not attempting to send the e-mail. Read back your settings in the Netiom-xAP Interface and make sure that e-mail is enabled. Also check that the trigger you are using is enabled.

If the red LED flickers only once (this may repeat after 10 seconds) then Netiom-xAP cannot find the e-mail server. Check that the IP address of the server is correct. If you have a hardware firewall installed it may be blocking access to the Internet or it might only allow you access to certain parts of the Internet.

If the red LED flickers several times over a period of about 1 second then it is likely that Netiom-xAP is connecting to the e-mail server but its request is being denied for some reason. Enable the “View Server Messages” option and observe the output from the serial port in HyperTerminal when you generate an e-mail.

If you get a “Request Denied” or similar message then you do not have access to that server.

If you get a “Authentication Failed” or similar message then either the password or account name is wrong.

If you get a * AUTH mechanism LOGIN not available” it is likely that the log in procedure is not supported by Netiom-xAP.

If you get a “Message accepted for delivery” the e-mail should have been sent. It may have simply been delayed.

Problem: When I send an e-mail and I have “View Server Messages” option enabled I see a “AUTH mechanism LOGIN not available” message followed by other error messages. The e-mail seems to be sent OK.

Solution: In this case the e-mail server does not have any authentication procedures in place. This may leave it open to spammers using it as a relay. It is nothing to concern Netiom-xAP.

Phaedrus Limited
Unit 1
Darwen Enterprise Centre
Railway Road
Darwen
BB3 3EH
U.K.

Tel / Fax +44 (0)1254 772622
e-mail tech@phaedrusltd.com

Issue 1
April 2005

Appendix 1: Dynamic Web Page Tags

%00	Last serial message received.
%01	Analogue Input 1 value (0 to 1024)
%02	Analogue Input 2 value (0 to 1024)
%03	Analogue Input 3 value (0 to 1024)
%04	Analogue Input 4 value (0 to 1024)
%05	State of Digital Inputs 1 to 8 (e.g. 10000000, Input 1 high others low)
%06	State of Digital Inputs 9 to 16 (e.g. 10000000, Input 9 high others low)
%07	State of Outputs 1 to 8 (e.g. 10000000, Output 1 high others low)
%08	State of Outputs 9 to 16 (e.g. 10000000, Output 9 high others low)
%09	State of Virtuals 1 to 8 (e.g. 10000000, Virtual 1 high others low)
%10	State of Virtuals 9 to 16 (e.g. 10000000, Virtual 9 high others low)
%11 to %26	Input 1 to 16 Counter value (0 to 65535)
%27	State of Inputs 1 to 8 Low Latch (e.g. 10000000, Input 1 latched)
%28	State of Inputs 9 to 16 Low Latch (e.g. 10000000, Input 9 latched)
%29	State of Inputs 1 to 8 High Latch (e.g. 10000000, Input 1 latched)
%30	State of Inputs 9 to 16 High Latch (e.g. 10000000, Input 9 latched)
%31	Insert Device Name
%41 to %56	Input 1 to 16 status. Reports as “On” or “Off”.
%61 to %76	Output 1 to 16 status. Reports as “On” or “Off”.
%81 to %96	Virtual 1 to 16 status. Reports as “On” or “Off”.
%99	Add a carriage return/line feed pair

The %99 tag is only really useful for e-mail body files (email.cgi) and client files (client.cgi). Browsers do not take any notice of carriage return / line feed pairs so they are removed from all the web page files to maximise the available memory space and speed downloads. Using this tag in the e-mail body can make it easier to read. Using it in Client files can make it easier to parse different sections of the file.

Appendix 2: HTTP Commands

A00	Turn all Outputs on.
A01 to A16	Turn a single Output (1 to 16) on.
B00	Turn all Outputs off.
B01 to B16	Turn a single Output (1 to 16) off.
C00	Clear all counters.
C01 to C16	Clear a single counter (1 to 16).
D00	Turn all virtuals on.
D01 to A16	Turn a single virtual (1 to 16) on.
E00	Turn all virtuals off.
E01 to B16	Turn a single virtual (1 to 16) off.
H00	Clear all high latches.
H01 to H16	Clear a single high latch (1 to 16).
L00	Clear all low latches.
L01 to L16	Clear a single low latch (1 to 16).
S00(Message)	Send the message to the serial port with carriage return / line feed.
S01(Message)	Send the message to the serial port.
T00	Toggle all Outputs.
T01 to T16	Toggle a single output (1 to 16).
U00	Toggle all virtuals.
U01 to U16	Toggle a single virtual (1 to 16).

Appendix 3: Node id's

00	Base address
01 to 10	Inputs 1 to 16
11 to 20	Outputs 1 to 16
21 to 30	Virtuals
31 to 40	Counters
41 to 50	Low latches
51 to 60	High Latches
61 to 64	Analogue inputs
65	Serial Port Input
66	Serial Port Output

Appendix 4: ASCII Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[		]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL