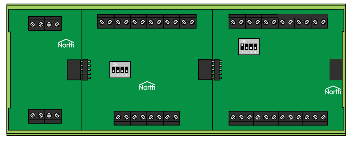
Zip Tutorial



This tutorial shows how to use the main features of Zip. It covers using inputs to measure the environment, and outputs to control equipment, as well as introducing North technology such as Commander and the Start Engineering applications.

You should have already completed the *Commander Tutorial*. Work through this tutorial using the North Training Pack, containing: ObSys software, Commander and Zip modules.

This tutorial relates to the Commander v2.0 (build 17/08/22) base software, distributed with the September 2022 release of North ObSys.

Contents

What is Zip?

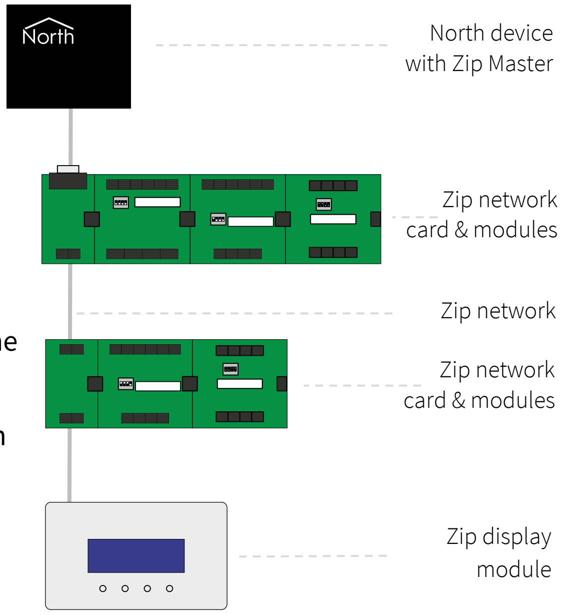
Control systems need connections to the real world – inputs to measure the environment, and outputs to control equipment. Zip is North's measurement and control system.

Zip modules perform the actual measurement and control and are available with different capabilities to match the building's requirements. Some provide general-purpose inputs for measuring, such as volt-free contacts, thermistors, and 0-10 volt analogues. Some provide general-purpose outputs for control, including relays, switched 12-Volt digitals, and 0-10 Volt analogues. Some perform a fixed-function, such as controlling door-access or displaying text.

Zip uses a network to connect all the modules together, so the engineer can distribute the modules where they are needed within a building. Fixed-function modules connect directly to the network, and general-purpose modules share a connection using a Network Card.

The Zip network is the link between the modules and the controller, called Zip Master. Any North device, such as Commander or ObSys, can be the Zip Master for a network.

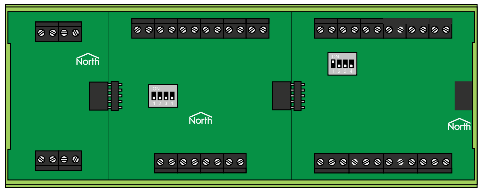
In addition, because the controller is a North device, the engineer can employ North's interface technology to link the Zip system to other external systems, including BACnet and Modbus.


North
North device with Zip Master
Zip network card & modules
Zip network
Zip network card & modules
Zip display module

Zip Modules

A range of different Zip modules is available. Some provide general-purpose inputs and outputs, and some perform a fixed-function such as controlling door-access or displaying text. The engineer chooses which they need, and where to place them within the building.

General-purpose Modules



Zip's general-purpose modules make panel design and construction simple and give the engineer most control over the inputs and outputs.

General-purpose modules clip together to form a single DIN-rail mounting block with the right amount of input and output for a control panel. They also save panel wiring, with up to four modules sharing the network and power connections of a Zip network card.

Fixed-function Modules

Common problems can be solved easily with Zip's fixed-function modules – these perform straight from the box and require no cause-and-effect engineering. Modules include the M7101 door controller, and M7204 user display.

Fixed-function modules have a built-in network card and require connecting directly to the Zip network and power supply.

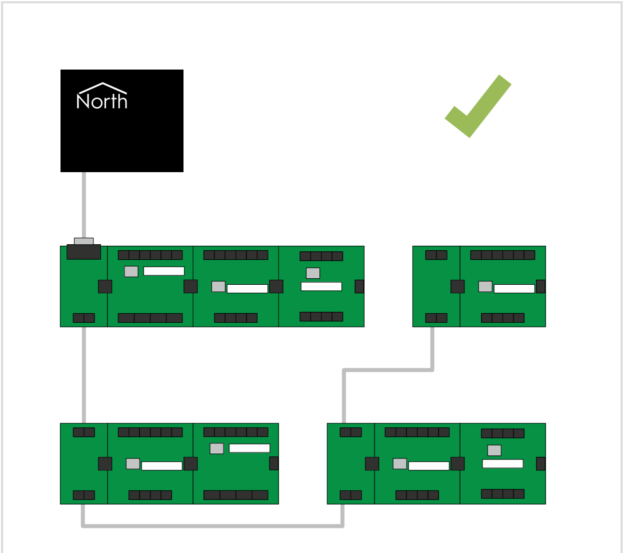
Zip Network

Zip modules connect to the North device with Zip Master over a shared network. RS485 forms the base of this network, which is a two-wire bus that can span up to one kilometre end-to-end. Up to 16 Zip modules and a North device can sit anywhere along the network.

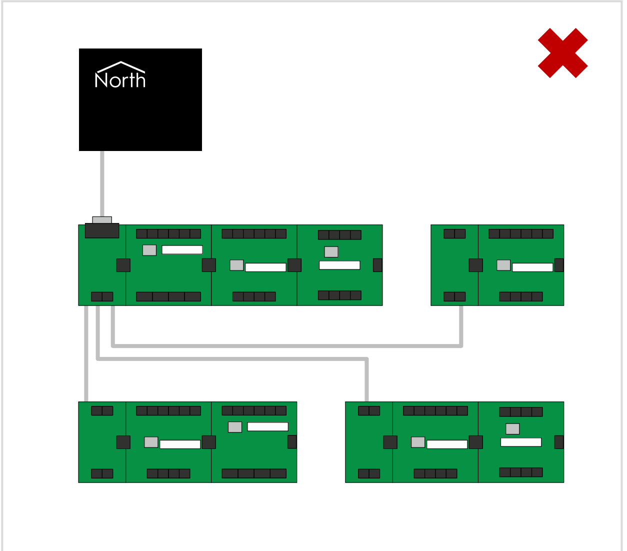
The Zip network is polarity-dependent with one positive and one negative wire. Over the whole network, the positive wire attaches to the positive terminals marked '+', and the negative wire attaches to the negative terminals marked '-' on the ZIPNET connector.

The Zip network must be wired as a single-line bus with only two ends. Do not use star or ring wiring topologies, T-junctions, or spurs. Termination resistance is typically not required.

Recommended cables for wiring the Zip network are Belden 8761, Alpha Wire 2401C, or similar shielded twisted -pair cable (minimum 22AWG). Avoid running network cables parallel to power cables, especially if they carry large currents or the power is switched frequently. To simplify installation even more, Zip modules are optically isolated from the network. This protects local equipment from high voltage spikes caused by lightning and simplifies electrical installation.



Correctly wired Zip network



Incorrectly wired Zip network

Zip Master

The ZipMaster driver, running within a North device such as Commander and ObSys, controls the network of Zip modules.

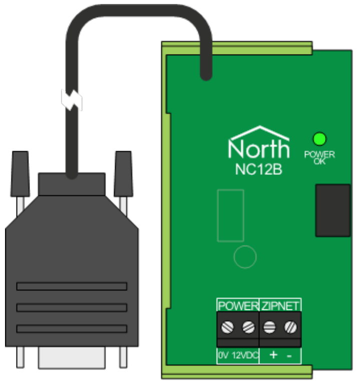
The North device also contains our interface technology, ObVerse cause-and-effect language, and easy-to-use web services. Alongside Zip, the North device can work as a stand-alone controller, or as part of a larger control and monitoring solution.

The North device also acts as a gateway to IP networks, including the internet. Zip inputs can trigger communications across these networks. Zip outputs can be controlled from other devices on these networks.

Training Pack Zip Contents

The North Training Pack contains items of Zip: an NC12B network card, an M7002A module, an M7007A module, and some example sensor and output devices. Let's look closer at some of these items…

NC12B Network Card



The NC12B network card connects a North device with Zip Master to the Zip network. The network card provides power and Zip network for up to four general-purpose modules via the five-way socket.

Power

There is one 12V DC power input, labelled POWER. The connector is polarity dependent.

The power input has a two-way connector labelled **0V** and **12VDC**. Connect a regulated 12V DC (±5%) 1A power supply. This allows 250mA for each connected module. The **POWER OK** light will illuminate green to indicate a healthy power.

Zip Network

There is one isolated Zip network connection, labelled ZIPNET.

The network connection has a two-way connector labelled **+** and **-**. Over the whole network, the positive wire attaches to the positive terminals marked '+', and the negative wire attaches to the negative terminals marked '-'.

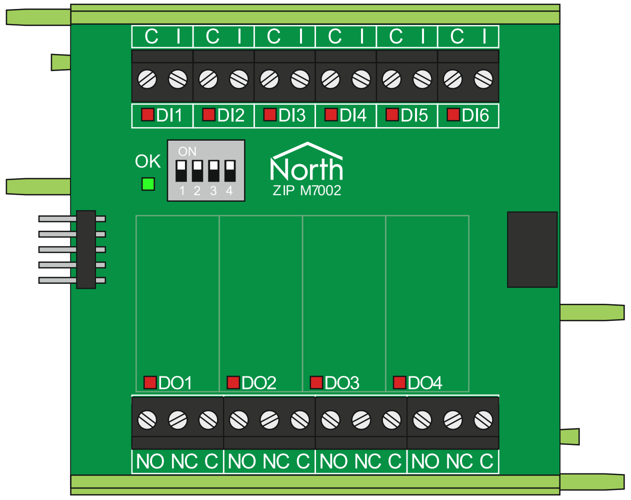
RS232

There is a 0.5m RS232 cable, with DB9 connector, attached to the network card. Connect this to a North device's COM port.

M7002A Digital Input Relay Output Module

The M7002 is a general-purpose module with six digital inputs and four relay outputs.

The module needs to connect to a network card or module via the five-way plug for access to power and Zip network.



Digital Inputs

There are six volt-free digital inputs, labelled DI1…DI6.

Each input has a two-way connector labelled **C** (common) and **I** (input). Each input also has a red LED, which lights when the contact closes. Each input senses the state of the contact and counts state changes (open-to-closed). Change rates up to 25Hz can be sensed (changes faster than this are taken to be noise).

Connect the digital input to a volt-free mechanical contact, e.g. switch, push button, relay output, pulse output meter, etc.

Relay Outputs

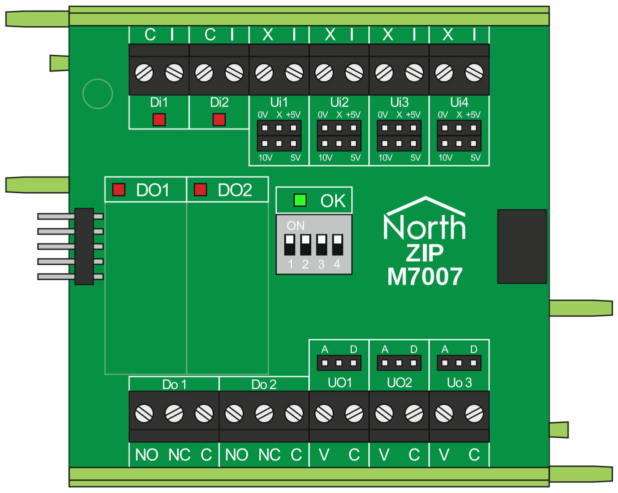
There are four relay outputs, labelled DO1…DO4.

Each output has a three-way connector labelled **NO** (normally open), **NC** (normally closed) and **C** (common). When the output is set to 'Off' or the module has no power, the relay is de-energised connecting C and NC. When the output is set to 'On', the relay energises connecting C and NO, and lighting the red LED.

Each relay is rated 240V AC/28V DC at 10A resistive load. If higher loads are required, the relay can switch an external contactor.

Connect the relay, for example, in-line with the power supply to an appropriate load such as a fan motor, motorised valve, lighting circuit, etc.

M7007A Mixed Input Output Module



The M7007A is a general-purpose module with a mix of different input-output capabilities. It has two digital inputs, four universal inputs, two relay outputs, and three analogue outputs.

The module needs to connect to a network card or module via the five-way plug for access to power and Zip network.

Digital Inputs

There are two volt-free digital inputs, labelled DI1…DI2.

Each input has a two-way connector labelled **C** (common) and **I** (input). Each input also has a red LED, which lights when the contact closes. Each input senses the state of the contact and counts state changes (open-to-closed). Change rates up to 25Hz can be sensed (changes faster than this are taken to be noise).

Connect the digital input to a volt-free mechanical contact, e.g. switch, push button, relay output, pulse output meter, etc.

Universal Inputs

There are four universal inputs, labelled UI1…UI4.

Each input has a two-way connector labelled **X** (external) and **I** (input), and two jumpers. Use the top jumper to select the output voltage at connector X: 0V (left position) or 5V DC (right position). Use the bottom jumper to select the measurement range of the input signal voltage connected to I: 5V (right position) or 10V DC (left position).

Connect the universal input to a range of sensor types, including 0-10V sensors, 0-5V sensors, thermistors (10K3A type), and volt-free contacts such as relays and switches. When the 0-10V range is selected, and an external 500Ω resistor is connected, they can also measure 0-20mA sensors (active type only). The 0-5V range, along with two external 1kΩ resistors, can provide monitored digital inputs (which sense contact states and connection faults).

Relay Outputs

There are two relay outputs, labelled DO1…DO2.

Each output has a three-way connector labelled **NO** (normally open), **NC** (normally closed) and **C** (common). When the output is set to ‘Off’ or the module has no power, the relay is de-energised connecting C and NC. When the output is set to ‘On’, the relay energises connecting C and NO, and lighting the red LED.

Each relay is rated 240V AC/28V DC at 10A resistive load. If higher loads are required, the relay can switch an external contactor.

Connect the relay, for example, in-line with the power supply to an appropriate load such as a fan motor, motorised valve, lighting circuit, etc.

Universal Outputs

There are three universal outputs, labelled UO1…UO3.

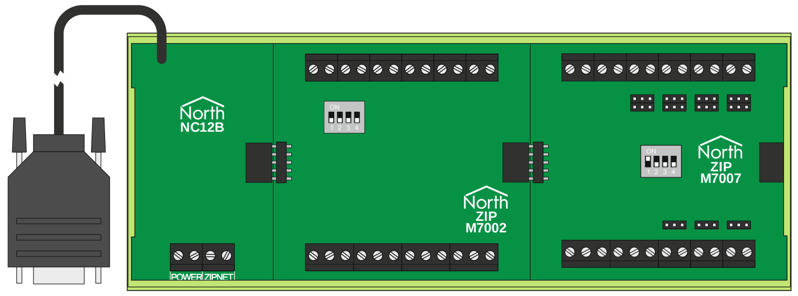
Each output has a two-way connector labelled **V** (voltage) and **C** (common). Use the jumper to select the output type: A (analogue, left position), or D (digital, right position). Analogue outputs generate a variable voltage in the range 0-10V DC at 10mA, while digital outputs switch 12V DC at 50mA on the V connector.

Connect the universal output to 0-10V DC motorised valve, 0-10V input on a variable frequency drive, 12V DC coil on an external relay, 12V DC indication lamp, etc.

Assembling and powering the Training Pack Zip components

This tutorial is written assuming you have access to a set of Zip modules. If you are using the Zip components from the Training Pack, they might need assembling.

Zip components typically require connecting, so that a Zip module can share a NetCard's power supply and Zip network. The components are made up of a circuit board with input and output connectors, and green carrier. Each module is supplied with the correct amount of green carrier for the module. Each NetCard is also supplied with two green carrier endcaps.



To assemble the Zip modules in the Training Pack, follow these steps:

→ Remove the components from the NC12B, M7002, and M7007 boxes, and slide the circuit boards from the green carrier

→ Clip together the green carrier, except for one end-cap.

→ Slide into the assembled carrier the NC12B first, then the M7002, then the M7007, ensuring that the 5-pin connection on each module slots into the 5-plug connector of the previous

→ Add the final endcap to hold everything in place. This keeps the circuit boards connected, and prevents you touching the underside of any modules when they are in use.

→ With the NC12B on the left, set the address switches of the M7002 to Off-Off-Off-Off (address 0) and the M7007 address switches to On-Off-Off-Off (address 1)

To check the address of each Zip module in the training pack, follow these steps:

→ Set the M7002 module to address 0, by setting the switches S1: Off, S2: Off, S3: Off, S4: Off.

→ Set the M7007 module to address 1, by setting the switches S1: On, S2: Off, S3: Off, S4: Off.

To connect power to the training pack, follow these steps:

→ Remove the power connector plug from the NC12B network card

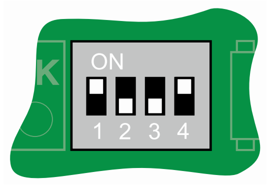
→ Check the polarity of the supplied 12VDC PSU is correct, before attaching it to the power connector plug using the screw terminals. The network card is marked to show the polarity of the power connector

→ Attach the power connector plug into the network card, before applying power to the PSU.

→ Check that the network card's POWER OK LED is lit. If the POWER OK LED does not light, check the polarity and voltage of the PSU output.

Common Zip Module Features

Address Switch



Address	S1	S2	S3	S4
0	Off	Off	Off	Off
1	On	Off	Off	Off
2	Off	On	Off	Off
3	On	On	Off	Off
4	Off	Off	On	Off
5	On	Off	On	Off
6	Off	On	On	Off
7	On	On	On	Off
8	Off	Off	Off	On
9	On	Off	Off	On
10	Off	On	Off	On
11	On	On	Off	On
12	Off	Off	On	On
13	On	Off	On	On
14	Off	On	On	On
15	On	On	On	On

Each Zip module must have a unique address on the Zip network. Using the module's binary address switches S1 to S4, set an address in the range 0 to 15 using the table shown below.

For example, to set the module to address 9 then position the switches S1: On, S2: Off, S3: Off, S4: On (as shown in the diagram.)

The order of the addresses on the network doesn't matter, but it makes sense to work to some plan.

OK Light

Each Zip module indicates, using a green OK light, if it is powered-on and communicating reliably with Zip Master:

On – module power is healthy, and it is communicating with Zip Master.

Blinking – module power is healthy, but it has not received a valid message from Zip Master. Check the network connection.

Off – problem with the module power. Check the power supply.

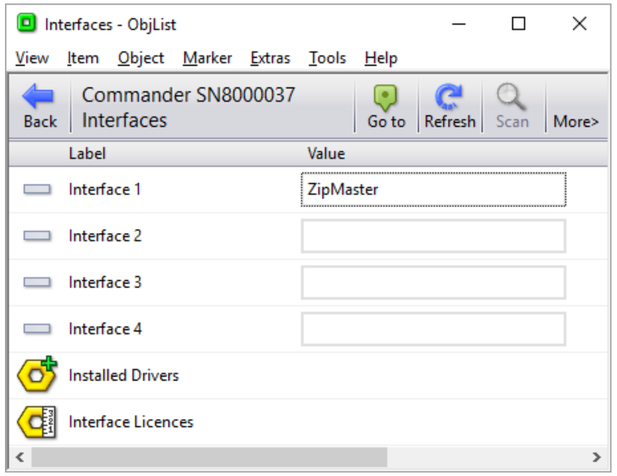
If a module cannot maintain reliable communications with Zip Master, after 30 seconds it will drive its outputs to a default state. Outputs are typically set to off or 0V.

Connecting Commander to Zip

You will have performed this in the Commander Tutorial.

Before we can start to use the Zip system, we need to start an interface within Commander to act as the Zip Master. We will also need to physically connect Commander to the Zip network card, using a RS232 COM Port.

Starting an Interface with the ZipMaster Driver



Label	Value
Interface 1	ZipMaster
Interface 2	
Interface 3	
Interface 4	
Installed Drivers	
Interface Licences	

The ZipMaster driver is installed on all North devices. We will use this driver to start an interface between Commander and Zip.

💻 To set up Interface 1 to connect to a Zip system, follow these steps:

→ From the top-level of Commander, select **Configuration**, **Interfaces** to view the interfaces currently set up – and the drivers they use – if they are all blank, then there are no interfaces. Open **Installed Drivers** to view a list of drivers currently installed in the Commander

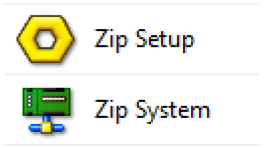
→ Scroll up and down and find the **Installed Driver** object that has the value 'ZipMaster'. Copy this value by right-clicking on the object and selecting **Copy Value** from the popup menu. (You can also copy the object's value by clicking to highlight the object, and pressing CTRL+C on the keyboard)

Press **Back**, then right-click on **Interface 1,** and select **Paste Value** from the popup menu. This will set the value to 'ZipMaster'. (You can also paste to an object's value by clicking to highlight the object and pressing CTRL+V on the keyboard).

Interface Set up

When Commander starts an interface, the interface normally adds two new objects to the top-level of Commander – you will therefore need to **Scan** the top-level.

The first object that the interface adds is the **Zip Setup** object, which has a yellow hexagonal nut icon (see the icon to the right). This is where values relating to the driver's operation are – things like RS232 port numbers, baud rates, and system labels. The second **Zip System** object added to the top-level of ObServer represents the Zip network that can be accessed using the driver.



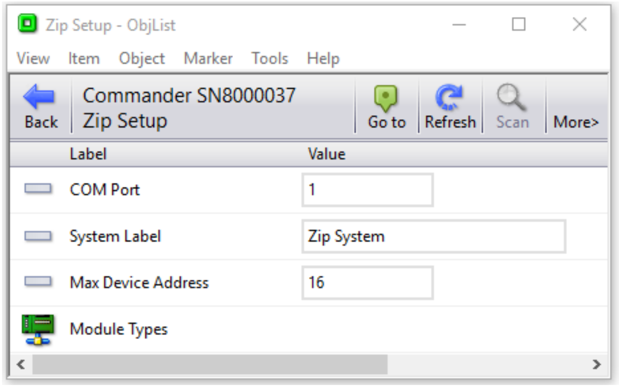
💻 To configure the Zip interface for the Zip modules in the training pack, follow these steps:

→ Open **Zip Setup**, (you may need to **Scan** the top-level object if you have just added the interface) to view the Zip setup objects

→ Set **COM Port** to '1', to tell the driver to use COM1 on Commander

→ Use the RS232 cable supplied, to connect the Commander's COM1 port to the NC12B's 9-way port

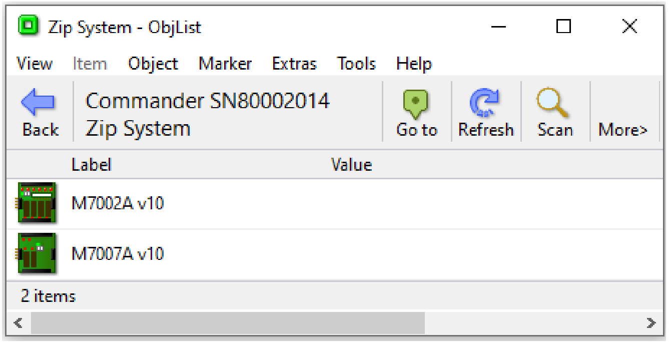
→ The OK Light of each Zip module should now light solid on to indicate that it has a link with the Zip Master.



Label	Value
COM Port	1
System Label	Zip System
Max Device Address	16
Module Types	

Note: A blinking OK light on a module means it has no communications link with the Zip Master - check the *wiring* and the *address switches*.

A Quick Tour of the Zip System



Label	Value
M7002A v10	
M7007A v10	

2 items

The interface we just started is communicating with the Zip modules in the training pack. The driver is polling each Zip module in turn, watching for changes. Let's see what the Zip system looks like from ObView.

To find modules on the Zip network, follow these steps:

→ Open **Zip System** – it will be empty because we have never scanned the system

→ Press **Scan** to instruct ObView to scan the network for modules. After it has finished scanning it will display the two modules – the M7002A and the M7007A

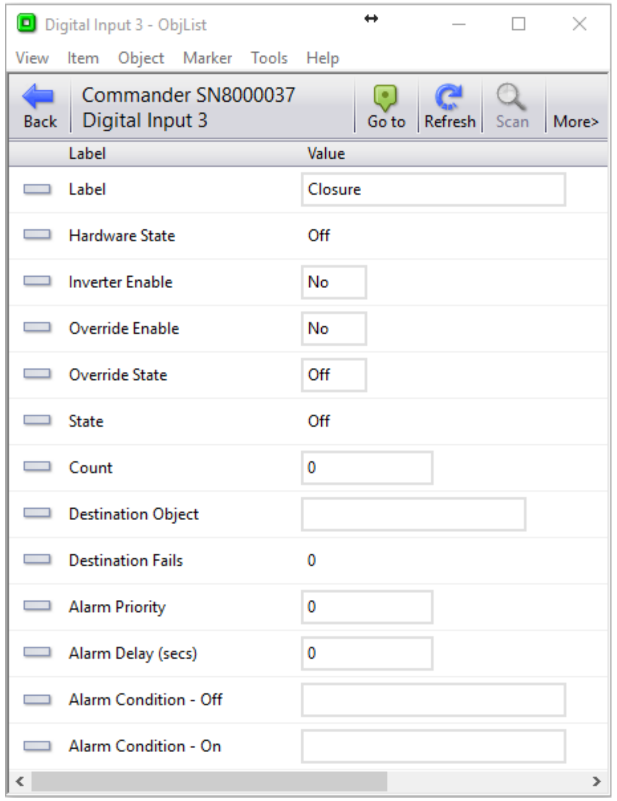
→ Open **M7002A** to see the Zip M7002 module's sub-objects

→ Open **Module Information** to view general information about the module – its label, if communications are ok, and the number of restarts

→ Navigate **back** to the M7002A module again and open **Digital Input 1** to see its label, hardware state, current state after processing, count of the closures, destination object information, and alarm information. We will set all these later in the tutorial.

→ Navigate **back** to the M7002A module and open **Digital Output 1** to see its label, current state, override information, and the hardware state. These will also be set later in the tutorial.

Sensing a Contact Closure


Digital Input 3 - ObjList

View Item Object Marker Tools Help

Back | Commander SN8000037 Digital Input 3 | Go to | Refresh | Scan | More>

Label	Value
Label	Closure
Hardware State	Off
Inverter Enable	No
Override Enable	No
Override State	Off
State	Off
Count	0
Destination Object	
Destination Fails	0
Alarm Priority	0
Alarm Delay (secs)	0
Alarm Condition - Off	
Alarm Condition - On	

The simplest input type to use is the M7002's digital input. We will use it here to sense a contact closure. We need to short together the inputs, and view the digital input using ObView.

To set up a simple digital input, follow these steps:

→ Navigate to the **M7002A**, and then the **Digital Input 3**

→ Set the **Label** to 'Closure', to remind yourself the purpose of the test

→ Make a link by removing the plug from DI3 (a two-way connector on the M7002) and link the two screw-terminals using wire – this will save time!

→ Using the link, connect together DI3's **I** (input) and **C** (common) – the red **DI3 LED** should light, the ObView object **Hardware State** and **State** should change to 'On', and the **Count** object should increment.

→ Remove the link to disconnect the connectors – the LED should go off and the Hardware State should be 'Off'.

To invert the digital input state, follow these steps:

→ Set **Inverter Enable** to 'Yes' – the **State** object will show the inverted **Hardware State**

→ Connect and disconnect the link – the Count increments when the Hardware State changes from 'Off' to 'On'.

As you can see, the **State** object is the one we have control over, and therefore used by other tasks – the **Hardware State** always shows the real hardware state.

To override a digital input state, follow these steps:

→ Set the **Override State** to 'On' – this is the value that we want when we override the input, and set the **Override Enable** to 'Yes' – the Hardware State is ignored, and the State value follows the Override State value – again the Hardware State always shows the real hardware state

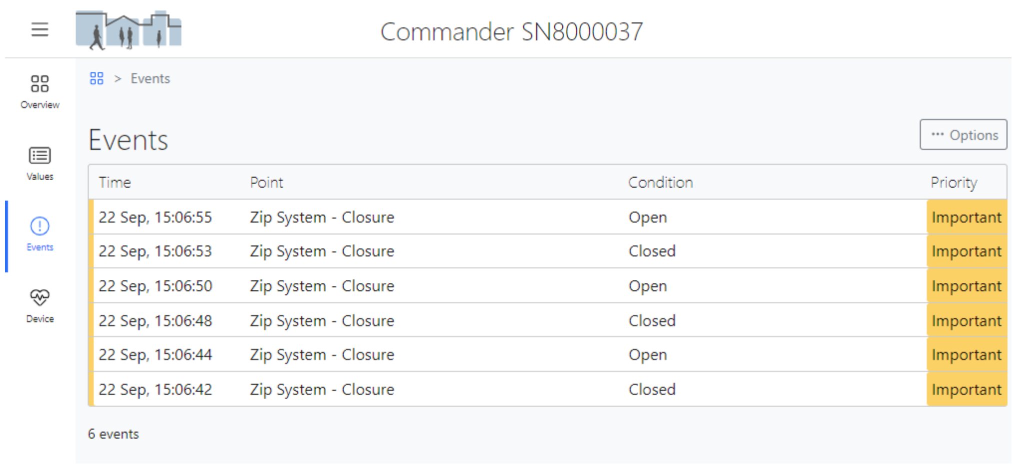
→ Remember to set Inverter and Override Enable to 'No' to when you are finished testing!

Generating an Alarm Message on Contact Closure

We can set the digital input to send an alarm message when the input closes ('On' state) and/or when the input opens ('Off' state). By default, Commander sends alarms to the alarm history, which can be viewed using the web interface.

To view the Commander's alarm history, follow the steps:

→ Navigate to the **top-level** of Commander
→ From the ObView menu, select **Extras > View in Browser**
→ From the side menu of the web page, select **Events**



Events

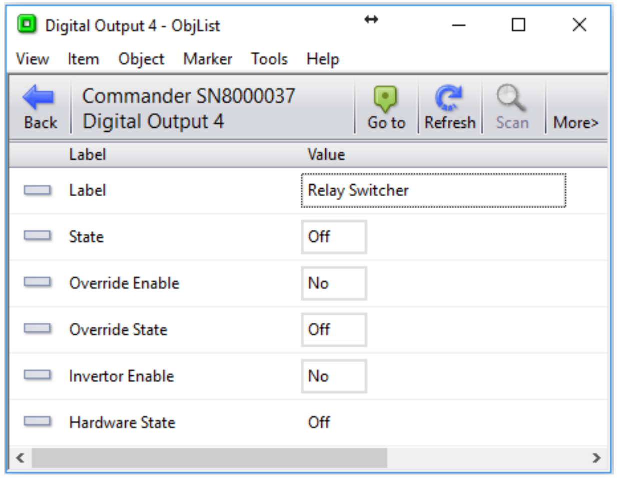
Time	Point	Condition	Priority
22 Sep, 15:06:55	Zip System - Closure	Open	Important
22 Sep, 15:06:53	Zip System - Closure	Closed	Important
22 Sep, 15:06:50	Zip System - Closure	Open	Important
22 Sep, 15:06:48	Zip System - Closure	Closed	Important
22 Sep, 15:06:44	Zip System - Closure	Open	Important
22 Sep, 15:06:42	Zip System - Closure	Closed	Important

6 events

To set Digital Input 3 to send an alarm on closure, follow the steps:

→ Navigate to **Zip System**, **M7002A**, **Digital Input 3** (labelled Closure)
→ Set **Alarm Priority** to '3' (important) – the priority of the alarm to be sent - where 0 means no alarm, 1 (critical) means highest priority, and 9 means lowest priority
→ Set **Alarm Condition – On** to 'Closed' – the text to send when the state changes to 'On'
→ Insert the link into DI3 to set the state to 'On' and generate the alarm message. Remove and insert the link to cause new alarms. You should see these on the events web page
→ Set **Alarm Condition – Off** to 'Open', and re-test – alarm messages are now sent when the link is inserted or removed

Switching a Relay



Digital Output 4 - ObjList

View Item Object Marker Tools Help

Back | Commander SN8000037 Digital Output 4 | Go to | Refresh | Scan | More>

Label	Value
Label	Relay Switcher
State	Off
Override Enable	No
Override State	Off
Invertor Enable	No
Hardware State	Off

The simplest output to control is an M7002's relay output. We will use ObView to open and close it.

To set up a relay output, follow these steps:
→ Navigate to the **M7002A**, **Digital Output 4**
→ Set **Label** to 'Relay Switcher'.

To turn the relay state on, follow these steps:
→ Set **State** to 'On' – the **Hardware State** object should change to 'On', the relay should click, and the red **DO4 LED** will light (positioned between the actual relay and its 3-way connectors).

As with the digital inputs, the output can have an inverter, which causes the Hardware State to be set to the opposite of the State object. It also has an override option, which causes the Hardware State to be controlled by the Override State object.

Note: if the M7002 has no power, or loses network connection to the controller, the relay will revert to 'inactive' operation, so choose carefully whether to use the normally open (NO) or normally closed (NC) contacts.

Using a Closure Input to Switch a Relay

When we engineered the digital input, we saw but didn't use the State. The digital input's State value is available to any task that can read the object – ObVerse or transfers for example. However, for speed, the digital input can write the state, when it changes, to another object. We will use this facility to explore the speed of Zip and the ZipMaster within Commander.

💻 To set the digital input to write the state to another object, follow these steps:

→ Navigate to **M7002A**, **Digital Input 3** (labelled Closure)

→ Set **Destination Object** to 'S1.M0.DO4.S' – System 1 (Zip), Module 0 (M7002), DO4 (output 1), State (S)

→ Short the contacts of DI3 together to force DI3's **State** to 'On', and the digital input's state will be sent to the destination object, which should then change (the output's relay will click.)

Note: if the Destination Object is set incorrectly, the writing will fail, and the Destination Fails object will increment, up to a maximum of 9.

Using a Thermistor

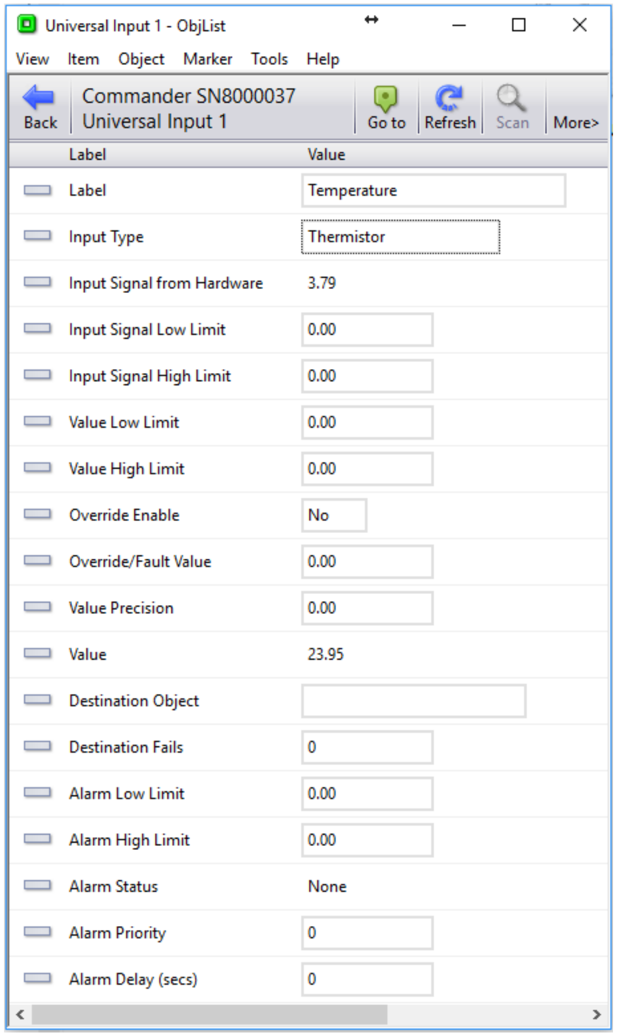
We will set the M7007's universal input to measure temperature using the thermistor supplied with the training pack.

Before we use a universal input, we need to set the two jumpers to select the type of input that is to be connected, in addition to setting the input type in software.

- To set the universal input UI1 for a thermistor, follow these steps:
  - → Remove the connector-block from UI1 on the top edge of the M7007 and fit the supplied 10K3A thermistor (small bead on two wires) ready
  - → Set the top jumper (voltage supply) to link the pins on the right, which sets the X pin to 5-volts – because a thermistor is a variable resistor
  - → Set the bottom jumper (range) to link the pins on the right, which sets the range to 0-5V
  - → Connect the thermistor to the X and I connectors.

- To set the M7007's UI1 for thermistor input, follow these steps:
  - → Navigate to **M7007A**, **Universal Input 1**
  - → Set **Label** to 'Temperature'
  - → Set **Input Type** to 'Thermistor' – the **Value** object will now show the temperature
  - → Hold the thermistor bead to cause a temperature change - Zip Master rescales the input voltage using tables designed for a 10K3A thermistor.

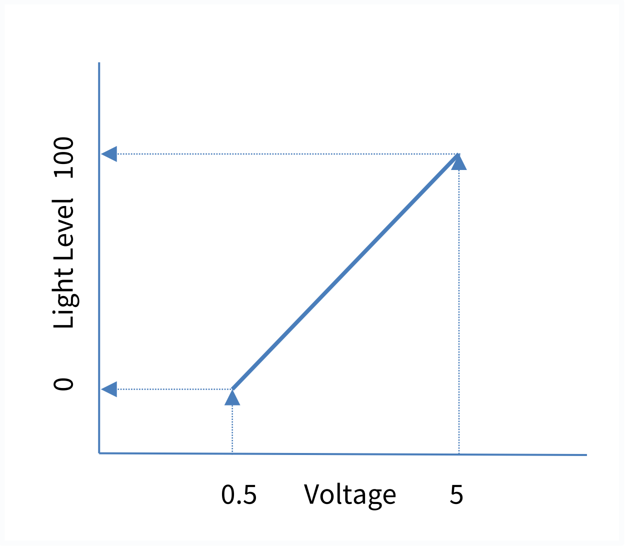
- To set an override value for sensor failure, follow these steps
  - → Set **Override Value** to '21' – the value to be used when the Override is enabled
  - → Set **Override Enable** to 'Yes' and Value becomes 21
  - → Set **Override Enable** to 'No' to disable the manual override
  - → Remove the thermistor connectors from the M7007, to simulate a sensor failure – notice the Value now becomes '21' - the override value.



Label	Value
Label	Temperature
Input Type	Thermistor
Input Signal from Hardware	3.79
Input Signal Low Limit	0.00
Input Signal High Limit	0.00
Value Low Limit	0.00
Value High Limit	0.00
Override Enable	No
Override/Fault Value	0.00
Value Precision	0.00
Value	23.95
Destination Object	
Destination Fails	0
Alarm Low Limit	0.00
Alarm High Limit	0.00
Alarm Status	None
Alarm Priority	0
Alarm Delay (secs)	0

Note: Value is set to Override Value whenever the input is outside of an acceptable range – with thermistors this is equivalent to -20 … 100 °C.

Scaling a 0-5V Analogue Input



We will set the M7007's universal input to measure a 0-5V input from a light-sensitive resistor (LSR) and rescale this to a light level percentage.

Before we use a universal input, we need to set the two jumpers to select the type of input that is to be connected, in addition to setting the input type in software.

The LSR supplied as part of the training pack has been tested and give the results in the graph: 0.5V when dark (when we want a value of 0%), and 5V when bright (when we want a value of 100%).

💻 To set the universal input UI2 for the LSR, follow these steps:

→ Remove the connector-block from UI2 on the top edge of the M7007 and fit the supplied LSR (flat zigzag-topped device with two wires) ready

→ Set the top jumper (voltage supply) to link the pins on the right – ie. X connector to 5-volts

→ Set the bottom jumper (range) to link the pins on the right – ie. the range to 0-5 volts

→ Connect the LSR to the X and I connectors.

💻 To set the M7007's UI2 for LSR input, follow these steps:

→ Navigate to **M7007A**, **Universal Input 2**

→ Set **Label** to 'Light Level', and **Input Type** to '5V', as the input is always in the range 0-5 volts

→ Set **Input Signal Low Limit** to '0.5', and **Input Signal High Limit** to '5' – these state the acceptable input signal range, outside which the override value is used

→ Set **Value High Limit** to '0', and **Value High Limit** to '100'

→ **Value** will now vary depending on the light level – the scale is not linear, but is good enough for our tutorial

💻 To send an alarm message when a sensor fault occurs, follow these steps:

→ Set the **Alarm Priority** to '2' (severe), to instruct the universal input to send alarms

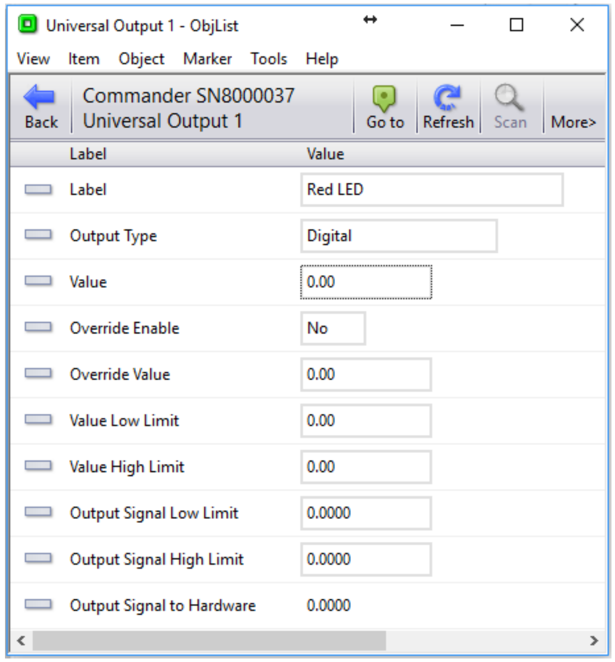
→ Remove the LSR connector from the M7007 – a fault occurs because the input has a reading of 0V, which is out-of-range of the values set in the scale 0 and scale 1 readings – so an alarm is sent and appears on the events web page, and the Value will also be set to the Override Value.

Switching an Output Voltage

We will set the M7007's universal output to turn an LED on and off.

Before we use a universal output, we need to set the jumper to select the type of output that is to be connected, in addition to setting the output type in software.

A red 12-Volt LED, with built-in resistor, is included in the training pack. The LED has a longer leg (the positive anode) and a shorter leg (the negative cathode). The cathode side of the LED also has a 'flat' side.


Universal Output 1 - ObjList
View Item Object Marker Tools Help
Back
Go to
Refresh
Scan
More>

Commander SN8000037
Universal Output 1

Label	Value
Label	Red LED
Output Type	Digital
Value	0.00
Override Enable	No
Override Value	0.00
Value Low Limit	0.00
Value High Limit	0.00
Output Signal Low Limit	0.0000
Output Signal High Limit	0.0000
Output Signal to Hardware	0.0000

💻 To set the universal output UO1 to drive an LED, follow these steps:

→ Remove the output connector from UO1 on the left-bottom of the M7007. Fit the red LED to the connector so the positive anode will connect to the V pin, and the negative cathode will connect to the C pin, when the connector is re-fitted to the M7007 module

→ Set the jumper to link the pins on the right, which sets the output to 'digital'

→ Connect the LED to the V and C connectors.

💻 To set the M7007's UO1 to drive an LED, follow these steps:

→ Navigate to **M7007A**, **Universal Output 1**

→ Set **Label** to 'Red LED' to act as a reminder

→ Set **Output Type** to 'Digital'

→ Set **Value** to '1' to turn the output on – 12 Volts

→ Set **Value** to '0' to turn the output off – 0 Volts.

If you set Value to anything less than 0.5 then the output is set to 0 Volts, otherwise it is set to 12 Volts.

As with other general-purpose outputs, the Override Enable and Override Value will allow secondary control of the actual output.

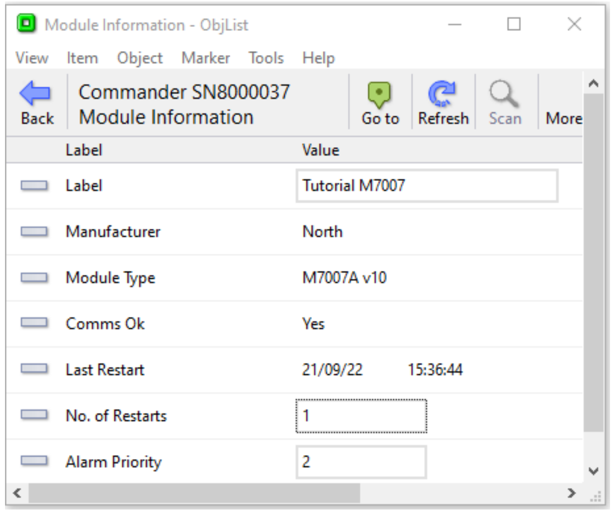
Note: If the module loses communications with the Zip controller, the output will be set to 0-volts.

Scaling a 0-10V Analogue Output

We can set the M7007's universal output to output a variable voltage in the range 0-10V. The easiest way to detect operation of this type is to use a voltmeter – set it to measure DC voltages in the range 0…10 volts

- To set a universal output to modulate the output voltage, follow these steps:
  - → The UO2 output jumper needs setting to select 'analogue' by placing the link on the left-hand pair of pins.
  - → Navigate to **M7007**, **Universal Output 2**
  - → Set **Label** to 'Voltmeter' to act as a reminder
  - → Set **Output Type** to '10V'
  - → Set **Value Low Limit** to '0', and **Output Signal Low Limit** to '0'– this states that the output should be 0V when we set the value to 0
  - → Set **Value High Limit** to '100', and **Output Signal High Limit** to '10'– this states that the output should be 10 Volts when we set the value to 100
- To control the output voltage of the universal output, follow these steps:
  - → Set **Value** to '100' to turn the output to 10 Volts – measure with a voltmeter
  - → Set **Value** to '50' to turn the output to 5 Volts

Module Information


Module Information - ObjList

Commander SN8000037
Module Information

Label	Value
Label	Tutorial M7007
Manufacturer	North
Module Type	M7007A v10
Comms Ok	Yes
Last Restart	21/09/22 15:36:44
No. of Restarts	1
Alarm Priority	2

Each module, regardless of function or type, has a module-information object, which contains information about the module's Type; when it last restarted communications; how many times it has restarted. It is also where the engineer enables the sending of communication-failure alarms.

💻 To set a module to send communications alarms, follow these steps:

→ Set **Label** to 'Tutorial M7007' – this will be used within the alarm message

→ Set **Alarm Priority** to '2' (severe) – priority 2 alarms are generally used for severe equipment failures

→ Unplug the RS232 cable from the Zip Network Card, to break communications, and wait for the alarm to appear in the Commander's alarm history.

→ Leave the RS232 cable unplugged for more than 30 seconds: modules will drive their outputs to their default states – relays will de-energise, and universal outputs will be set to 0V

→ Re-insert the RS232 cable to allow communications to restart: an alarm is sent that appears in alarm history, and outputs will return to their previous levels

View the alarms using the Commander's web pages

Note: any alarms from any inputs on this module will now have the extra label 'Tutorial M7007' inserted before the label of the input

Next Steps

In this tutorial you have learnt about North's Zip system. You have powered-on and networked together the Zip modules in the training pack. Following the steps, you have also connected to a range of input and output types.

To learn about the full range of Zip modules available and the inputs and outputs supported, refer to the *Zip Manual*.

Next, use the information you have learnt in both this tutorial and the *Commander Tutorial* to add more powerful control using ObVerse.

If you require help, contact support on 01273 694422 or visit *www.northbt.com/support*

North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

Author: TM
Checked by: JF

Document issued 22/09/2022.