Honeywell

Excel 50/500

LONWORKS® MECHANISMS

EXCEL 5000 OPEN™ SYSTEM

INTERFACE DESCRIPTION

Trademark Information

CONTENTS

REVISION OVERVIEW

The following pages have been changed from the previous issue of this document:

page	change
6	Corrected firmware version to 2.06.02 in chapter “Building Management Functionality, first paragraph.
14	Added “LON Bus ID priority handling
28	Added “remote application flashing via LON for firmware 2.06.04” in chapter “Restoring Binding Information”

EXCEL 5000 LONWORKS SYSTEM ARCHITECTURE

Firmware Version 2.04.xx

Beginning with Excel 50/500 firmware version 2.04.xx, the capabilities of the controllers belonging to the Excel 5000 family were greatly expanded:

- Excel 500 controllers, including the XC5210C (with a CPU module featuring an expanded RAM) and the Excel 500 Smart controller (XCL5010), were made capable of operating in open LONWORKS systems (see Table 3 and Table 4 on page 8 for definitions of the terms "open," "shared," and "local").
- Distributed I/O modules were made LonMark-compliant and also capable of use in LONWORKS networks independently of Excel 500 controllers.
- Excel 50 controllers (which, in LONWORKS networks, could previously operate only with configurable applications) can now also be used as freely-programmable controllers.
- Freely-programmable Excel 50/500 controllers have become capable of communicating with the Excel 10 family of controllers as well as with third-party LONWORKS devices.
- Excel 50/500 controllers now support standard LONWORKS NV's according to the LONMARK Interoperability Guidelines. Such NVs can be bound using any LONWORKS network management tool (LNS-based or non-LNS-based), and are also fully documented in the node’s self-documentation.

Building Management Functionality

Table 1 summarizes the Building Management Functionality (BMF) available under firmware version 2.04.xx via LONWORKS systems / buses (by means of either direct hardware connections or dial-up). See also section "Dial-Up Access Options" on page 53.

Table 1. Availability of BMF via LonWorks and via Dial-up (2.04.xx)

BMF	EBI	SymmetrE	Synopsis	XBS/XBSi
time programs access	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
alarms access	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
trends access	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
parameters access	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
application download	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
firmware download[2]	dial-up[1]	dial-up[1]	n.a.	dial-up[1]
bus-wide MMI	[1]	[1]	n.a.	[1]

[1] Not available via LonWorks (but available by directly-connecting a C-Bus).

[2] With the exception of the controller equipped with a modem.

Fig. 1 presents the Excel 5000 architecture under firmware version 2.04.xx.

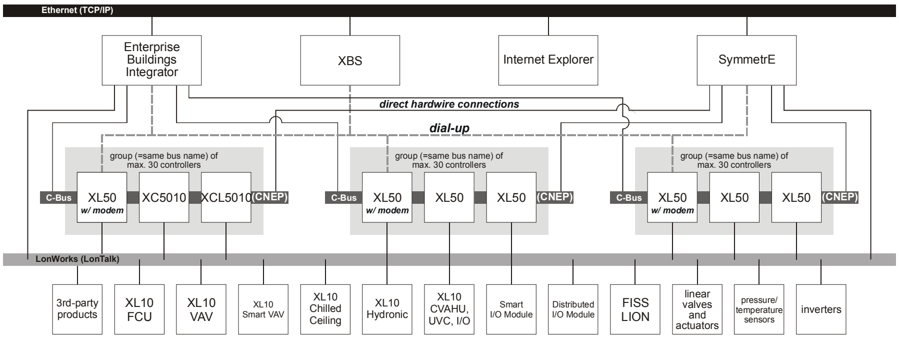


Fig. 1. Excel 5000 architecture (firmware version 2.04.xx)

Firmware Version 2.06.xx

Beginning with Excel 50/500 firmware version 2.06.xx, the capabilities of the controllers belonging to the Excel 5000 family were further expanded:

- Besides such LONWORKS network management tools (LNS-based or non-LNS-based) as LonMaker for Windows, CARE 4.xx, too, can now also be used to perform the LONWORKS binding of Honeywell and 3rd-party LONWORKS products.
- In the case of Excel 50/500 controllers, Building Management Functionality is (to varying degrees) available via direct hardware connections to LONWORKS systems / buses or via dial-up to LONWORKS systems / buses. See following section.

Building Management Functionality

Table 2 summarizes the Building Management Functionality (BMF) available under firmware version 2.06.xx via LONWORKS systems / buses (by means of either direct hardware connections or dial-up). See also section "Dial-Up Access Options" on page 53.

Table 2. Availability of BMF via LonWorks and via Dial-up (2.06.xx or higher)

BMF	EBI	SymmetrE	Synopsis	XBS/XBSi
time programs access	direct+dial-up	direct+dial-up	direct	dial-up, only
alarms access	direct+dial-up	direct+dial-up	direct	dial-up, only
trends access	direct+dial-up	direct+dial-up	direct	dial-up, only
parameters access	direct+dial-up	direct+dial-up	direct	dial-up, only
application download	direct+dial-up	direct+dial-up	direct	dial-up, only
firmware download*	dial-up, only	dial-up only	n.a.	dial-up, only
bus-wide MMI	direct, only	direct, only	direct, only	n.a.
* With the exception of the controller equipped with a modem.				

Fig. 2 presents the Excel 5000 architecture under firmware version 2.06.xx.

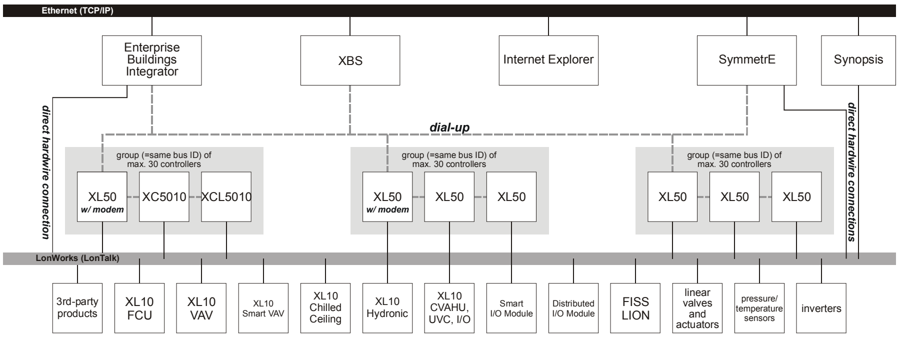


Fig. 2. Excel 5000 architecture (firmware version 2.06.xx)

COMPATIBILITY OF EXCEL 50/500 CONTROLLERS AND DISTRIBUTED I/O MODULES

IMPORTANT:

Only those Excel 50/500 controllers bearing the LonMark logo feature fully open LonWorks functionality (due to RAM limitations of the 3120B1 Neuron chip, open LonWorks functionality is not supported in earlier models [date code: 0044 or earlier] of Excel 50/500 controllers containing that chip).

CARE will not support the engineering of a LonWorks network interface if the user selects a controller or an application module containing any LonWorks chip other than the 3120E5. The controller will reject the download of application files containing the LonWorks network interface (.ymp; *.ynv; *.ycv) if any LonWorks chip other than 3120E5 is detected (even if the download source is XBS / XI584 / CARE). In this case, an alarm message "HW Config. Failure" is also generated.*

LonMark identification

LonMark-approved devices are identified with the following logos:

Fig. 3. LonMark logos

Devices which comply with the LonMark™ Interoperability guidelines but for which there is no LonMark™ Object profile are identified with the LonMark™ symbol shown above without the name "LonMark".

The XC5010C bears the LonMark symbol on the front label. The XCL5010, XD50-FL, and XD50-FCL feature the LonMark symbol on the warning label of the application module. The XFL52xB modules bear the LonMark logo on the type sticker. In addition, all of the aforementioned hardware features the LonMark symbol on the unit package label.

Table 3 provides an overview of the LonWorks functionality of the Excel 500 and freely programmable Excel 50 controllers.

NOTE: Configurable LonWorks applications for the Excel 50 (applications requiring a particular application module [e.g. XD50-FL-AH03-EN] and configured by Lizard) are LonMark-compliant, so this feature is affected by neither the new V2.04.xx, V2.05.xx, or V2.06.xx firmware nor the new 3120E5 Neuron chip. Upgrading either the firmware or the Neuron chip (via an enhanced application module) is possible but not necessary.

Table 3. Controller compatibility (non-LonMark CPUs/application modules, date code later than week 44 in 2000)

Controller type	Controller firmware	Open LonWorks functionality	CPU autobinding[1] with		CARE 4.0 LonWorks binding	LM4W binding
			XFL52x	XFL52xB		
XC5010C, XCL5010	2.00.xx – 2.03.xx	not possible	local	local	not possible	not possible
	2.04.xx	not possible	local	local/shared	not possible	not possible
	2.06.xx	not possible	local	local/shared	not possible	not possible
XD50-FL, XD50-FCL	2.04.xx – 2.06.xx	not possible	not possible	not possible	not possible	not possible
XD50-FL-xxxx-yy[2], XD50-FCL-xxxx-yy[2]	2.00.xx – 2.05.xx	possible	not possible	not possible	possible	possible
	2.06.xx	possible	not possible	not possible	possible	possible

[1] The term "local" refers to an operating mode in which a maximum of 16 modules are assigned (automatically) to each controller and only a single controller is connected to each LonWorks bus.
The term "shared" means that a maximum of 16 modules are assigned (manually) to each controller, but that multiple controllers can be connected to a single LonWorks bus.
The term "open" refers to an open LonWorks system, i.e. the use of CARE to generate a LonMark-compliant external interface file (XIF) capable of providing NVs which can be bound to other devices (Excel 50 or Excel 10 controllers, third-party devices); further, the limitation of max. 16 modules per controller can also be exceeded.
See also section "Determining the Operating Mode of a New LonWorks System" on page 40 for more-detailed information on these operating modes.

[2] "xxxx-yy" stands for configurable applications, e.g. AH03-EN.

Table 4. Controller compatibility (LonMark CPUs/application modules, date code later than week 44 in 2000)

Controller type	Controller firmware	Open LonWorks functionality	CPU autobinding[1] with		CARE 4.0 LonWorks binding	LM4W binding
			XFL52x	XFL52xB		
XC5010C, XCL5210C, XCL5010	2.00.xx – 2.03.xx	not possible	local	local	not possible	not possible
	2.04.xx	in use	not possible	shared/open	not possible	possible
	2.04.xx	not in use	local	local/shared	not possible	possible
	2.06.xx	in use	not possible	not possible	possible	possible
	2.06.xx	not in use	local	local/shared	not possible	not possible
XD50-FL, XD50-FCL	2.04.xx – 2.05.xx	in use	not possible	open	not possible	possible
	2.04.xx – 2.05.xx	not in use	not possible	not possible	not possible	not possible
	2.06.xx	in use	not possible	not possible	possible	possible
	2.06.xx	not in use	not possible	not possible	not possible	not possible
XD50-FL-xxxx-yy[2], XD50-FCL-xxxx-yy[2]	2.00.xx – 2.05.xx	in use	not possible	not possible	not possible	possible
	2.00.xx – 2.05.xx	not in use	not possible	not possible	not possible	not possible
	2.06.xx	in use	not possible	not possible	possible	possible
	2.06.xx	not in use	not possible	not possible	not possible	not possible

[1] The term "local" refers to an operating mode in which a maximum of 16 modules are assigned (automatically) to each controller and only a single controller is connected to each LonWorks bus.
The term "shared" means that a maximum of 16 modules are assigned (manually) to each controller, but that multiple controllers can be connected to a single LonWorks bus.
The term "open" refers to an open LonWorks system, i.e. the use of CARE to generate a LonMark-compliant external interface file (XIF) capable of providing NVs which can be bound to other devices (Excel 50 or Excel 10 controllers, third-party devices); further, the limitation of max. 16 modules per controller can also be exceeded.
See also section "Determining the Operating Mode of a New LonWorks System" on page 40 for more-detailed information on these operating modes.

[2] "xxxx-yy" stands for configurable applications.

Table 5. Distributed I/O module compatibility

Distributed I/O modules	LonWorks Functionality, by XL500 controller firmware version		
	V2.00.xx to V2.03.xx	V2.04.xx	V2.06.xx
XFL521, XFL522A, XFL523, XFL524A	One controller to which Distributed I/O modules are assigned on a single LonWorks bus; operating mode: local	One controller to which Distributed I/O modules are assigned on a single LonWorks bus; operating mode: local	One controller to which Distributed I/O modules are assigned on a single LonWorks bus; operating mode: local
XFL521B, XFL522B, XFL523B, XFL524B	One controller to which Distributed I/O modules are assigned on a single LonWorks bus (if you wish to enable this backwards-compatible mode[1] for the XFL52xB modules, press the LonWorks service pin while turning HEX switch); operating mode: local	Full LonWorks functionality: Multiple Distributed I/O modules and multiple controllers[2] possible on a single LonWorks bus; operating mode: open	Full LonWorks functionality: Multiple Distributed I/O modules and multiple controllers[2] possible on a single LonWorks bus; operating mode: open

[1] To cancel the backwards-compatible mode for XFL52xB modules (date code: 0044 or later), thus allowing full LonWorks functionality, press and hold down the LonWorks service pin for at least 3 seconds.

[2] Excel 500 controller with Neuron 3120E5 chip required!

NOTE: The compatibility of XFR522A and XFR524A Manual Override modules is affected by neither the firmware version nor the Neuron chip version.

FREELY PROGRAMMABLE LONWORKS EXCEL 50/500 CONTROLLERS

Number of NVs supported

Excel 50: The Excel 50's network interface can contain up to 46 NVs (in addition to the Node Object's NVs).

The Excel 50 will reject applications having more than 46 NVs. In this case, the following system alarm will be issued:

Alarm number: 61; alarm text: “Too many Globals”

Excel 500: The Excel 500's network interface can contain up to 512 NVs (in addition to the Node Object's NVs).

The Excel 500 supports 128 physical data-points (I/Os) and 256 pseudo data-points. Every data-point can be mapped to an input NV, or to an output NV, or to both. The data-points can be mapped to a maximum of 512 LONWORKS NVs. A theoretical maximum of 381 physical data-points (I/Os) are supported via NVs.

Memory Requirements

The memory requirements (in bytes) can be calculated by adding together the memory requirements attributable to the following individual items:

Default texts (ASPECD, descriptors, alarm texts, engineering units, status texts): The maximum memory allotted to default texts is **21,780** bytes

On-line changes to the time program All annual programs are automatically erased by the controller when they turn more than one year old. The maximum memory allotted to on-line changes to the time program is **4,096** bytes.

The time program, itself The first switching point requires 12 bytes, and each additional switching point with the same switching time requires another 6 bytes. A time interval with an exception day program in the annual program requires 9 bytes. A holiday with an exception day program in the holiday list requires 2 bytes. A today exception for a point requires 17 bytes. A day program requires 21 bytes X no. of switching points X 12. Thus, the total time program requires 35 bytes + (size, in bytes, of all day programs) + (no. of today exceptions X 17 bytes) + (no. of time intervals with an exception X 9 bytes) + (no. of holidays with an exception X 2 bytes).

The CARE application program Assuming one time program, five control loops, five switching tables, CARE (including RAL, RAP, RAT, RAZ) will require **10,000** bytes. Assuming several time programs, ten control loops, ten switching tables, CARE will require **20,000** bytes.

NOTE: The use of complex ModAL XFMs can further boost that portion of the memory requirements attributable to the control loops.

Data-points Each data-point requires **67** bytes.

NVs Memory requirements depend upon the number of structural components (n) of each NV. The maximum memory allotted for all NVs is **2,048** bytes. The NV-dependent memory requirements can be calculated as follows:
(number of NVs) X (3 bytes + (n X 3 bytes))
For example: Given 100 NVs with four structural components per NV:
100 X (3 bytes + (4 X 3 bytes)) = 1500 bytes

Many-to-one (M-T-O) bindings The memory requirements of many-to-one bindings depends upon the number of NVs and the number of structural components (n) per NV. The maximum memory requirements due to all many-to-one bindings = **12,800** bytes. The actual memory requirements depend upon whether the NVs are analog or digital, and can be calculated as follows:
no. of analog NVs X (9 bytes + no. of analog M-T-O X (4 bytes + 4 bytes)) +
no. of digital NVs X (9 bytes + no. of digital M-T-O X (4 bytes + 1 byte)
Example: Given ten analog NVs and five digital NVs, each with 20 M-T-O,
10 X (9 bytes + 20 X (4 bytes + 4 bytes) +
5 X (9 bytes + 20 X (4 bytes + 1 byte) = 1690 + 545 = 2235
The maximum memory allotted for all many-to-one bindings is **12,800** bytes.

The remote trend buffer

Each trend entry requires **47** bytes. Centrals A, B, and C can each contain a maximum of 100 trend entries. Further, unused application memory (if any) can be allotted to Central A, thus enabling it to contain more than 100 trend entries. The number of additional trend entries which central A can contain =
(128 Kbytes – application size in Kbytes) X 1,024 bytes / 47 bytes

Overall size constraints

If the memory requirements amount to less than 110,000 bytes, then the size is OK.

If the memory requirements amount to between 110,000 and 128,000 bytes AND the RAL is greater than 18 Kbytes (in which case RAL will run from the flash memory), then the size is OK.

If the memory requirements amount to between 110,000 and 128,000 bytes AND the RAL is less than 18 Kbytes AND the rest of the application is less than 110,000 bytes, then the size is OK.

If the memory requirements amount to between 110,000 and 128,000 bytes AND the RAL is less than 18 Kbytes AND the rest of the application is greater than 110,000 bytes, then the application is too large and must be reduced.

If the memory requirements exceed 128,000 bytes, then the application size must be reduced (e.g. by lowering the complexity of the application by reducing the number of or simplifying control loops).

Extending the Number of Physical I/Os using NVs

It is possible to extend the number of physical I/Os to more than 22 (in the case of the Excel 50) and to more than 128 (in the case of the Excel 500). This is done by mapping pseudo data-points to NVs, followed by binding NVs to physical I/Os on the LonWorks network.

On the average, two NVs will be needed for each physical I/O. This will typically allow for

- 23 additional physical I/Os for the Excel 50 and for
- 62 additional physical I/Os for the Excel 500.

Excel 500

Specifically, in the case of the Excel 500 Controller, this will allow for

- 128 I/Os via autobinding (128 physical data-points)
- 62 I/Os via NV-binding (124 pseudo data-points)

This will leave 132 pseudo data-points which can be mapped to the remaining 388 NVs for application interoperation purposes (communication with other devices on the LONWORKS network).

Node Object NVs

Fig. 4 presents controller Node Object NVs, categorized according to whether they are mandatory or optional and listing optional configuration properties.

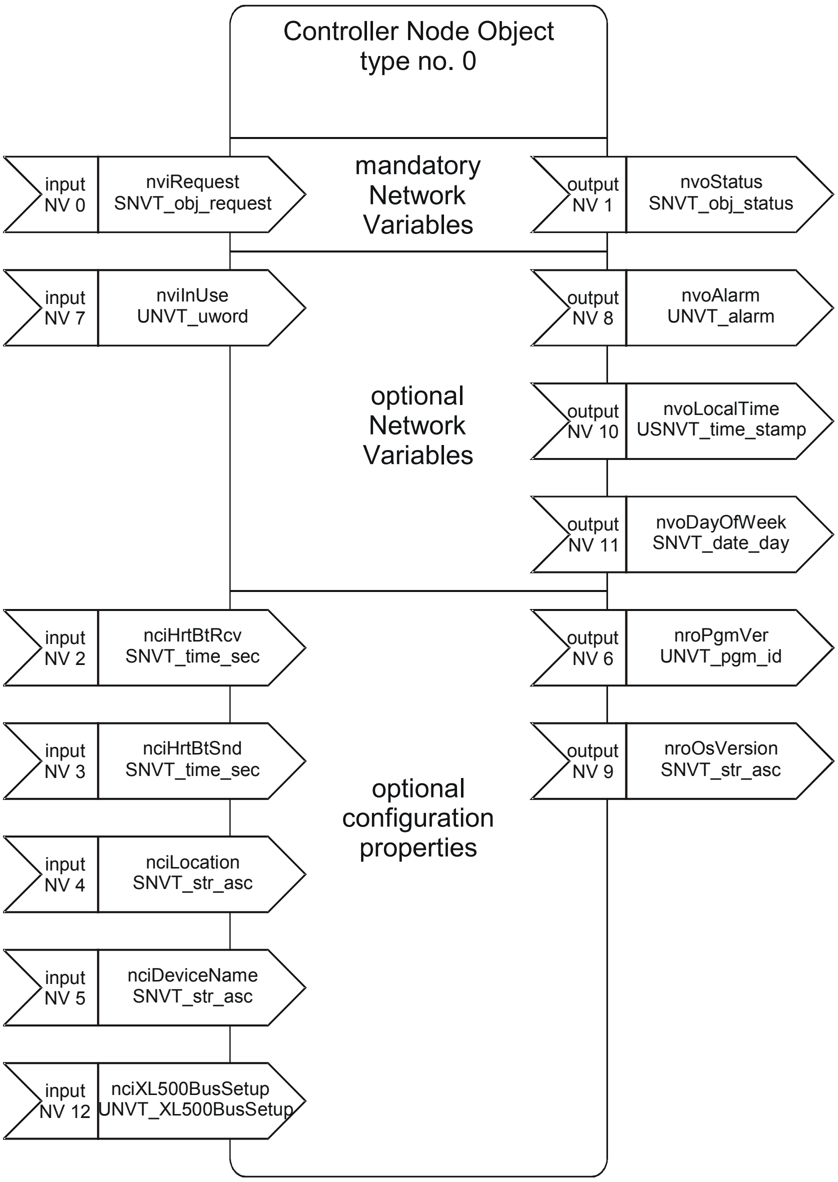


Fig. 4. Controller node object NVs

Table 6 presents information on the LONWORKS Node Object NVs in each LONWORKS Excel 50/500 controller.

Table 6. Node Object NVs (left)

NV Index	NV	Field	NV Type	States / engineering units + range	Value
0	nviRequest	object_id	SNVT_obj_request		
	nviRequest	object_request	SNVT_obj_request	RQ_NORMAL RQ_DISABLED RQ_UPDATE_STATUS RQ_SELF_TEST RQ_UPDATE_ALARM RQ_REPORT_MASK	0 1 2 3 4 5
1	nvoStatus		SNVT_obj_status		
2	nciHrtBtRcv		SCPTmaxRcvTime (SNVT_time_sec)	10 to 150 seconds	Initialized by CARE (90s default)
3	nciHrtBtSnd		SCPTmaxSendTime (SNVT_time_sec)	10 to 150 seconds	Initialized by CARE (60s default)
4	nciLocation		SCPTlocation (SNVT_str_asc)		empty string
5	nciDeviceName		SCPToemType (SNVT_str_asc)		initialized from controller name in application (unless changed by a LONWORKS tool)
6	nroPgmVer	id	UNVT_pgm_id		Initialized by CARE
	nroPgmVer	major_ver	UNVT_pgm_id		0
	nroPgmVer	minor_ver	UNVT_pgm_id		0
	nroPgmVer	bug_ver	UNVT_pgm_id		0
	nroPgmVer	node_type	UNVT_pgm_id		0
7	nviInUse		UNVT_uword		initialized to 65,535 at start-up
8	nvoAlarm		UNVT_alarm	NO_ALARM ALARM_WARM_BOOT ALARM_RACL_ERROR ALARM_POWER_FAIL ALARM_APPLICATION_STOPPED RETURN_TO_NORMAL	0 1 2 3 4 128_U
9	nroOsVersion		SNVT_str_ascii		
10	nvoLocalTime	year	SNVT_time_stamp	0 to 3000	
	nvoLocalTime	month	SNVT_time_stamp	0 to 12	
	nvoLocalTime	day	SNVT_time_stamp	0 to 31	
	nvoLocalTime	hour	SNVT_time_stamp	0 to 23	
	nvoLocalTime	minute	SNVT_time_stamp	0 to 59	
	nvoLocalTime	second	SNVT_time_stamp	0 to 59	
11	nvoDayOfWeek		SNVT_date_day	DAY_SUN DAY_MON DAY_TUE DAY_WED DAY_THU DAY_FRI DAY_SAT DAY_NUL	0 1 2 3 4 5 6 0xFF
12	nciXL500BusSetup	message code	BYTE	0 to 99	0x4D
	nciXL500BusSetup	bus ID	BYTE		0xFF
	nciXL500BusSetup	controller no.	BYTE		0xFF
	nciXL500BusSetup	heartbeat	BYTE		0x14 seconds
	nciXL500BusSetup	domain	BYTE		0x00
	nciXL500BusSetup	dom. ID length	BYTE		0xFF / unused
	nciXL500BusSetup	dom. ID value	BYTE		0

Table 3. Node Object NVs (right)

Comments
This input NV belongs to the Node Object and provides the mechanism for requesting a particular object within a node.
See above, but for object status.
This output NV reports the status of the controller upon request through nviRequest. If bound, a change of data will be sent. If not bound, the data is updated internally, only, and can be polled.
This is the configuration property used to control the maximum time that elapses after the last update to certain NVs before these input NVs adopt their default values. For each input NV, CARE can be used to set whether an NV is to be checked according to nciHrtBtRcv, and all mapped NVs are checked in this way by default. If the controller does not receive an update within the specified interval, it will try to poll the output NV from the source device. In this way, the heartbeat mechanism also works for nodes without periodic updates. If the polling also fails, the data-point mapped to this NV is set to NO RESPONSE and the invalid value (if specified) is adopted. If the value of nciHrtBtRcv is changed using a LONWORKS network management tool, and if the application is saved to Flash memory, the value is saved as well.
This is the configuration property used to control the maximum time that expires before the controller automatically sends the current values of certain NVs, even if its value did not change. This provides a heartbeat output NV that can be used by the destination objects to ensure that the node is still healthy. CARE can set whether an NV is to be sent periodically, and all mapped NVs are sent in this way by default. If the value of nciHrtBtSnd is changed using a LONWORKS network management tool, and if the application is saved to Flash memory, the value is saved as well.
Contains an empty string of 31 bytes that can be used to store installation location information.
Controller or application module name (up to 18 characters). If it is changed by a LONWORKS network management tool, the name in the application changes too, and if the application is saved to Flash memory, the name is saved as well.
Program version ID identifies the LONWORKS application running in each controller (unique for each controller). Also called network interface program ID.
not used
not used
not used
not used
This is initialized to 0xFFFF at start-up. It is then set by an engineering tool or other supervisory node that is “logged-on” to the node to prevent concurrent access by two such devices. When the updating of a node's configuration is finished, the initial value is restored. This input NV is stored in RAM and is lost after a restart.
When bound, this output NV will be sent whenever a system or application alarm condition occurs. Alarm values 5 through 127 are for application alarms, where data-point alarms are mapped to nvoAlarm using CARE (TRUE for alarm status and FALSE for OK status, with application alarm number assigned). The alarm is added to RETURN_TO_NORMAL (128) when the alarm condition is no longer true.
Contains the controller firmware version number.
The current local time of the controller, updated every minute (seconds field always equals 0). When the real-time clock of the controller is changed, it may take up to one minute until that change is reflected in the output NV.
See above.
See above.
See above.
See above.
See above.
Updated with the current day enumerator, typically at midnight. When the real-time clock of the controller is changed, it may take up to one minute until that change is reflected in the output NV.
The message code can be changed in the event of trouble with 3rd-party devices.
The bus ID distinguishes between virtual C-buses (0x00 = physical C-bus active, 0xFF = initialized by controller).
The controller number is the same as the C-bus controller number (0x00 = communication disabled, 0xFF = initialized by controller).
The heartbeat is the time between wink messages.
The domain is a flag (0 = 1st domain or 1 = 2nd domain).
The domain ID length is valid only for the 2nd domain.
The domain ID value is valid only for the 2nd domain.

Activating and Configuring LonWorks plus BMF

With XL50/500 firmware 2.06.xx and higher, the node object will support the configuration property nciXL500BusSetup.

This configuration property defines whether the XL50/500 supports

- C-bus and/or standard LonWorks or
- LonWorks plus Building Management Functionality (which employs the complete LonTalk protocol, including telegrams of the type "explicit message").

In the event that you opt for LonWorks plus BMF, a maximum of 25 global points can be used. The purpose of this restriction is to limit the traffic load on the LonWorks bus by avoiding frequent message updating and broadcast messaging in the Excel 50/500 controllers.

In the event that you need to use more than 25 global points, additional global points can be obtained by means of NV mapping.

CARE 4.01.03 and higher

When creating an application using CARE 4.01.03 and higher, you will be called upon to define each individual bus (i.e. group of max. 30 controllers with the same bus ID) as communicating either by means of C-bus and standard LonWorks on one hand or by means of LonWorks plus BMF on the other. This is done in CARE by clicking the (automatically generated) name of each individual bus appearing beneath "Bus", going to the "Properties" pane, and selecting the desired radio button: Select either "C-Bus" for C-bus and standard LonWorks or "LON-Bus" for LonWorks plus BMF, as appropriate. If you choose "LON-Bus", CARE will then automatically do the following two things:

- ensure that the LonWorks bus contains not more than 30 controllers;
- ensure that no two buses in the same LonWorks network have the same bus ID.

Lizard

If you have created a configurable Excel 50 application using Lizard, you must define the bus ID using either one of the following two procedures:

- setting the bus ID during the MMI's start-up sequence or
- using EBI, Excelon or any LON tool to edit the ”bus ID” byte in the configuration property nciXL500BusSetup.

NOTE: Downloading an application created using CARE 4.01.03 will overwrite the results of any such procedure.

Table 7. nciXL500BusSetup

Attribute	type	default value	remark
Message Code	BYTE	0x4D	changeable in case of trouble with 3rd party devices
bus ID	BYTE	0xFF	distinguish virtual C-buses 0x00 = physical C-bus active 0xFF = initialized by controller
controller number	BYTE	0xFF	same as C-bus Controller no. 0x00 = communication disabled 0xFF = initialized by controller
heartbeat	BYTE	0x14 sec	time between wink messages
domain	BYTE	0x00	flag, 0=1st domain or 1=2nd domain
domain ID length	BYTE	0xFF / unused	only for 2nd domain
domain ID value	BYTE[6]	0	only for 2nd domain

LON Bus ID priority handling

XL50/500 firmware 2.06.00 through 2.06.03: The LON Bus ID setting residing in the controller has priority over and will override any LON Bus ID setting set using CARE.

XL50/500 firmware 2.06.04: The LON Bus ID setting set using CARE has priority over and will override any LON Bus ID setting residing in the controller.

Standard NVs

Table 8 list the standard LONWORKS NVs (SNVTs) supported for mapping. From controller firmware version 2.06.xx onwards, NVs having a SNVT index of 132 and higher are also supported.

Table 8. Supported SNVTs

SNVT Index	Name	Units/description	Resolution	Invalid value
1	SNVT_amp	amps	0.1 A	0x7FFF
2	SNVT_amp_mil[1]	milliamps	0.1 mA	0x7FFF
3	SNVT_angle	rads	0.01 rad	0xFFFF
4	SNVT_angle_vel	rads per sec	0.1 rad/sec	0x7FFF
5	SNVT_btu_kilo	thousands of BTU's	1 KBTU	0xFFFF
6	SNVT_btu_mega	millions of BTU's	1 MBTU	0xFFFF
7	SNVT_char_ascii	ASCII characters	1 character	—
8	SNVT_count	event count	1	0xFFFF
9	SNVT_count_inc	event count	1	0x7FFF
11	SNVT_date_day	day names	1	0xFF
13	SNVT_elec_kwh	kilowatt-hours	1 kWh	0xFFFF
14	SNVT_elec_whr	watt-hours	0.1 Wh	0xFFFF
15	SNVT_flow	liters per second	1 l/sec	0xFFFF
16	SNVT_flow_mil	milliliters per second	1 ml/sec	0xFFFF
17	SNVT_length	meters	0.1 m	0xFFFF
18	SNVT_length_kilo	kilometers	0.1 km	0xFFFF
19	SNVT_length_micr	microns	0.1 microns	0xFFFF
20	SNVT_length_mil	millimeters	0.1 mm	0xFFFF
21	SNVT_lev_cont	percentage	0.5 %	0xFF
22	SNVT_lev_disc	level names	1	0xFF
23	SNVT_mass	grams	0.1 g	0xFFFF
24	SNVT_mass_kilo	kilograms	0.1 kg	0xFFFF
25	SNVT_mass_mega	tons	0.1 ton	0xFFFF
26	SNVT_mass_mil	milligrams	0.1 mg	0xFFFF
27	SNVT_power	watts	0.1 W	0xFFFF
28	SNVT_power_kilo	kilowatts	0.1 kW	0xFFFF
29	SNVT_ppm	parts per million	1 ppm	0xFFFF
30	SNVT_press	kilopascals	0.1 kPa	0x7FFF
31	SNVT_res	ohms	0.1 Ohm	0xFFFF
32	SNVT_res_kilo	kilo-ohms	0.1 kOhm	0xFFFF
33	SNVT_sound_db	decibels	0.01 dB	0x7FFF
34	SNVT_speed	meters / second	0.1 m/s	0xFFFF
35	SNVT_speed_mil	millimeters / second	0.001 m/s	0xFFFF
36	SNVT_str_asc	used in Node Object		—
38	SNVT_telcom	telecomm state names	1	0xFF
39	SNVT_temp	degrees Celsius	0.1 °C	0xFFFF
41	SNVT_vol	liters	0.1 liter	0xFFFF
42	SNVT_vol_kilo	kiloliters	0.1 kl	0xFFFF
43	SNVT_vol_mil	milliliters	0.1 ml	0xFFFF
44	SNVT_volt	volts	0.1 V	0x7FFF
45	SNVT_volt_dbmv	decibels*millivolts	0.1 db mV	0x7FFF
46	SNVT_volt_kilo	kilovolts	0.1 kV	0x7FFF
47	SNVT_volt_mil	millivolts	0.1 mV	0x7FFF
48	SNVT_amp_f	amps	1 A	0x7F000000
49	SNVT_angle_f	rads	1 rad	0x7F000000
50	SNVT_angle_vel_f	rads per second	1 rad/sec	0x7F000000
51	SNVT_count_f	dimensionless	1	0x7F000000
52	SNVT_count_inc_f	dimensionless	1	0x7F000000
53	SNVT_flow_f	liters per second	float value	0x7F000000
54	SNVT_length_f	meters	1 m	0x7F000000
55	SNVT_lev_cont_f	percentage	1%	0x7F000000
56	SNVT_mass_f	grams	1	0x7F000000
57	SNVT_power_f	watts	float value	0x7F000000
58	SNVT_ppm_f	parts per million	float value	0x7F000000
59	SNVT_press_f	Pascals	float value	0x7F000000

SNVT Index	Name	Units/description	Resolution	Invalid value
60	SNVT_res_f	ohms	float value	0x7F000000
61	SNVT_sound_db_f	decibels	float value	0x7F000000
62	SNVT_speed_f	meters per second	float value	0x7F000000
63	SNVT_temp_f	degrees Celsius	float value	0x7F000000
64	SNVT_time_f	seconds	float value	0x7F000000
65	SNVT_vol_f	liters	float value	0x7F000000
66	SNVT_volt_f	volts	float value	0x7F000000
67	SNVT_btu_f	British Thermal Units	float value	0x7F000000
68	SNVT_elec_whr_f	watt-hours	float value	0x7F000000
69	SNVT_config_src	dimensionless (defining self-installation)		0xFF
70	SNVT_color	structured		—
71	SNVT_grammage	grams / sq. meter	0.1 g/m^2	0xFFFF
72	SNVT_grammage_f	grams / sq. meter	float value	0x7F000000
75	SNVT_freq_f	Hertz	float value	0x7F000000
76	SNVT_freq_hz	Hertz	0.1 Hz	0xFFFF
77	SNVT_freq_kilohz	kilohertz	0.1 kHz	0xFFFF
78	SNVT_freq_milhz	megahertz	0.1 mHz	0xFFFF
79	SNVT_lux	lux	1 lux	0xFFFF
81	SNVT_lev_percent	% of full-scale / ppm	0.005	0x7FFF
82	SNVT_multiplier	--	--	0xFFFF
84	SNVT_time_stamp	structured		Per NV field
85	SNVT_zerospan	structured		Per NV field
87	SNVT_elapsed_tm	structured		Per NV field
91	SNVT_muldiv	structured		Per NV field
92	SNVT_obj_request	structured		—
93	SNVT_obj_status	structured		—
95	SNVT_switch	structured, but mapped with a single data-point		0xXXFF
97	SNVT_override	dimensionless	1	0xFF
98	SNVT_pwr_fact	multiplier	0.00005	0x7FFF
99	SNVT_per_fact_f	multiplier	float value	0x7F000000
100	SNVT_density	kilograms / meter^3	0.5 kg/m^3	0xFFFF
101	SNVT_density_f	kilograms / meter^3	float value	0x7F000000
102	SNVT_rpm	revolutions / minute	1	0xFFFF
103	SNVT_hvac_emerg	emergency mode names	1	0xFF
104	SNVT_angle_deg	degrees of arc	0.02 deg	0x7FFF
105	SNVT_temp_p	degrees Celsius	0.01 °C	0x7FFF
106	SNVT_temp_setpt	structured		Per NV field
107	SNVT_time_sec	seconds	0.1 sec	0xFFFF
108	SNVT_hvac_mode	HVAC mode names	1	0xFF
109	SNVT_occupancy	occupancy mode names	1	0xFF
110	SNVT_area	square millimeters	200 mm^2	0xFFFF
111	SNVT_hvac_overid	structured		0xFF
112	SNVT_hvac_status	structured		0xFF
113	SNVT_angle_deg	degrees of arc	0.02 deg	0x7FFF
115	SNVT_scene	structured		Per NV field
116	SNVT_scene_cfg	structured		Per NV field
117	SNVT_setting	structured		Per NV field
118	SNVT_evap_state	evaporation states	1	0xFF
119	SNVT_therm_mode	thermostat mode names	1	0xFF
120	SNVT_defr_mode	defrost mode names	1	0xFF
121	SNVT_defr_term	defrost termination names	1	0xFF
122	SNVT_defr_state	defrost state names	1	0xFF
123	SNVT_time_min	minutes	1	0xFFFF
124	SNVT_time_hour	hours	1	0xFFFF
125	SNVT_ph	acidity	0.001 pH	0x7FFF
126	SNVT_ph_f	acidity (pH)	float value	0x7F000000
128	SNVT_tod_event	for inputs, only current state is read	1	0xFFFFFFFF

SNVT Index	Name	Units/description	Resolution	Invalid value
129	SNVT_smo_obscur	percentage smoke obscuration	0.001% (0 to 5%)	0xFFFF
130	SNVT_fire_test	fire initiator types		---
131	SNVT_temp_ror	° Celsius / min	0.5 °C/min	0x7FFF
132	SNVT_fire_init	type names	1	FN_NUL
133	SNVT_fire_indcte	type names	1	FI_NUL
134				
135	SNVT_earth_pos	direction of latitude, structured		---
136				
137*	SNVT_reg_val_ts	register value	bit fields	---
138	SNVT_volt_ac	volts (a.c.)	1 V	0xFFFF
139	SNVT_amp_ac	amperage (a.c.)	1 A	0xFFFF
140				
141				
142				
143	SNVT_turbidity	Nephelometric Turbidity Units	0.001 NTU	0xFFFF
144	SNVT_turbidity_f	Nephelometric Turbidity Units	float value	---
145				
146	SNVT_elec_kwh_l	kilowatt-hours	1_kWh	0x7FFFFFFF
147	SNVT_temp_diff_p	degrees C	0.01 °C	0x7FFF
148				
149				
150				
151				
152	SNVT_pos_ctrl	structured		----
153	SNVT_enthalpy	kilojoules/kg	0.01 kJ/kg	0x7FFF
154	SNVT_gfci_status	names of ground fault circuit interrupter	1	GFCI_NUL
155	SNVT_motor_state	names of mechanical motor	1	MOTOR_NUL
156	SNVT_pumpset_mn	status of main features of pumpset	1	---
157				
158	SNVT_pumpset_sn	sensor readings of mechanical pumpset	1	---
159	SNVT_pump_sensor	sensor readings of mechanical vacuum pumps	1	---
160				
161	SNVT_flow_p	cubic meters / hour	0.01 m^3/h	0xFFFF
162				
163	SNVT_valve_mode	valve mode names	1	VALVE_NUL
164				
165	SNVT_state_64	64 bits	single bits	??

*The XL50/500 supports a special version of SNVT_reg_val_ts (SNVT index: 137) called SNVT_reg_val_ts_XL500, which provides separate bytes instead of the value of the bit fields. Via a special XFM (available from CARE) these bytes can be separated into bit values.

[1] The Excel 50/500 controller always converts units to the basic unit, i.e. milliamps to amps. As a consequence, the SNVT_amp_mil will show amps on the datapoint side (mapped datapoint).

Network Interface Program ID

The program ID of the network interface is provided in the XIF. The XIF itself is determined by the application programmed in CARE. Along with the application translation, CARE will create the XIF automatically.

During CARE engineering; the program ID can be changed by changing its last byte in the data-point editor. The default setting of the last byte of the program ID represents the controller number.

External Network Interface File (XIF)

CARE 3.0 or higher will create a XIF (External Interface File) for each application engineered. This file contains the interface description required by a LONWORKS network management tool for installation and binding.

The following file formats are provided:

1. One file in ASCII format for import into LonMaker for Windows.
2. One file in binary format for download into the controller's Neuron chip for later upload from the controller's Neuron chip by LonMaker for Windows, when the controller is online and the application is running.

A device template for commissioning LONWORKS controllers can be created either by importing the XIF or by uploading online from the controller.

The maximum number of different XIF files per LONWORKS network is 255.

Binding and Mapping NVs

Binding

Binding is the process by which NVs from different nodes are connected to each other for passing data on the LONWORKS bus. This is achieved, online with the controller, using a LONWORKS network management tool (LNS-based or non-LNS-based) such as LonMaker for Windows. Typically, the output NV of one node is bound to the input NV of another node. Fig. 5 depicts the binding of an output NV with an input NV.

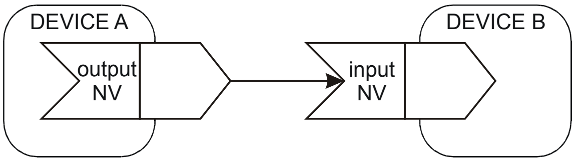


Fig. 5. NV binding

Mapping

Mapping is the process by which Honeywell Excel 5000 controller data-points are connected internally with NVs. This is performed during CARE engineering. Fig. 6 depicts the mapping of an input NV with the data-point of an Excel 50/500 controller.

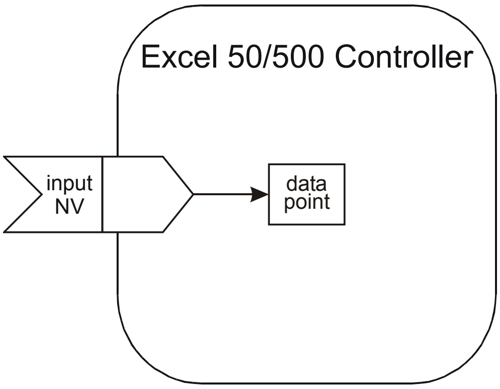


Fig. 6. Data-point mapping

Binding Options

In addition to the simple one-to-one binding relationship shown in Fig. 5, LONWORKS technology also allows the following other binding relationships to be established.

Many-to-one binding

For example, it is allowed to bind several output NVs to a single input NV. This is referred to as a many-to-one relationship (see Fig. 7).

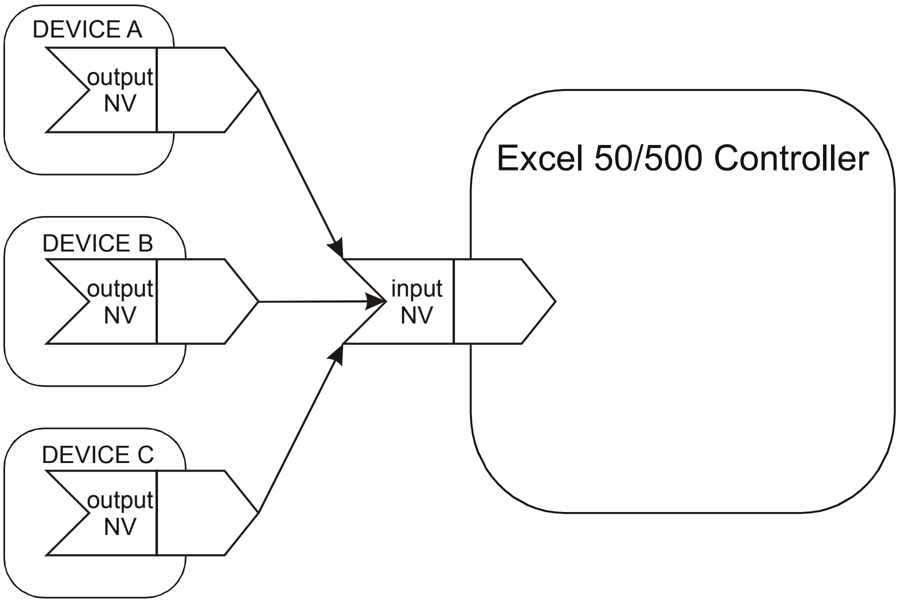


Fig. 7. Many-to-one binding (allowed)

NOTE: A maximum of 64 output NVs can be bound to a single input NV.

IMPORTANT:

Although the many-to-one binding is created online with a LONWORKS network management tool, for proper memory allocation, the binding must be specified during CARE engineering. If the user tries to make more many-to-one bindings than he/she specified during CARE engineering (a maximum of 64 many-to-one bindings are allowed), an MTO BINDING FAILED alarm will be issued.

One-to-many binding

It is also allowed to bind a single output NV to multiple input NVs of other LONWORKS devices (so-called "one-to-many" binding) as long as no two of these multiple input NVs are bound to the **same** LONWORKS device (see Fig. 8).

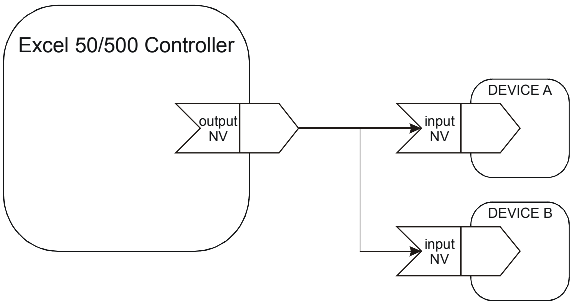


Fig. 8. One-to-many binding (supported)

Turnaround binding

Turnaround binding, in which an output NV is bound to an input NV of the same device (see also Fig. 9), is **not** supported.

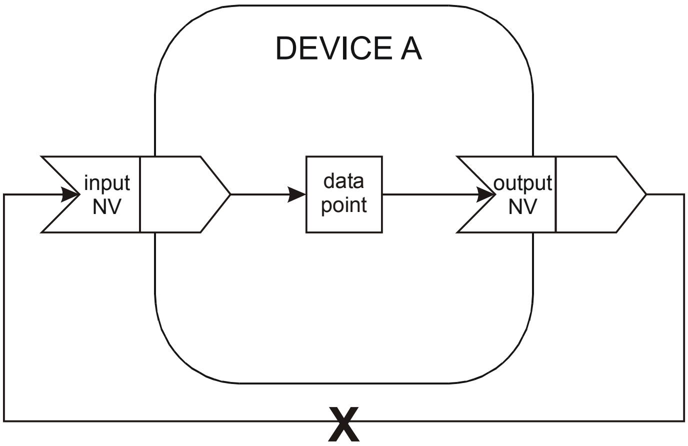


Fig. 9. Turnaround binding (not supported)

Alias binding

Alias binding (in which both a copy **and** the original of a node's output variable are bound to input variables of another node; see Fig. 10) is **not** supported. However, such copies (which have the same data as the original output variable, but which may have their own address table entries and selectors) can be bound to input variables of differing nodes (equivalent to making "one-to-many" bindings).

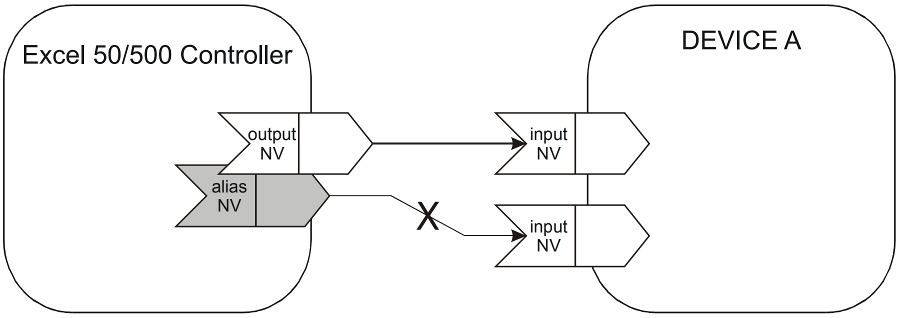


Fig. 10. Alias binding (not supported)

Workaround for alias binding

The following workaround can be used in place of alias binding. During CARE engineering, it is possible (using the IDT control icon) to establish a logical relationship between one data-point (called the *trigger* data-point; data-point A in Fig. 11) and other data-points (called *triggered* data-points; data-points 1, 2, and 3 in Fig. 11). After being connected in this fashion, when the trigger data-point is updated, it will automatically switch (i.e. trigger) the triggered data-points.

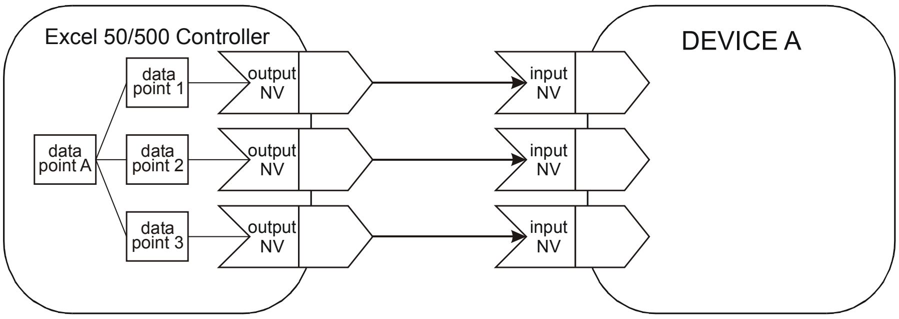


Fig. 11. Workaround for alias binding (supported)

Mapping Options

It is allowed to map a single input NV with multiple data-points (see Fig. 12).

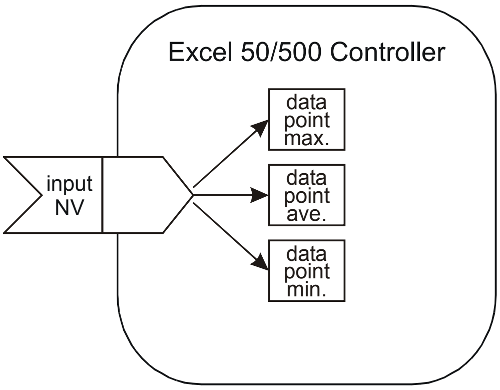


Fig. 12. Mapping a single NV with multiple data-points (allowed)

It is also allowed to map a single data-point with both a single input and a single output NV (so-called "double-mapping"; see Fig. 13). This feature also makes it possible to convert NVs of one type into NVs of another type.

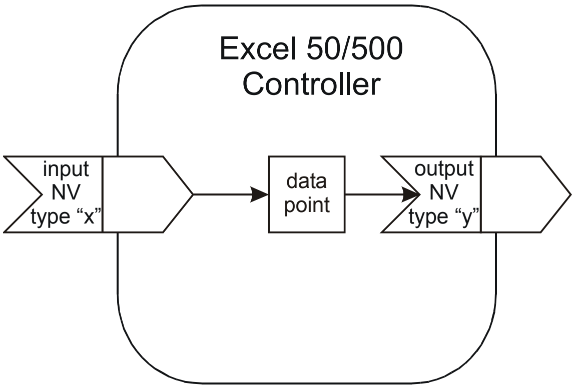


Fig. 13. Double-mapping a data-point (allowed)

It is *not* allowed to map a single data-point with multiple input or output NVs (see Fig. 14).

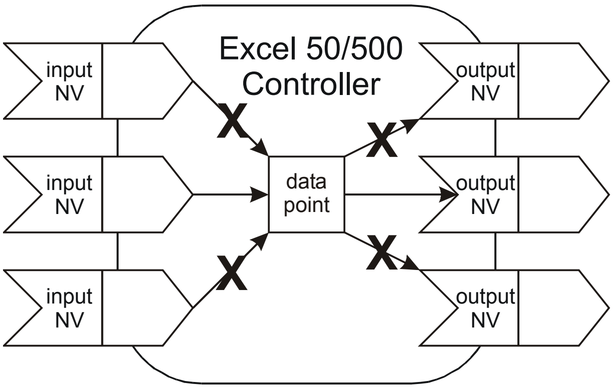


Fig. 14. Mapping a single data-point with multiple NVs (not allowed)

It is allowed to map multiple data-points with multiple fields of a structured NV (see Fig. 15).

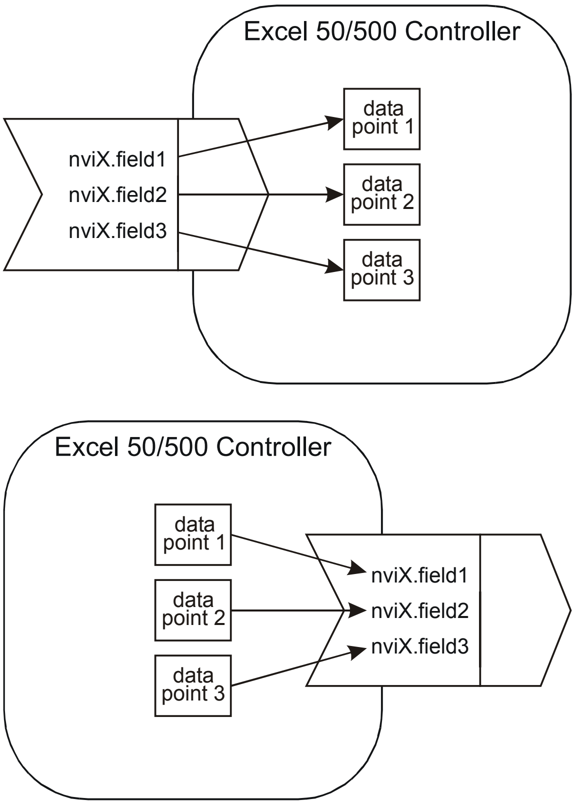


Fig. 15. Mapping multiple data-points with different fields of a structured NV (allowed)

It is *not* allowed to map multiple data-points with a single field of a structured NV (see Fig. 16).

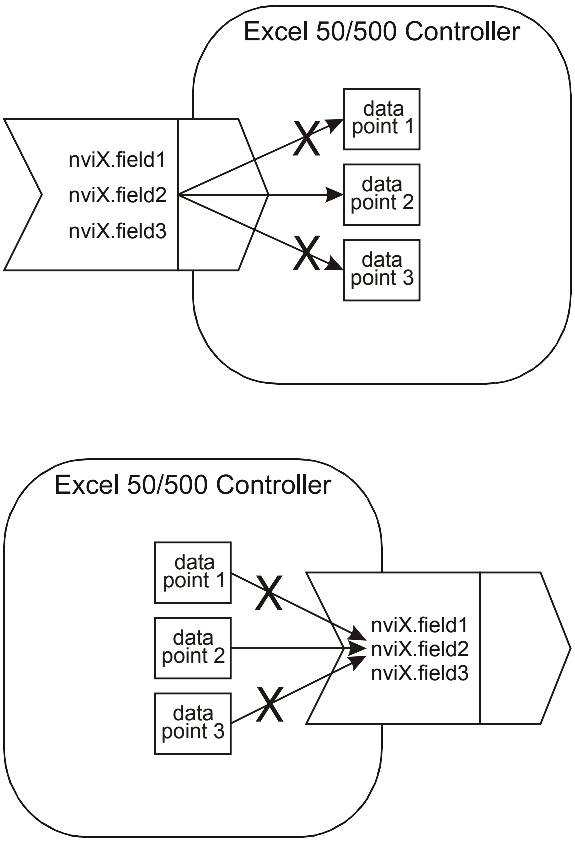


Fig. 16. Mapping multiple data-points with a single field of a structured NV (not allowed)

The name and the index of mapped NVs can be displayed on an MMI like data-point attributes. If no NVs are mapped, blanks will be displayed in the index field and in the name field.

CARE 4.0, Controller Firmware Version 2.06.xx

Automatic manual override mapping

With CARE 4.0 and controller firmware version 2.06.xx or higher, there is an additional type of data-point mapping called "automatic manual override mapping". In automatic manual override mapping, both output control and manual override feedback are effected via a single data-point.

Automatic manual override mapping functions only with NVs of the type "SNVT_Switch". Further, it is supported only for:

- XL500, firmware version 2.04.xx / CARE 3.xx and
- XL50/500, firmware version 2.06.xx / CARE 4.xx
- Those Distributed I/O modules and Smart I/O modules equipped with both digital outputs and manual overrides (i.e. the XFL524B, as well as the XFC2D and XFC3D).

With automatic manual mapping, during CARE engineering, you map the "value" attribute of one of the controller's data-points to one of the controller's output NVs. During CARE autobinding, the following steps are then automatically performed:

- CARE binds the aforementioned output NV to an input NV (previously specified by the user during CARE engineering) of one of the I/O modules assigned to the controller.
- CARE generates a new input NV for the controller and binds it to the output NV of the I/O module.

If the module's manual override switch is now operated, that output NV of the module containing information about the status of the I/O module's manual override switch is updated and sent to the new input NV. The new input NV activates the data-point's "manual value" attribute.

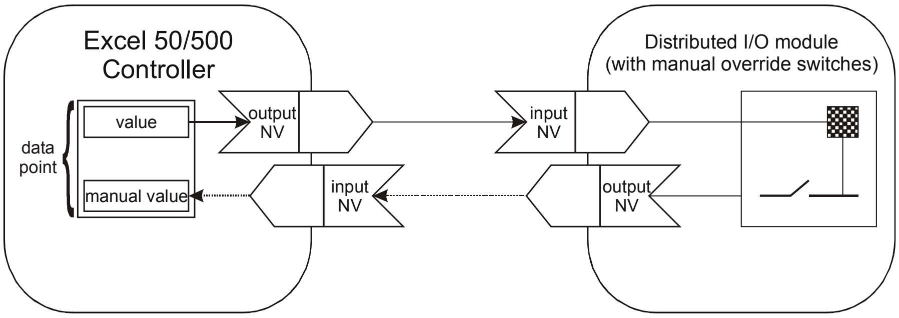


Fig. 17. Automatic manual override mapping

Automatic auto/manual mapping

With CARE 4.0 and controller firmware version 2.06.xx and higher, there is an additional type of data-point mapping called "automatic auto/manual mapping." It can be used to enable an NV to be overridden and the corresponding value to be displayed – all via a single data-point. If the data-point's "auto/manual flag" attribute has been set (by using the MMI to put the data-point into the "manual" mode) to the "manual" value, the data-point will be in the "manual" mode, and the "manual value" attribute's value is sent; if its "auto/manual flag" attribute has been set to the "auto" value, the data-point will be in the "auto" mode, and the invalid value is sent.

This mapping is activated during CARE engineering by selecting "auto" for the invalid match for the output NV of the Excel 50/500 controller.

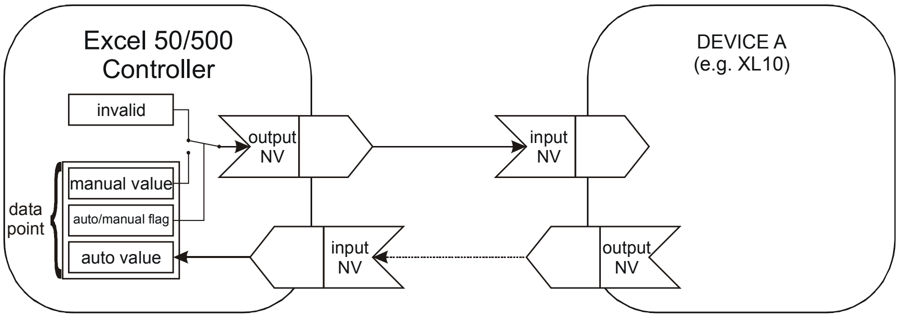


Fig. 18. Automatic auto/manual mapping

Data Priority of NVs and Data-Points

This section describes the priority between the NVs and the data-points in the application. The value of a valid input NV always has priority over the value from a sensor or switch wired directly to the controller or the values from the internal control algorithm or time program.

When an input NV is not bound and its value is invalid, then the value is ignored and the value from the local sensor/switch, the control algorithm, or the time program is written to the data-point.

When an input NV is not bound and has valid data, then the data is written to the mapped data-point. The data-point will then be in the "manual" mode, and values from the internal algorithm, time program, or local sensor/switch are ignored.

When an input NV is bound and its value is invalid, then this value is ignored and the value from the local sensor/switch, the control algorithm, or the time program is written to the data-point.

When an input NV is bound and has valid data, then this data is written to the mapped data-point. The data-point will then be in the "auto" mode, and values from the internal algorithm, time program, or local sensor/switch are ignored.

When an input NV is bound and reports NO RESPONSE (sending device or communication failure), then the predefined invalid value (if specified) is written to the mapped data-point. If no invalid value has been specified, the mapped data-point will retain the last value. The data-point will then be in the "auto" mode, and values from the internal algorithm, time program, or local sensor/switch are ignored.

Table 9 summarizes this information:

Table 9. Data-point updates according to status of input NV

Status of input NV	Data-point access
bound, valid value	NV value written to “auto value” attribute
bound, invalid value	internal value from local I/O, control algorithm, time program
bound, no response	no response, predefined invalid value from CARE
unbound, valid value	manual mode, NV value written to “manual value” attribute
unbound, invalid value	auto mode, internal value from local I/O, control algorithm, time program

Data-Point Types for NV Mapping

Table 10 lists the various different types of data-points for which mapping is allowed.

All hardware modules (XF52xx, XFL52xx, and XFCxx) are supported for NV mapping.

Table 10. Data-point types supported for NV mapping

Data-point type (CARE)	Subtype (CARE attribute text)
analog input	Slow AI
analog input	Fast AI
analog output	n/a
AO_3_pos	n/a
digital input	2-state DI or DO
digital output	2-state DI or DO
digital output	Pulse on DO/DI card
pseudo analog	n/a
pseudo digital	2-state
pseudo multistage	n-state

Boardless Data-Points

Under some circumstances, you may wish to create a data-point which has the characteristics of a physical data-point but which is also mapped with a LONWORKS NV.

This can be accomplished by mapping a data-point with an NV, but not assigning it to any I/O board or Distributed I/O module. As a result, the NV will have to be bound using a LONWORKS network management tool (e.g. LonMaker for Windows or CARE 4.xx).

In such a case, the board number of the data-point's technical address will be ≥64 and no BOARD MISSING alarm will be issued for that data-point. CARE lists these points as ≥91.

Conversion of Data-Points to NVs

When mapping analog points to analog SNVTs, a linear characteristic can be defined.

A nonlinear relationship can be defined by a look-up table. In the look-up table, a conversion for each discrete value or point stage is defined.

Example: A multistage digital data-point is mapped with an output NV:

Table 11. Mapping a multistage digital data-point with an output NV

Data-point value	NV value
1	state 0
2	state 0
3	state 0
4	state 1
5	state 1

NV-BOOSTER®

Honeywell controllers provide a mechanism that allows multiple instances of a single input NV to be combined into a single data-point. This can drastically reduce the number of controller NVs required. This enhancement not only saves application engineering, it can lower hardware costs in the project by helping to minimize the number of controllers required.

When selection logic is required to perform a function such as selecting maximum value, average value, or number of active devices from multiple input values, standard one-to-one binding of NVs is normally required, and an input NV is required for each output NV as shown in Fig. 19 below. Application engineering is required to add the selection logic before the desired value can be used by the application.

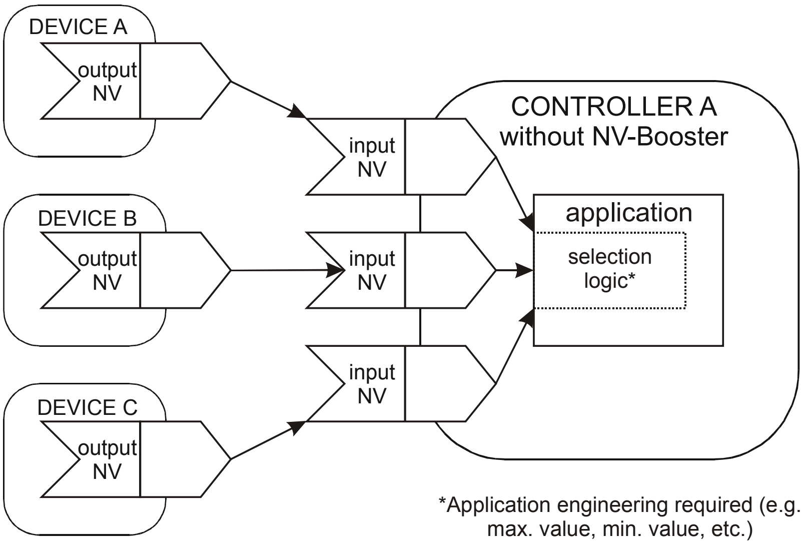


Fig. 19. Standard LONWORKS binding, without NV-Booster

However, Honeywell's NV-Booster allows binding all of the output NVs to a single input NV with a many-to-one binding, with the selection logic(s) chosen using CARE when the input NV is mapped with a single data-point. The NV-Booster® mechanism is illustrated in Fig. 20 below.

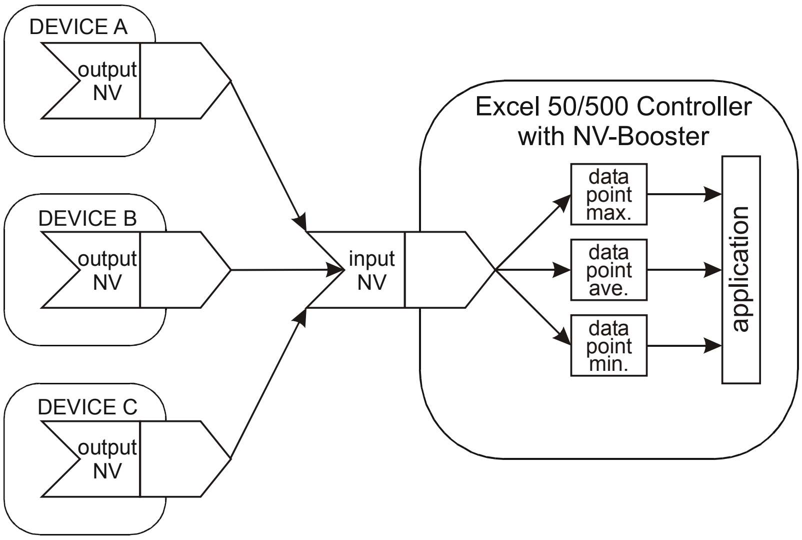


Fig. 20. Many-to-one binding, with NV-Booster

Many-to-one bindings can be established using any of the following selection logics:

- maximum value
- minimum value
- average value
- sum
- number of active devices

NOTE: The number (max.: 64) of source devices in a many-to-one binding must be specified during CARE engineering, as this is required for calculating the correct amount of controller application memory to be allocated for the binding.

In the case of many-to-one bindings, the controller's heartbeat must be activated (see section "Device Heartbeat Activation") if the NV-Booster is to function properly.

Binding alarm

If the user has tried to make more many-to-one bindings during controller runtime than were specified during CARE engineering, the following System Alarm will be issued:

Alarm number: 128; alarm text: “MTO Binding failed”

Device Heartbeat Activation

The general heartbeat of the LONWORKS node (i.e. device, e.g. XCL5010) can be activated or deactivated for every NV.

For each input NV, CARE can be used to set whether an NV is to be checked according to nciHrtBtRcv, and all mapped NVs are checked in this way by default. If the controller does not receive an update within the specified interval, it will try to poll the output NV from the source device. In this way, the heartbeat mechanism also works for nodes without periodic updates. If the polling also fails, the data-point mapped with this NV is set to NO RESPONSE and the invalid value (if specified) is adopted. Additionally, if alarming has been enabled for the data-point, a NO RESPONSE alarm will be issued.

If the value of nciHrtBtRcv is changed using a LONWORKS network management tool, and if the application is saved to Flash memory, the changed value is saved as well.

For each output NV, a heartbeat can be defined that can be used by the destination objects to ensure that the node is still healthy, and that if an NV update is lost, it will be re-sent. During CARE engineering, it is possible to set whether an NV should be sent periodically; all mapped NVs are sent in this way by default. If the value of nciHrtBtSnd is changed using a LONWORKS network management tool, and if the application is saved to Flash memory, the changed value is saved as well.

Bit-Field NVs

These NVs may have multiple fields of bit strings. These bit fields cannot be mapped with multiple data-points.

Example: SNVT_state

Type Size: 2 bytes

```
Structure:  typedef struct
{
            unsigned bit0  : 1;
            unsigned bit1  : 1;
            …
            unsigned bit15: 1;
}           SNVT_state;
```

Each bit indicates a Boolean state with the following interpretations:

Table 12. Boolean states

0	1
off	on
inactive	active
disabled	enabled
low	high
false	true
normal	alarm

This NV has up to 16 fields, each of which has a size of 1 bit:

Field 1: bit 0
Field 2: bit 1

…

If you wish to map these bit fields with data-points, the byte value (in this example: the 2-byte value) must be mapped with data-points using CARE, and the RACL split statement or a special XFM can be used to isolate single bits.

Restoring Binding Information

Excel 500, Firmware Version 2.04.xx

Restoration by means of flashing

All binding information can be saved into the flash memory together with the application data, thus allowing it to be completely restored following a reset.

Restoration using CARE

Furthermore, by using CARE to upload the application data (including the binding information), it is possible to exchange a controller and, after downloading the application into the new controller, to automatically restore all application data and binding information.

Excel 50/500, Firmware Versions 2.06.00 through 2.06.03 and CARE 4.xx

Restoration by means of flashing

The restoration of binding information and/or application data from the flash memory is not possible with firmware versions 2.06.00 through 2.06.03 and CARE 4.xx. Rather, the binding information exists only in the CARE application. As a consequence, if you wish to re-establish bindings, you will have to download the CARE application.

Excel 50/500, Firmware Version 2.06.04 and CARE 4.xx

Restoration by means of flashing

From firmware 2.06.04 onwards, the application (including the LON bindings) can be restored remotely (i.e. over the LONWORKS network) by means of flashing.

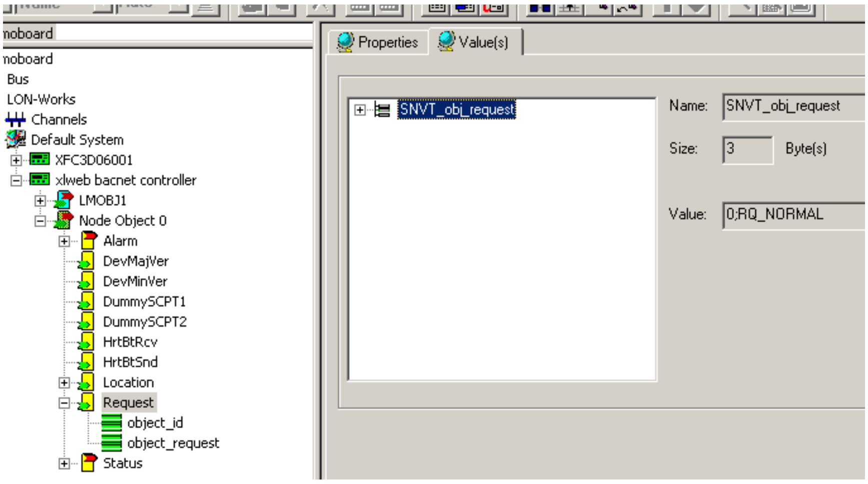
CARE 4.xx flashing

To restore the application (including the LON bindings) by means of flashing using CARE 4.xx in the online mode, proceed as follows:

1) In the node object, select nviRequest / SNVT_obj_request

2) Then set the value to “RQ_PROGRAM”

This will flash the application (including the LON bindings).



Name: SNVT_obj_request

Size: 3 Byte(s)

Value: 0;RQ_NORMAL

Verifying CARE 4.x flashing

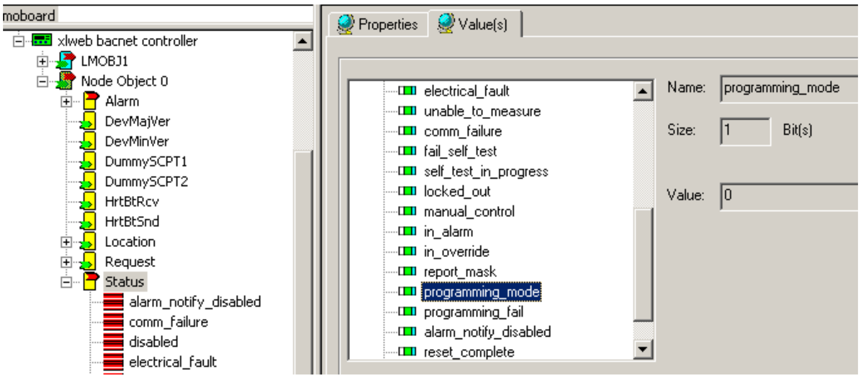
You can verify if the application has been flashed as follows:

1) In the node object, select nvoStatus / SNVT__obj_status

2) Select the bit-field “programming mode”.

Value =”1” means application has been flashed.

Value =”0” means application not flashed (e.g. due to a controller reset).



Name: programming_mode

Size: 1 Bit(s)

Value: 0

EXCELON flashing

To restore the application (including the LON bindings) by means of flashing using EXCELON in the online mode, proceed as follows:

1) In the node object, select nviRequest

2) Write "0; 15" (RQ_PROGRAM - Enable programming of special configuration properties)

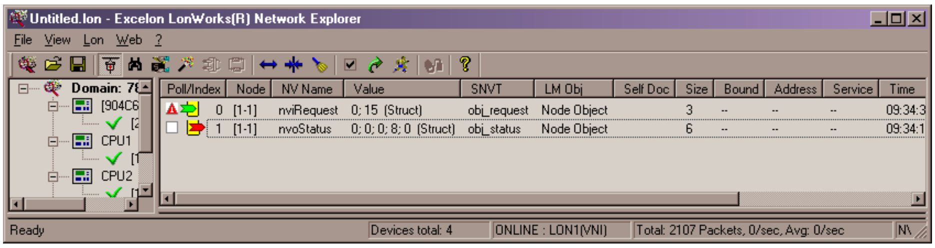
This will flash the application (including the LON bindings).

After the flashing is finished, the values will return to "0; 2" (RQ_UPDATE_STATUS - Report object status)

Verifying EXCELON flashing

You can verify if the application has been flashed as follows:

1) In the node object, select nvoStatus



Poll/Index	Node	NV Name	Value	SNVT	LM Obj	Self Doc	Size	Bound	Address	Service	Time
0	[1-1]	nviRequest	0; 15 (Struct)	obj_request	Node Object		3	--	--	--	09:34:3
1	[1-1]	nvoStatus	0; 0; 0; 8; 0 (Struct)	obj_status	Node Object		6	--	--	--	09:34:1

Ready | Devices total: 4 | ONLINE : LON1(VNI) | Total: 2107 Packets, 0/sec, Avg: 0/sec | N\

2) Check Byte 4:

-> "0; 0; 0; **0;** 0" : Application is not in flash or does not match application in RAM

-> "0; 0; 0; **8;** 0" : Application in flash matches application in RAM

-> "0; 0; 0; **4;** 0" : Flash EPROM fault (e.g. flash defect)

NOTE: After firmware download, a one-time flashing is required in order to ensure that nvoStatus shows the correct status.

System Alarms Defined for LONWORKS Applications

The following system alarms are defined for LONWORKS applications:

Table 13. Alarm descriptions

Alarm no.	Alarm text	Explanation
56	HW config. failure	1. Hardware configuration file (*.kfx file) was not completely downloaded. 2. Different modules are plugged in with the same address (set using the rotary HEX switch) in the required hardware configuration; the board number is stated. 3. Application containing NV mapping was rejected by target controller because target does not have 3120E5 Neuron chip.
61	Too many Globals	The controller rejected an application with too many NVs: Excel 50: max. 46 NVs; Excel 500: max. 512 NVs
115	NV Bindings lost	The network interface has been changed using CARE, and the changed application has been downloaded. All bindings are now lost.
129	MTO Binding failed	The user has tried to make more many-to-one bindings during controller runtime than were specified during CARE engineering.

DISTRIBUTED I/O MODULES

Handling with Excel 50/500 Firmware Version 2.04.xx

Operating Modes of Distributed I/O Modules

It is important to remember the following definitions:

Local The term "local" refers to an operating mode in which a maximum of 16 Distributed I/O modules are connected to a single host Excel 50/500 controller via a LONWORKS bus, and in which *no other* devices co-exist on that bus. In the local operating mode, the Distributed I/O modules are assigned to their host Excel 50/500 controller automatically, and autobinding is performed.

Shared The term "shared" means that, aside from the host Excel 500 controller and its Distributed I/O modules, *other devices* (which may include other Excel 500 controllers with their own Distributed I/O modules, Excel 50 or Excel 10 controllers, or third-party devices) co-exist on the LONWORKS bus. In the shared operating mode, autobinding may still be used for the NVs of a maximum of 16 Distributed I/O modules assigned (manually) exclusively to the host Excel 500 controller.

NOTE: It is recommended that you use CARE to assign the Distributed I/O modules to the host Excel 500 controller (i.e. to enter the Distributed I/O modules' Neuron IDs). The alternative is to assign them using the MMI.

Open The term "open" refers to an *interoperable* LONWORKS system in which CARE has been used to generate a LONMARK-compliant external interface file (XIF) capable of providing NVs which can be bound to other devices (which may include other Excel 500 controllers with their own Distributed I/O modules, Excel 50 or Excel 10 controllers, or third-party devices). In the open operating mode, the NVs of the Distributed I/O modules exceeding 16 must be bound manually using a LONWORKS network management tool (an LNS-based tool capable of using Honeywell plug-ins is recommended).

Combined shared and open The shared and the open operating modes can be in effect simultaneously. In this case, autobinding is performed for the NVs of a maximum of 16 Distributed I/O modules, while the data-points of additional Distributed I/O modules must be mapped with shared NVs, and the NVs of the additional Distributed I/O modules must be bound manually (e.g. using an LNS-based tool).

Autobinding (Excel 500, only)

When Distributed I/O modules are used exclusively by Honeywell Excel 500 controllers, it is possible to automatically bind their NVs to the controller. This is referred to as "autobinding." In autobinding, each controller on the bus finds the Distributed I/O modules assigned to it and binds the required NVs.

IMPORTANT:

Autobinding does not work across routers. Distributed I/O modules must be located within the same router segment as the controller to which their NVs are to be bound. However, autobinding is possible across repeaters.

IMPORTANT:

The autobound NVs of a controller are not visible to a LONWORKS network management tool, and there is hence no danger that a careless user will attempt to re-bind them. However, the NVs of the Distributed I/O modules are visible to a LONWORKS network management tool. Any attempt to re-bind the autobound NVs of Distributed I/O modules will corrupt the autobindings. In such a case, the Excel 500 controller will restore the autobindings automatically, but there will be numerous system and application alarms as a result.

If, prior to autobinding, the Distributed I/O modules have been accessed by a LONWORKS network management tool, the modules will remain in the "configured" mode. In this state, they cannot be found by the controller during autobinding – they do not appear in the list of modules on the controller MMI. Such modules must be decommissioned using the

> *LONWORKS network management tool, or the LONWORKS service pin must be pressed for at least three seconds.*

If an Excel 500 controller operating in the shared/open mode is deleted from the LonMaker project, all of its bindings will also be deleted. In this case, the Excel 500 controller will restore all of the autobindings (if any) automatically after 3 minutes (provided that no bindings are performed or changed in LonMaker in the meantime), but there will be numerous system and application alarms as a result.

Assignment (Excel 500, only)

There are two methods of assigning Distributed I/O modules to a particular Excel 500 controller.

Recommended assignment method

The Ideal approach is to know the Neuron IDs of the Distributed I/O modules at the time of CARE engineering, thus enabling you to enter the Neuron ID during CARE terminal assignment. When this is done, every module will be fully identified and assigned automatically by the Excel 500 controller after the application is downloaded.

Alternate assignment method

If the Neuron ID is not available at the time of CARE engineering, it will be possible to correctly assign the Distributed I/O modules to their controller(s) only after having downloaded the application. In this case, assignment is performed via the MMI. This procedure is described in detail in the XI581/XI582 User Guide, EN2B-126.

IMPORTANT:

> *It is essential that Distributed I/O modules not be assigned simultaneously via different MMIs. When assigning Distributed I/O modules using the alternative method, the assignments must be made on only one MMI at a time so as to avoid competing network accesses. If this is disregarded, this will result in contradictory and unreliable assignments. There will be incomplete Distributed I/O module lists displayed, and the danger exists that one controller will take away an existing assignment from another controller.*

Regardless of which of these two assignment methods is employed, assignment requires that the modules' rotary HEX switches be set according to the CARE terminal assignment

Priority of Distributed I/O Module Assignments

Assignments made via an MMI always have priority over assignments made using CARE. Thus, in the event of a conflict (e.g. when the Neuron ID entered using CARE differs from the Neuron ID entered via the MMI), the assignment carried out using the MMI will have priority.

Flashing of Distributed I/O Module Assignment

The Distributed I/O module assignment that was made during CARE engineering or via the controller MMI must be saved to Flash memory manually. When Distributed I/O module assignment has been made during the test mode, the assignments are saved in Flash memory automatically. These assignments can be reused for the application after the application has been downloaded (the MMI's assignment dialog will offer the option of keeping the existing assignment).

Controller Reset

IMPORTANT:

> *A controller reset will erase the Distributed I/O module assignment. After a reset, one of the following procedures must be performed.*

- Restore the application (including the assignments) from Flash (this is the simplest method).
- Restore the assignments during the "start-up" sequence (this requires somewhat more effort because all of the modules are searched on the LonWorks network automatically)
- Download the application and re-assign the Distributed I/O modules (this method requires the most effort because it must be done manually).

Manual Binding

There are several cases in which it is necessary to manually bind the NVs of Distributed I/O modules to their respective controller(s). This is done using a LONWORKS network management tool (e.g. LonMaker for Windows).

More than 16 modules per Excel 50/500

Autobinding can be used to bind the NVs of a maximum of 16 Distributed I/O modules per controller, only. If the application requires more than 16 Distributed I/O modules per controller, you must use CARE to allocate those additional NVs requiring mapping with data-points, and you will also have to use a LONWORKS network management tool to bind the NVs of the additional modules to the controller.

Binding of NVs of other devices to Distributed I/O modules

When the NVs of other devices on the LONWORKS bus (other than the host Excel 50/500 controller) require binding to Distributed I/O modules, autobinding cannot be used. A LONWORKS network management tool (e.g. LonMaker for Windows) is required to (manually) bind all of the Distributed I/O modules' NVs.

Double-mapping a data-point

It is possible to preserve the autobinding by mapping the data-point with a second NV. However, the second NV must then be bound (using a LONWORKS network management tool) to another LONWORKS device (see Fig. 21). While this method preserves autobinding, it does require one controller NV more than if all the binding is performed using a LONWORKS network management tool (e.g. LonMaker for Windows).

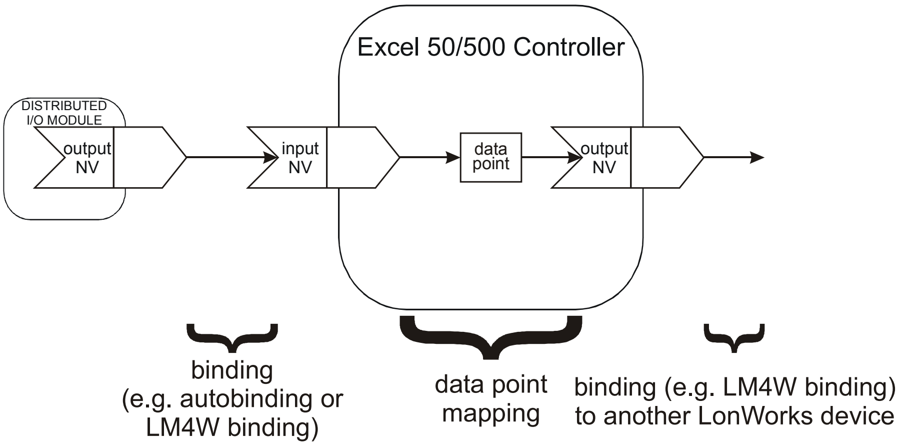


Fig. 21. Mapping with a second NV for binding to LONWORKS devices

Use of E-Vision

IMPORTANT

E-Vision cannot be used.

Handling with Excel 50/500 Firmware Version 2.06.xx and CARE 4.xx

In combination Excel 50/500 controller firmware 2.06.xx, CARE 4.xx allows Honeywell devices and/or third-party LONWORKS devices to be automatically bound without having to use LM4W or any other LONWORKS network management/binding tool.

Upgrading applications

Excel 50/500 controller firmware version 2.04.xx applications (including Excel 500 controller auto-binding) can be upgraded to firmware version 2.06.xx. This reduces the amount of LONWORKS network engineering required should you then extend the LONWORKS system (i.e. add controller[s] and Distributed I/O modules).

To upgrade applications from version 2.04.xx to version 2.06.xx, proceed as follows:

1. Upgrade the application from version 2.04.xx to version 2.06.xx.
2. Add or substitute I/O modules or third-party devices.

3. Translate the application for the Excel 50/500 controller firmware version 2.06.xx.
4. Download the Excel 50/500 controller firmware version 2.06.xx into that controller.
5. Dowload the upgraded application.

NOTE: Excel 50/500 controller firmware version 2.06.xx no longer supports auto-binding for Excel 500 controllers.

Distributed I/O Node Object NVs

All four Distributed I/O module types have the LONWORKS Node Object. The Node Object allows the function of objects within a node to be monitored. When nviRequest is updated, nvoStatus is updated. The definition of SNVT_obj_request includes an object ID field to allow the Node Object to report status conditions for all objects on a node.

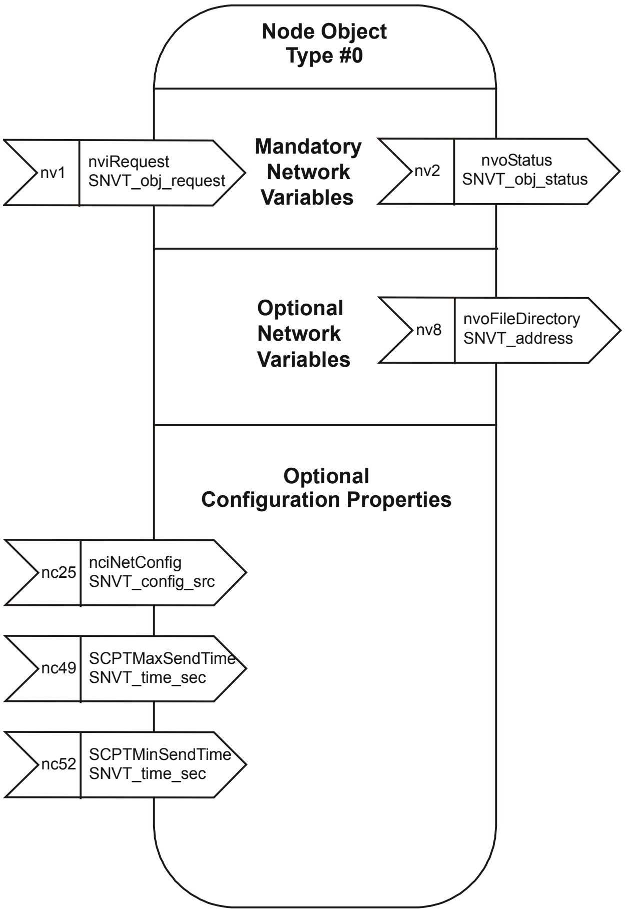


Fig. 22. Node Object for Distributed I/O modules

Summary of Distributed I/O NVs

Table 14. XFL521B Analog Input module NVs

NV Index	Direction	Name	Type	Mechanism
0	in	nviRequest	SNVT_obj_request	
1	out	nvoStatus	SNVT_obj_status	
2	in	nciNetConfig	SNVT_config_src	configuration
3	out	nvoFileDirectory	SNVT_address	
4	out	nvoAiValue[0]	SNVT_volt_f	periodic update
5	out	nvoAiValue[1]	SNVT_volt_f	periodic update
6	out	nvoAiValue[2]	SNVT_volt_f	periodic update
7	out	nvoAiValue[3]	SNVT_volt_f	periodic update
8	out	nvoAiValue[4]	SNVT_volt_f	periodic update
9	out	nvoAiValue[5]	SNVT_volt_f	periodic update
10	out	nvoAiValue[6]	SNVT_volt_f	periodic update
11	out	nvoAiValue[7]	SNVT_volt_f	periodic update
12	out	nvoAiTemp[0]	SNVT_temp_p	periodic update
13	out	nvoAiTemp[1]	SNVT_temp_p	periodic update
14	out	nvoAiTemp[2]	SNVT_temp_p	periodic update
15	out	nvoAiTemp[3]	SNVT_temp_p	periodic update
16	out	nvoAiTemp[4]	SNVT_temp_p	periodic update
17	out	nvoAiTemp[5]	SNVT_temp_p	periodic update
18	out	nvoAiTemp[6]	SNVT_temp_p	periodic update
19	out	nvoAiTemp[7]	SNVT_temp_p	periodic update
52	in	SCPTminSendTime	SNVT_time_sec	configuration
49	in	SCPTmaxSendTime	SNVT_time_sec	configuration
nci1	in	UCPTSensorConfig	none	configuration
nci2	in	UCPTSendOnDelta	SNVT_cont	configuration
nci3	in	UCPTWireOffset	SNVT_res	configuration

Table 15. XFL522B Analog Output module NVs

NV Index	Direction	Name	Type	Mechanism
0	in	nviRequest	SNVT_obj_request	
1	out	nvoStatus	SNVT_obj_status	
2	in	nciNetConfig	SNVT_config_src	configuration
3	out	nvoFileDirectory	SNVT_address	
4	in	nviValue[0]	SNVT_switch	
5	in	nviValue[1]	SNVT_switch	
6	in	nviValue[2]	SNVT_switch	
7	in	nviValue[3]	SNVT_switch	
8	in	nviValue[4]	SNVT_switch	
9	in	nviValue[5]	SNVT_switch	
10	in	nviValue[6]	SNVT_switch	
11	in	nviValue[7]	SNVT_switch	
12	out	nvoFeedback[0]	SNVT_switch	
13	out	nvoFeedback[1]	SNVT_switch	
14	out	nvoFeedback[2]	SNVT_switch	
15	out	nvoFeedback[3]	SNVT_switch	
16	out	nvoFeedback[4]	SNVT_switch	
17	out	nvoFeedback[5]	SNVT_switch	
18	out	nvoFeedback[6]	SNVT_switch	
19	out	nvoFeedback[7]	SNVT_switch	
nc1	in	UCPTSensorConfig	none	configuration
nc2	in	UCPTdriveTimeClose	SNVT_time_sec	configuration
nc3	in	UCPTdriveTimeOpen	SNVT_time_sec	configuration
nc96	in	SCPTdelayTime	SNVT_time_sec	configuration
nc88	in	SCPTminDeltaLevel	SNVT_lev_cont	configuration
nc4	in	UCPTTsyncMin	SNVT_lev_cont	configuration
nc5	in	UCPTTsyncMax	SNVT_lev_cont	configuration
nc6	in	UCPTTsyncCharge	SNVT_lev_cont	configuration

Table 16. XFL523B Digital Input module NVs

NV Index	Direction	Name	Type	Mechanism
0	in	nviRequest	SNVT_obj_request	
1	out	nvoStatus	SNVT_obj_status	
2	in	NciNetConfig	SNVT_config_src	configuration
3	out	nvoFileDirectory	SNVT_address	
4	out	nvoDiValueCnt[0]	SNVT_count	periodic update
5	out	nvoDiValueCnt[1]	SNVT_count	periodic update
6	out	nvoDiValueCnt[2]	SNVT_count	periodic update
7	out	nvoDiValueCnt[3]	SNVT_count	periodic update
8	out	nvoDiValueCnt[4]	SNVT_count	periodic update
9	out	nvoDiValueCnt[5]	SNVT_count	periodic update
10	out	nvoDiValueCnt[6]	SNVT_count	periodic update
11	out	nvoDiValueCnt[7]	SNVT_count	periodic update
12	out	nvoDiValueCnt[8]	SNVT_count	periodic update
13	out	nvoDiValueCnt[9]	SNVT_count	periodic update
14	out	nvoDiValueCnt[10]	SNVT_count	periodic update
15	out	nvoDiValueCnt[11]	SNVT_count	periodic update
16	out	NvoDiValue[0]	SNVT_switch	periodic update
17	out	NvoDiValue[1]	SNVT_switch	periodic update
18	out	nvoDiValue[2]	SNVT_switch	periodic update
19	out	nvoDiValue[3]	SNVT_switch	periodic update
20	out	nvoDiValue[4]	SNVT_switch	periodic update
21	out	nvoDiValue[5]	SNVT_switch	periodic update
22	out	nvoDiValue[6]	SNVT_switch	periodic update
23	out	nvoDiValue[7]	SNVT_switch	periodic update
24	out	nvoDiValue[8]	SNVT_switch	periodic update
25	out	nvoDiValue[9]	SNVT_switch	periodic update
26	out	nvoDiValue[10]	SNVT_switch	periodic update
27	out	nvoDiValue[11]	SNVT_switch	periodic update
52	in	SCPTminSendTime	SNVT_time_sec	configuration
49	in	SCPTmaxSendTime	SNVT_time_sec	configuration
nc1	in	UCPTSensorConfig	none	configuration
nc2	in	UCPTSendOnDelta	SNVT_cont	configuration
nc27	in	SCPTDirection	SNVT_state	configuration

Table 17. XFL524B Digital Output module NVs

NV Index	Direction	Name	Type	Mechanism
0	in	nviRequest	SNVT_obj_request	
1	out	nvoStatus	SNVT_config_src	
2	in	nciNetConfig	SNVT_config_src	configuration
3	out	nvoFileDirectory	SNVT_address	
4	in	nviValue[0]	SNVT_switch	
5	in	nviValue[1]	SNVT_switch	
6	in	nviValue[2]	SNVT_switch	
7	in	nviValue[3]	SNVT_switch	
8	in	nviValue[4]	SNVT_switch	
9	in	nviValue[5]	SNVT_switch	
10	out	nvoFeedback[0]	SNVT_switch	
11	out	nvoFeedback[1]	SNVT_switch	
12	out	nvoFeedback[2]	SNVT_switch	
13	out	nvoFeedback[3]	SNVT_switch	
14	out	nvoFeedback[4]	SNVT_switch	
15	out	nvoFeedback[5]	SNVT_switch	
16	out	nvoManCnt[0]	SNVT_count	
17	out	nvoManCnt[1]	SNVT_count	
18	out	nvoManCnt[2]	SNVT_count	
19	out	nvoManCnt[3]	SNVT_count	
20	out	nvoManCnt[4]	SNVT_count	
21	out	nvoManCnt[5]	SNVT_count	
22	out	nvoDiagnose	SNVT_count	
nc1	in	UCPTSensorConfig	none	configuration

Distributed I/O Plug-Ins

LNS plug-ins are provided for use with LonMaker for Windows for commissioning Distributed I/O modules. These plug-ins make it very easy to set the configuration properties of the Distributed I/O modules.

IMPORTANT

Distributed I/O plug-ins must be used only for manually binding modules. Do not use Distributed I/O plug-ins during or after the autobinding procedure.

GUIDELINES FOR SPECIFYING LONWORKS EXCEL 500 SYSTEMS

NOTE: The following applies only to Excel 500 controllers with firmware version 2.04.xx.

It is of enormous importance to follow the following guidelines when specifying and designing a new LONWORKS system or when specifying changes or amendments to an existing LONWORKS system. Consideration of these guidelines will avoid unnecessary re-engineering measures, it will avoid time and cost overruns, and it will avoid frustration for field engineers and customers.

Determining the Operating Mode of a New LonWorks System

If you intend to install one or more new Excel 500 systems, i.e. controller(s) and Distributed I/O modules, the following flow chart can be used to determined whether your new system(s) will be in the local, the shared, or the open operating mode (or in both the shared and the open operating modes).

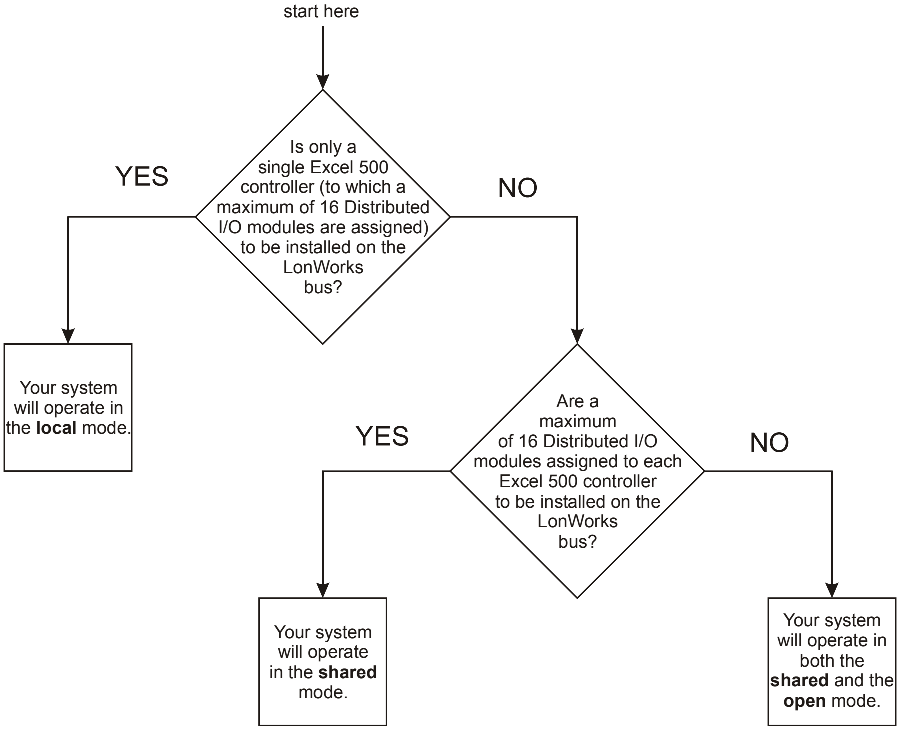

start here
Is only a single Excel 500 controller (to which a maximum of 16 Distributed I/O modules are assigned) to be installed on the LonWorks bus?
YES
NO
Your system will operate in the **local** mode.
Are a maximum of 16 Distributed I/O modules assigned to each Excel 500 controller to be installed on the LonWorks bus?
YES
NO
Your system will operate in the **shared** mode.
Your system will operate in both the **shared** and the **open** mode.

Fig. 23. Flow chart for determining the operating mode of a new LONWORKS system

Implications of Changes to an Existing LONWORKS System

Before changing an existent LONWORKS system, you must first determine its operating mode (local, shared, or open). This can be done as follows:

Local The system is in the local operating mode if the Distributed I/O module assignments are not editable using the MMI.

Shared The system is in the shared operating mode if the Distributed I/O module assignments are editable using the MMI.

Open The system is in the open operating mode if LonMaker for Windows indicates that the controller has been equipped with a network interface.

Shared and open The system is in both the shared and the open operating modes if the Distributed I/O module assignments are editable, and if LonMaker for Windows indicates that the controller has been equipped with a network interface.

Table 18. Implications of changes to an existent system operating in the local mode

Change	Implication
Replacement of Distributed I/O modules	No further action necessary
Addition of Distributed I/O modules (without exceeding 16)	No further action necessary
Addition of Distributed I/O modules (thereby exceeding 16)	Operating mode transition (from "local" to "shared / open")! Use CARE to adapt the application and to map data-points with the NVs of those Distributed I/O modules exceeding 16. Use a LonWorks network management tool for manual binding.
Addition of further devices (for which no interoperability is desired)	Operating mode transition (from "local" to "shared")! Use CARE to re-translate the application. Manually re-assign the (previously automatically assigned) Distributed I/O modules.
Addition of further devices (for which interoperability is desired)	Operating mode transition (from "local" to "shared / open")! Use CARE to adapt the mapping of all those data-points with the NVs of those Distributed I/Os for which interoperability is desired. Use a LONWORKS network management tool for manual binding.

Table 19. Implications of changes to an existent system operating in the shared and/or open mode

Change	Implication
Replacement of Distributed I/O modules	Update module assignment via the MMI.
Addition of Distributed I/O modules (without exceeding 16)	Assign the modules via the MMI.
Addition of Distributed I/O modules (thereby exceeding 16)	Use CARE to adapt the application and to map data-points with the NVs of those Distributed I/O modules exceeding 16. Use a LonWorks network management tool for manual binding.
Addition of further devices (for which no interoperability is desired)	No effect.
Interoperability of existent Distributed I/O modules desired.	Network interface change[1]. Use mapping to preserve autobinding. See also section "Manual Binding" on page 34.
Addition of further devices (for which interoperability is desired) = Network interface change[1]	Use CARE to adapt the mapping of all those data-points with NVs that are needed for interoperation. Use a LONWORKS network management tool for manual binding. Redo all LONWORKS tool bindings from and to the host Excel 50/500.

[1] The expression "network interface change" refers to any of the following actions: The addition/deletion of NVs; the changing of the NV type; the changing of an input to an output NV or vice-versa; the changing of the self-description of an NV; the changing of the NV index.

Application Changes

Certain application changes can have a significant impact on the LONWORKS network interface and consequently necessitate the performance of various engineering tasks using the LONWORKS network management tool (e.g. LonMaker for Windows).

Critical application changes

Any one of the following actions will affect the XIF (i.e. external interface file), and is therefore considered to be a critical application change:

- The deletion or addition of an NV.
- Changing an NV type.
- Changing an input NV into an output NV or vice versa.
- Changing an NV name.
- Changing an NV self description.
- Changing an NV index.

In the event of one or more critical application changes, the following steps must be taken.

1. Using a LONWORKS network management tool (e.g. LonMaker for Windows), delete the LONWORKS device of the corresponding controller. This will result in the complete deletion of all bindings.
2. Using a LONWORKS network management tool (e.g. LonMaker for Windows), delete the LONWORKS device template.
3. Download the changed application into the controller.
4. Using a LONWORKS network management tool (e.g. LonMaker for Windows), install a new LONWORKS device (controller).
5. Redo all bindings!

Non-critical application changes

None of the following actions will affect the XIF; they are therefore considered to be non-critical application changes:

- Changing data-point names.
- Changing the mapping of data-points with NVs.
- Changing time programs.
- Changing the data-point description.
- The addition, deletion, or editing of C-Bus data-points.

Download Scenarios and Impacts

The following two tables describe various possible scenarios which may be encountered when downloading applications featuring or lacking a LONWORKS external interface file (XIF) , respectively.

Table 20. Download of application featuring a LONWORKS XIF

Current state of controller	Application download
Controller has **no** application yet	Download is executed
Controller already has a **non**-LONWORKS application	Download is executed
Controller already has a LONWORKS application	CARE will alert the operator that all bindings will be lost and ask: Continue (= redo all bindings manually) or Quit (= cancel download)?

Table 21. Download of application not featuring a LONWORKS XIF

Current state of controller	Application download
Controller has **no** application yet	Download is executed
Controller already has a **non**-LONWORKS application	Download is executed
Controller already has a **different** LONWORKS application	CARE will alert the operator that all bindings will be lost and ask: Continue (= redo all bindings manually) or Quit (= cancel download)?
Controller already has the **identical** LONWORKS application	CARE will ask the operator: Keep existing bindings (this is highly recommended), loose existing bindings, or Quit?

LONWORKS SYSTEM ENGINEERING

LONWORKS Network Layout

The LONWORKS network layout can be daisy-chain, star, loop or any combination thereof as long as the maximum wire length requirements given below are met.

Doubly-terminated daisy chain

The recommended configuration is a daisy chain with two bus terminations (see Fig. 24). This layout allows for maximum length of the LONWORKS bus, and its simple structure presents the least number of possible problems, particularly when adding on to an existing bus.

Table 22. Specifications of doubly-terminated busses

Cable type	Max. bus length for segments with FTT-10 or FTT-10A transceivers, only
Belden 85102	8,900 ft (2,700 m)
Belden 8471	8,900 ft (2,700 m)
Level IV, 22AWG	4,600 ft (1,400 m)
JY (St) Y 2x2x0.8, twisted pair	3,000 ft (900 m)
TIA568A Category 5 24AWG, twisted pair	3,000 ft (900 m)

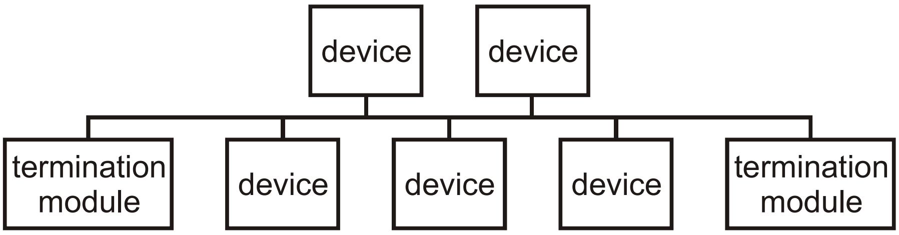


Fig. 24. Recommended configuration of doubly-terminated busses

Free topology requires only one bus termination and allows a variety of bus configurations (see Fig. 25).

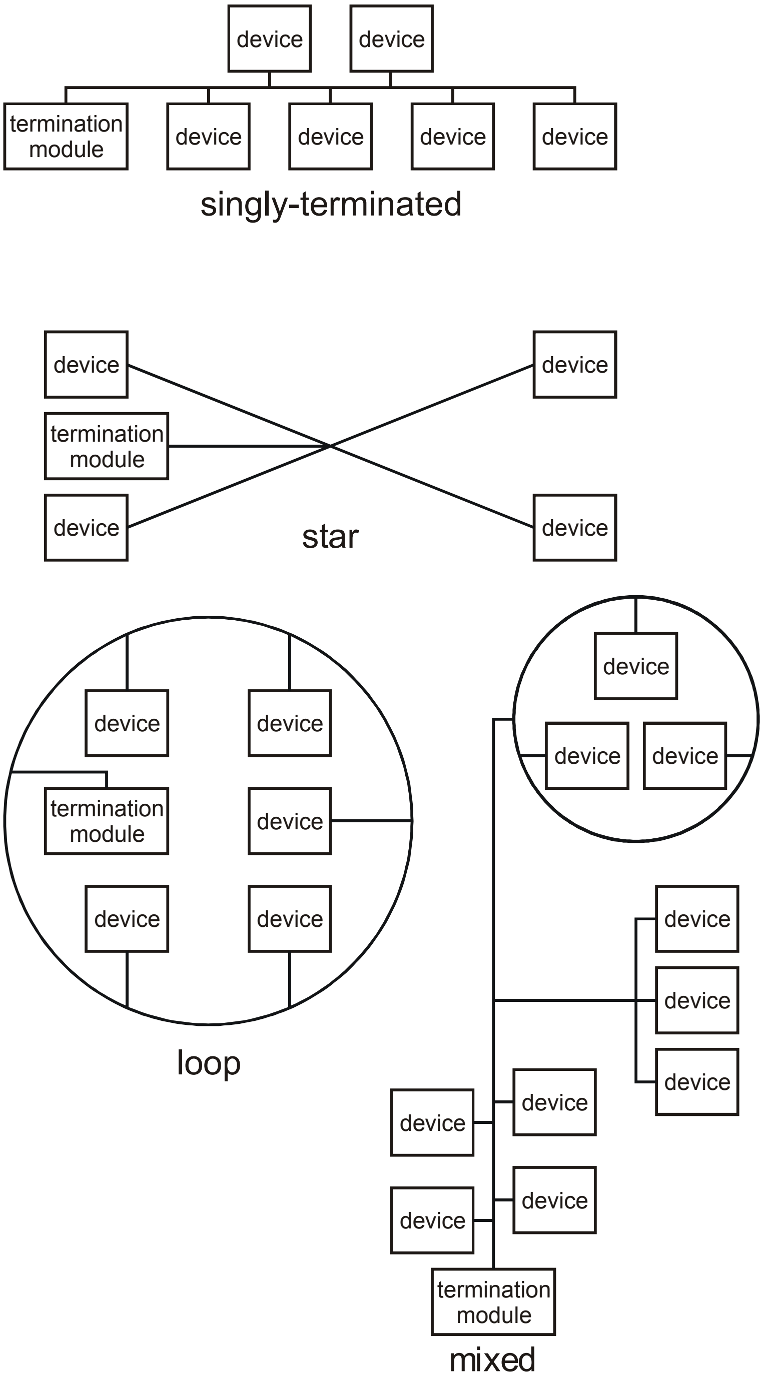


Fig. 25. Examples of free topology bus layouts

In the event that the limits on the number of transceivers or total wire distance are exceeded, then one FTT physical layer repeater can be added to interconnect two segments, thus doubling the overall system capabilities. The FTT-10A transceiver can also be used with LONWORKS routers (e.g. the LPR-10 Model 42100) to interconnect a TP/FT-10 channel with another TP/FT-10 channel, or with any other LONWORKS channel.

NOTE: The following specifications apply to a single network segment. Multiple segments may be combined using repeaters in order to increase the allowed number of nodes and distance.

System specifications

Up to 64 FTT-10A transceivers (i.e. devices equipped with FTT-10A transceivers) are allowed per network segment.

LPT-10 transceivers may be used on network segments with FTT-10A transceivers, but are subject to additional constraints (particularly on distance).

The average temperature of the wire must not exceed +55 °C, although individual segments of wire may be as hot as +85 °C.

As a general rule, the TP/FT-10 channel communication cables should be separated from high-voltage power cables. Follow local electrical codes with regards to cable placement.

Distance rules

The free topology transmission (FTT) specification includes two further requirements which must be met for proper system operation. The distance from each transceiver to all other transceivers and to the termination (including the LPT-10 termination, if used) must not exceed the *maximum node-to-node distance*. If multiple paths exists, the *maximum total wire length* is the total amount of wire used (see Table 23).

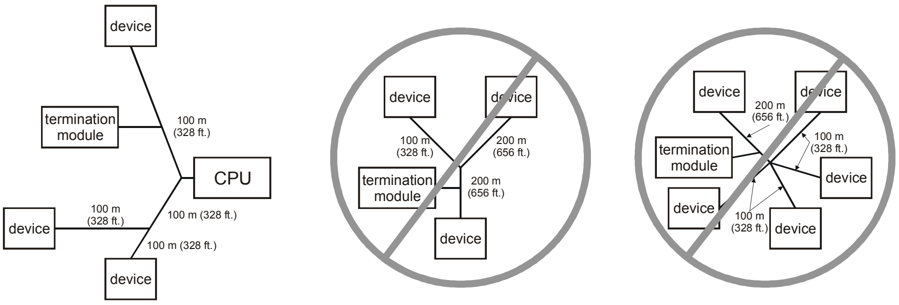
Table 23. Specifications of free topology (singly-terminated) busses

Cable type	Max. node-to-node distance	Max. total wire length
Belden 85102	1,650 ft (500 m)	1,650 ft (500 m)
Belden 8471	1,300 ft (400 m)	1,650 ft (500 m)
Level IV, 22AWG	1,300 ft (400 m)	1,650 ft (500 m)
JY (St) Y 2x2x0.8, twisted pair	1,050 ft (320 m)	1,650 ft (500 m)
TIA568A Category 5 24AWG, twisted pair	825 ft (250 m)	1,500 ft (450 m)

IMPORTANT

> *Do not use different wire types or gauges on the same segment of the LONWORKS bus. The step change in line impedance characteristics would cause unpredictable reflections on the bus.*

Examples of allowed and not-allowed free topology layouts for cable JY (St) Y 2x2x0.8 are shown in Fig. 26.



ALLOWED:
node-to-node = 200 m (656 ft.)
total wire length = 400 m (1312 ft.)

NOT ALLOWED:
node-to-node = 400 m (1312 ft.)
total wire length = 500 m (1640 ft.)

NOT ALLOWED:
node-to-node = 200 m (656 ft.)
total wire length = 600 m (1968 ft.)

Fig. 26. Example of allowed/not-allowed free topology layouts (max. node-to-node distance: 320 m, max. wire length: 500 m)

NOTE: In the event that the limit on the total wire length is exceeded, then FTT physical layer repeaters (FTT 10A) can be added to interconnect segments and increase the overall length by an amount equal to the original specification for that cable type and bus type for each repeater used. For example, adding repeaters for a doubly-terminated bus using JY (St) Y 2x2x0.8 cable increases the maximum length 900 m (3,000 ft) for each repeater.

IMPORTANT

> *The LONWORKS transceiver can be affected by electromagnetic fields generated by frequency converters. If possible, position frequency converters in a different cabinet, or allow a minimum distance of 18 in. (50 cm) between frequency converters and their respective cabling, and Distributed I/O modules.*

LONWORKS Bus Termination

One or two Termination Modules, part no. 209541B, are required for a LONWORKS Bus with FTT devices on it, depending upon the configuration. The maximum lengths described in the previous section must be adhered to for either a daisy chain or free topology layout of the LONWORKS bus. See Fig. 27 and Fig. 28 for connection details for the 2095401B Termination Module.

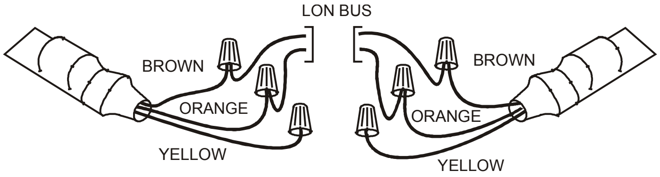


Fig. 27. Double termination

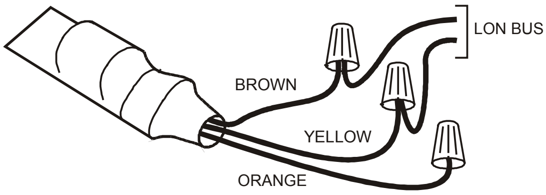


Fig. 28. Single termination

NOTE: After stripping away the cable insulation to expose the ends of the individual wires, do not untwist the wires.

NOTE: Slack cable should not be looped or coiled.

NOTE: See also section "Grounding Shielded Twisted Pair Cable" in chapter 4, "Network Cabling and Connection," of "LONWORKS FTT-10A Transceiver User's Guide."

Network Segment Load Management

In order to ensure the proper functioning of a LONWORKS network segment, the load limitations of its constituent components (controllers, Distributed I/O modules, etc.) must be taken into consideration. If necessary, measures must be taken to reduce the load contributed by the different components.

The two most important constraints are the load limitations of

- the Excel 500 controller:
  Average: 50 nv updates per second.
  At maximum: 10 nv updates per 100 ms or 100 nv updates per second
- the LONWORKS bus, itself (200 updates per second).

In addition to having their own load limitations, the various different network components also *contribute* to overall load. In section "Total Network Segment Load", the contributions of the various different components to total network segment load is examined.

In section "Minimization Measures", methods of reducing the loads contributed by the different components are explained.

Total Network Segment Load

The total network segment load can be disaggregated into three sub-loads as follows:

- The sum of the loads contributed by all of the individual devices (Honeywell and/or third-party controllers, the Distributed I/O modules assigned to them, as well as sensors and actuators) in the network segment;
- The sum of the loads contributed by network management and monitoring; and
- The load entering the network segment from adjacent network segments via routers.

In the following, the loads attributable to individual devices will be examined more closely and example calculations presented.

Load from devices

In turn, that portion of the network load of a segment attributable only to the devices is comprised of the **basic load** and the **dynamic load**. The basic load depends upon the number of (automatically or manually) bound output NVs and their heartbeats. The dynamic load is dependent upon the given application conditions and affects the frequency of changes in the values of the output NVs.

Table 24. Basic and Dynamic Loads Contributed by Different Devices

Device	Basic load	Dynamic load
Excel 500 with LONWORKS interface	Output NVs: If the heartbeat has been activated, the basic load will amount to one update per minute (default; can be changed using nciHrtBtSnd) per bound output NV	Is dependent upon the DDC cycle time as well as upon the given application conditions
XFL modules whose physical inputs/outputs have been auto-bound	Analog inputs and digital inputs: one update per minute per physical input (fixed) Analog outputs and digital outputs: two updates per minute per physical output (fixed)	Analog inputs and digital inputs: max. one update per second per physical input Analog outputs and digital outputs: max. two updates per second per physical output (given a DDC cycle time of one second; in the case of longer DDC cycle times, the max. number of updates per second per physical output is lower)
XL10 and other Honeywell devices	Is dependent solely upon the heartbeat of the bound output NVs (see device documentation).	Is dependent upon changes in the number of bound output NVs; further, it is dependent upon the given application conditions as well as upon the device's filtering capacity.
NOTE: The term "updates" includes updates of input/output NVs, polls of input/output NVs, and (in the case of the XFL modules) updates of any of the physical inputs/outputs.		

Calculating the network segment load

Fig. 29 is a schematic of an example network segment consisting of thirty XL10s and ten XFL modules, all connected to a single Excel 500 controller. However, to calculate the load, the actual number of XL10s and XFL modules is irrelevant as long as the same number of updates per second must be performed.

NOTE: In this example, it is assumed (see inset) that each XL10 assigned to the Excel 500 controller has a total of six output NVs (two of which are updated six times per minute, two of which are updated three times per minute, and two of which are updated once per minute) and that each XFL521B module assigned to the Excel 500 controller has two physical inputs (each updated once per minute) and two physical inputs (each updated twice per minute).

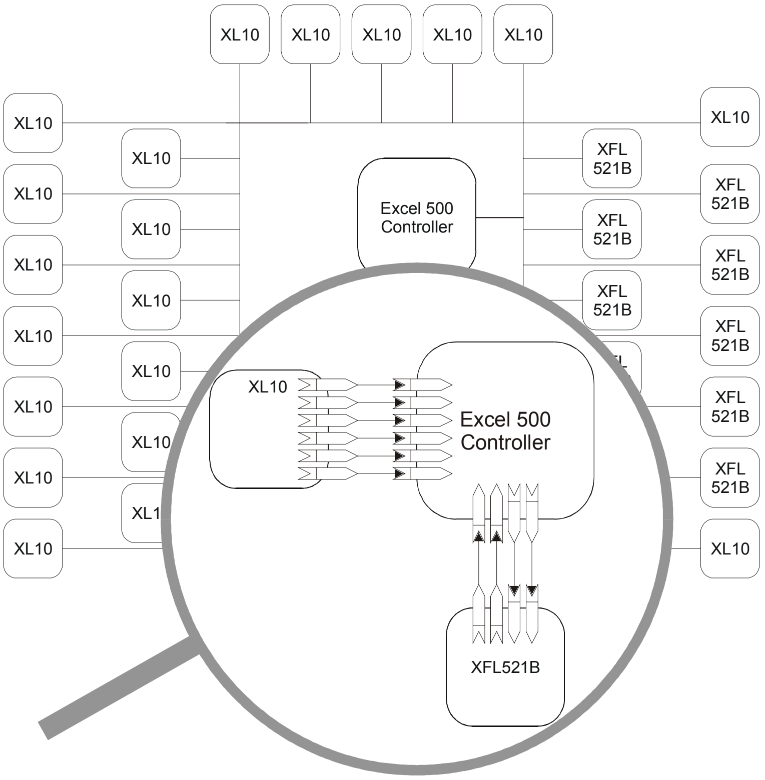


Fig. 29. Example network segment

The load which must be handled by the Excel 500 depicted in this example can be calculated as follows.

Load Attributable to the XL10s

For each XL10 assigned to the Excel 500 controller, there are two output NVs, each with six updates per minute, two output NVs, each with three updates per minute, and two output NVs, each with one update per minute. Thus, each XL10 results in 20 updates per minute. For thirty XL10s, this makes 600 updates per minute, equivalent to 10 updates per second.

Load attributable to the XFL modules

Basic Load: Each of the XFL modules' physical inputs (of which there are a total of 40) is updated once per minute. Each of the XFL modules' physical outputs (of which there are likewise a total of 40) is updated twice per minute. This results in a total of 120 updates per minute, i.e. 2 updates per second.

Dynamic Load: In this example, a dynamic load of 60 additional updates per minute, i.e. 1 update per second, has been assumed.

Net Load: The net load attributable to the XFL modules is thus the sum of the basic load plus the dynamic load, i.e. 2 updates per second plus 1 update per second = 3 updates per second.

Overall load on controller

The Excel 500 controller will thus have to handle a load of 13 updates per second (= 10 updates per second from the XL10s plus 3 updates per second from the XFL modules).

Because an Excel 500 Controller is capable of handling an average of 50 updates per second (see also section "Network Segment Load Management" on page 47), this means that, in this example, a LONWORKS network management tool (e.g. LonMaker for Windows) could be used to allocate an additional 37 poll messages per second to this particular Excel 500 Controller.

Initial load when commissioning

During the commissioning procedure, more and more controllers go on-line and begin normal operation. As a consequence, the load on the LONWORKS channel increases. If the devices going on-line send a large number of messages, the maximum allowable load may be exceeded.

General load reduction options

A general reduction in load can be achieved by lowering the number of messages sent on the LONWORKS channel by:

- increasing the hysteresis for sensors which send on delta (e.g. RIO W7761, counters, meters, etc.);
- increasing the nodes' heartbeat times.

Minimization Measures

In the following, measures aimed at reducing network segment load stemming from the various different network components are presented.

Load from Excel 50/500

The updating of the output NVs of the Excel 50/500 controllers present in the network segment is responsible for a certain amount of network segment load. This is because whenever the value of an output NV changes, this change will be sent onto the network.

In addition, it is also possible to send changes periodically. This is done using the node's heartbeat (nciHrtBtSnd).

The node's heartbeat is activated during CARE engineering and is defined online using LonMaker for Windows.

Depending upon the values chosen for the node's heartbeat, the network load will increase.

In order to reduce the number of updates for analog values, a SNVT with a lower resolution can be chosen.

In extreme cases, a hysteresis function can be created during CARE engineering. Thus, a temperature might have to change by e.g. at least 0.5 K before being sent to its mapped output NV on the network.

Load from Excel 10

The updating of the output NVs of any Excel 10 controllers present in the network segment is likewise responsible for a certain load. An additional measure is thus to adjust the update frequency via the configuration property nciHrtBtSnd (which has a default value of one minute and can be set to a minimum of one second).

Load from temperature sensors

The updating of the output NVs of temperature sensors, too, contributes to network segment load. LonMark temperature sensors follow the functional profile 1040. Within this profile, the following configuration properties can be tuned in order to reduce network load:

nc49 (nciMinSendTime)	Indicates the minimum time period between transmissions of the output NVs; default value = 5 sec
nc64 (nciMinDelta)	Indicates the minimum temperature change required to update the output NVs; default value = 0.3 °C

The following Honeywell LONMARK temperature sensors are available:

- LF20-L: Air Temperature sensor for air ducts, immersion type
- VF20-L: Water Temperature Sensor for pipes, immersion type
- AFF-L: Air Temperature Sensor, wall mount
- T7425A1005-L: Air/Water Temperature sensor for ducts/pipes, immersion type
- T7425A1013-L: Air/Water Temperature sensor for ducts/pipes, immersion type

Load from Distributed I/O Modules

The load contributed by Distributed I/O modules in the open mode can also be reduced by adjusting certain configuration properties as follows:

- **XFL523B**: The configuration property UCPTSendOnDelta specifies what difference in totalizer count value is required before the Sensor Object's value output is sent. This configuration property is stored in the configuration parameter file, and can be set using plug-ins.
- **XFL521B**: The configuration property UCPTSendOnDelta specifies what difference in raw value measured by the AD converter is required before the Sensor Object's value output is sent. This configuration property is stored in the configuration parameter file, and can be set using plug-ins.
- **XFL521B and XFL523B:** The configuration property SCPTminSendTime (Send Throttle) defines the minimum period of time (in seconds) between transmissions of the output NV. This configuration property is stored in the configuration parameter file, and can be set using plug-ins.

Load from pressure sensors

The load contributed by updates of the output NVs of pressure sensors connected to the network segment can also be reduced. LonMark-compliant pressure sensors follow the functional profile 1030. Within this profile, the following configuration properties can be tuned in order to reduce network load:

nci52 (nciMinSendTime) Indicates the minimum time period between transmissions of the output NVs; default value = 5 sec

nci27 (nciMinDelta) Indicates the minimum pressure change required to update the output NVs. Default value = 5% of the specified pressure range

The following Honeywell LONMARK pressure sensors are available:

- SN025-355-L: Low-Pressure sensor for pipes or vessels, immersion type
- SKVN250-L: Low-Pressure sensor for pipes or vessels, wall mount
- SKVN1250-L: Low-Pressure sensor for pipes or vessels, wall mount
- SKN1250-L: Low-Pressure sensor for pipes or vessels, wall mount
- SN..-355-L: High-Pressure sensor for pipes or vessels, immersion type
- FHBN..-355-L: High-Pressure sensor for pipes or vessels, immersion type or wall mount

Load from bus-wide MMI

With firmware 2.06.xx and higher, the bus-wide alarm feature should not be used when bus-wide MMI is used over LONWORKS; this is to avoid excessive LONWORKS bus traffic load.

XFL52xB and XL50/500 Response Times

The *response time of Distributed I/O modules* is defined as the period of time between the updating of the physical signal and the updating of the NV. The response time varies somewhat depending upon certain factors, including the module type. See also the Table 25.

Table 25. Response time (RT) of Distributed I/O modules

Device	Typical RT	Max. RT	Min. time between two consecutive NV updates
XFL521B (AI)	0.8 sec	1.6 sec	SCPTMinSendTime (default: 1 sec)*
XFL522B (AO)	0.2 sec	0.4 sec	n.a.
XFL523B (DI)	0.3 sec	0.5 sec	SCPTMinSendTime (default: 1 sec)
XFL524B (DO)	0.2 sec	0.4 sec	n.a.
Excel Smart I/O AI	2 sec	3 sec	SCPTMinSendTime (default: 60 sec)
Excel Smart I/O DI	1.3 sec	2 sec	SCPTMinSendTime (default: 60 sec)
*This configuration property defines the minimum period of time (in seconds) between output NV transmissions.			

In this context, the *controller response time* must also be taken into account. This is defined as the period of time between the updating of an NV and the updating of the data-point.

Table 26. Controller response time

Operating mode	Updating input NVs		Updating output NVs		Total RT (input and output NVs)	
	Typical RT	Max. RT	Typical RT	Max. RT	Typical RT	Max. RT
local	0.5 sec	1 sec	0.5 sec	1 sec	1 sec + (DDC cycle time / 2)	2 sec + DDC cycle time
open mode with DDC	0.02 sec	0.1 sec	0.02 sec	0.1 sec	0.04 sec + (DDC cycle time / 2)	0.2 sec + DDC cycle time
open mode without DDC*	0.02 sec	0.1 sec	0.02 sec	0.1 sec	0.04 sec	0.2 sec
*I.e. the data-point is mapped to either an input or an output NV						

The total system response time is defined as the sum of the response times of all of the individual devices involved. In the case of the integration of third-party products, refer to the appropriate product documentation.

New Application Opportunities

The shorter response times (for both Distributed I/O modules and controllers) possible when operating in the open mode provide access to new areas of application for the user. Given faster response times, time-critical applications can be better handled.

Example: Light Control

A data-point of an Excel 50/500 Controller can be mapped to an output NV. This output NV is then bound to an input NV of the XFL524B module, which switches the light. The output NV of a LONWORKS light switch is also bound to the same input NV of the XFL524B. By doing so, the light can be controlled not only manually via the light switch (with a response time of 400 ms), but also centrally via the time program, the controller's MMI, or a Honeywell central.

Excel 50/500 Limitations

Firmware Version-Independent Limitations

Regardless of the Excel 50/500 firmware version, various limitations apply.

Alarming Limitations

- Excel 10 alarms cannot be mapped with the NVs of Excel 50/500 controllers.
- Excel 50/500 system alarms cannot be mapped with the NVs of Excel 50/500 controllers.

Firmware Downloading Limitations

The downloading of firmware to the controller via a direct hardware connection from the front-end (e.g. EBI, SymmetrE, Synopsis, or XBS) to the LONWORKS bus is not supported. Rather, depending upon the front-end, this may be possible via dial-up.

Firmware Version-Dependent Limitations

The following limitations and advantages apply to Excel 50/500 firmware version 2.06.xx:

C-bus or LonWorks plus BMF

Under firmware version 2.06.xx and higher, communication between different Excel 50/500 controllers on the LONWORKS network is possible only when using

- both a C-bus and the standard LONWORKS bus or
- LONWORKS plus BMF.

NOTE: The C-Bus employs a CNAP protocol. Standard LONWORKS employs the LonTalk protocol (minus telegrams of the type "explicit message"). LONWORKS plus BMF employs the LONTALK protocol (including telegrams of the type "explicit message").

During CARE engineering, you will be called upon to choose between C-bus and standard LONWORKS on one hand or LONWORKS plus BMF on the other. In the event that you opt for LONWORKS plus BMF, a maximum of 25 global points can be used. In the event that you need to use more than 25 global points, additional global points can be obtained by means of NV mapping.

Bus-wide MMI via LONWORKS plus BMF

Under firmware version 2.06.xx, via LONWORKS plus BMF, you can use a bus-wide MMI to access any one of the max. 30 Excel 50/500 controllers belonging to a given group. See also section "Activating and Configuring LonWorks", sub-section "CARE 4.01.03 and higher)" on page 15.

Flexible points for internal modules

Under firmware version 2.06.xx, flexible (multistage) data-points are supported only for internal modules.

(Were Distributed I/Os and Smart I/Os to also be supported, changing LONWORKS network loads and device NV polling cycles would make it impossible to guarantee switching sequences and delays.)

“wink” functionality not supported

Excel 500, Excel 50, Distributed I/O (XFL52xB), and Smart I/O (XFCxxx) do not support the LONWORKS “wink” functionality.

BMF and net buffer settings

In order to establish LONWORKS plus BMF, it is mandatory that the net buffer settings be changed according to the grayed fields of Table 27.

Specifically, it is necessary that "net buffer in" and "net buffer out" each be allocated at least 114 bytes of storage space. In the case of standard routers (i.e. such routers as SyMik, Gesytec, etc., all of which are based on the router core module), these are ***not*** the default settings. Further, by lowering the count of "net buffer out" from 15 to 7, it is possible to allocate the necessary storage space without exceeding the total available storage capacity requirements.

The net buffer settings can be changed in the following three different ways:

- **Using CARE 4.01 or higher (this is the recommended method!)**
  When commissioning the routers, CARE 4.01 (or higher) will automatically change the net buffer settings, only asking the operator to confirm. Note: By lowering the buffer counts, CARE attempts to adjust the buffer sizes without increasing overall storage capacity requirements. If it fails to do this, a corresponding report is made. As long as all of the routers still have their default settings, success is guaranteed. However, in the event that any of their settings have been changed manually, it is possible that CARE may fail. In this case, it is necessary to use Excelon (or NodeUtil®) to set the settings to their defaults or to those values which CARE would have chosen.
- **Using EXCELON 2.0 build 11 or higher**
  EXCELON 2.0 build 11 or higher offers a menu allowing the automatic changing of the net buffer settings.
- **Using NodeUtil®**
  With NodeUtil®, you will have to manually perform the net buffer setting changes as specified in Table 27.

In the case of I-Lon 1000, the buffers are already of sufficient size; thus, no alteration of the settings is necessary.

Table 27. Mandatory router buffer settings for BMI over LonWorks functionality (values in brackets are defaults)

type	count	size (bytes)	total size (bytes)
receive transaction	[3]	[13]	[39]
transmit transaction	[2]	[28]	[56]
application buffer in	[2]	[42]	[84]
application buffer out	[1]	[42]	[42]
net buffer in	[2]	**114 [66]**	228 [132]
net buffer out	**7 [15]**	**114 [66]**	798 [990]
application buffer out priority	[0]	[42]	[0]
net buff out priority	[2]	[114]	[228]
sum total	--	--	[1475]

Dial-Up Access Options

There are various different options for enabling dial-up (i.e. remote) access in a LONWORKS system equipped with Excel 50/500 controllers. The choice of the appropriate option depends upon

- whether you have Excel 50/500 firmware version 2.04.xx or 2.06.xx,
- whether or not you have established direct hardware connections from the front-end to a C-bus (besides to the already-existent LONWORKS bus),
- whether or not you are using LONWORKS plus BMF,
- the particular network management tool (LonMaker for Windows, CARE, etc.) you are using, and
- the particular front-end (Synopsis, EBI, XBS, SymmetrE) you are using.

The resultant degree of Building Management Functionality available via dial-up is listed in Table 1 on page 6 and in Table 2 on page 7.

Option 1: Dial-Up Access without Using a C-Bus

Under firmware version 2.04.xx, it is possible to enable dial-up access without having to establish direct hardware connections from the front-end to a C-bus (besides to the already-existent LONWORKS bus). To do this, map all those NVs with data-points that are needed for dial-up to one Excel 50/500 controller.

Advantage:	No C-bus need be installed, and thus no direct hardware connections between it and the front-end need be established.
Disadvantage:	The large number of data-point mappings in each Excel 50/500 controller.

Option 2: Dial-Up Access Using a C-Bus

Under firmware version 2.04.xx, it is possible to enable dial-up access by installing a C-bus (besides the already-existent LONWORKS bus).

Advantage:	No additional data-point mapping required.
Disadvantages:	A C-bus must be installed for every group (max. 30) of controllers. Depending upon the front-end you are using, a direct hardware connection must then be established from the front-end to each group.

Option 3: Dial-Up Access via LonWorks plus BMF

Under firmware version 2.06.xx and with CARE 4.xx, it is possible to enable dial-up access provided you use LONWORKS plus BMF.

Communication is then possible among a maximum of 30 Excel 50/500 controllers.

See also section "Activating and Configuring LonWorks", sub-section "CARE 4.01.03 and higher)" on page 15.

Option 4: Dial-Up Access Using an SLTA to Connect a Modem

Regardless of the firmware version, it is also possible to enable a standard dial-up access to a LONWORKS network by connecting a modem via an SLTA (Serial LonTalk Adapter).

This scenario supports only standard access to NVs in LONWORKS devices as defined by LONWORKS, and therefore does not support BMF as described above.

APPLICABLE LITERATURE

EN0B-0088GE51	Excel 50 Controller Specification Data
EN1B-0101GE51	Excel 50 Controller Installation Instructions
EN2B-0137GE51	Excel 50 Controller User Guide

EN2B-0166GE51	Excel 50 HT01 LonWorks Application
EN2B-0173GE51	Excel 50 HT02 LonWorks Application
EN2B-0179GE51	Excel 50 HE01 LonWorks Application
EN2B-0164GE51	Excel 50 AH01 LonWorks Application
EN2B-0165GE51	Excel 50 AH02 LonWorks Application
EN2B-0165GE51	Excel 50 AH03 LonWorks Application
EN1R-0143GE51	Excel 100 Controller Specification Data
EN1R-0144GE51	Excel 100 Controller Installation Instructions
EN0B-0089GE51	Excel 500/600 Controller Specification Data
EN0B-0203GE51	Excel 500-XCL5010 Specification Data
EN1R-1047GE51	Excel 500 Controller Installation Instructions
EN0B-0090GE51	Distributed I/O Product Data
EN2B-0092GE51	Excel 50/100/500/600 Controllers Software Description
EN0B-0091GE51	Excel 100/500/600 Controllers System Overview
EN2B-0126GE51	XI581AH/XI582AH Buswide Operator Interface User Guide
EN2B-0138GE51	CARE User Guide

ABBREVIATIONS AND ACRONYMS

binding The process that defines connections between the NVs of nodes. Connections define the data that nodes share with one another. The binding is performed during the commissioning phase using a LonWorks network manager tool.

LNS Acronym for LonWorks Network Services.

LonWorks Local Operations Network: A set of tools and devices forming a control network.

LonMark LonMark Association: An independent organization of LonWorks developers, system integrators, and end-users that define standards to ensure interoperability between LonWorks devices from multiple manufacturers.

LonWorks bus The wiring for the LonWorks network.

LonWorks NV A network variable created using CARE for open LonMark interoperability within the given application. In contrast to LonWorks NVs, there are also NVs for communicating with the Distributed I/O Modules via autobinding.

mapping The process of logically connecting an NV to one or more data-points or one or more data-points to one NV. This includes the data conversion of values.

MMI Acronym for Man-Machine Interface - operator tool connectable to the controller serial port.

NV Acronym for Network Variable on a LonWorks system: NVs are high-level data objects that LonWorks application nodes use to communicate with one another. The types, functions, and number of NVs are determined by the application code within the node.

SCPT Acronym for Standard Configuration Parameter Type: A reference to a predefined configuration type in the self-documentation for an NV that applies to the entire node, to one or multiple objects within the node, or to just one NV. Configuration parameters can be realized either as configuration NVs or in a Configuration File. For Excel 50/500, only configuration NVs are supported.

SNVT Acronym for Standard Network Variable Type. SNVTs facilitate interoperability by providing a well-defined interface for communication between nodes.

UCPT Acronym for User-Defined Configuration Parameter Type: UCPTs resemble SCPTs, but are not of a predefined type, see the Resource File Description delivered with Echelon's LNS package.

UNVT Acronym for User-Defined Network Variable Type. UNVTs are application-specific Network Variable Types and are thus not SNVTs.

XIF Acronym for External Interface File.

INDEX

Honeywell

Automation and Control Solutions
Honeywell GmbH
Böblinger Straβe 17
D-71101 Schönaich
Germany

http://europe.hbc.honeywell.com

DIN EN ISO
9001/14001