

# Conson XP Servicetool overview

[Connecting to XP130](#)

[Discovering modules on Conbus](#)

[Updating firmware](#)

[Log facilities](#)

[Loading and programming action tables](#)

## Connecting to XP130

---

The XP130 is the interface module between the Conbus and your network. It is connected to your network with an Ethernet cable.

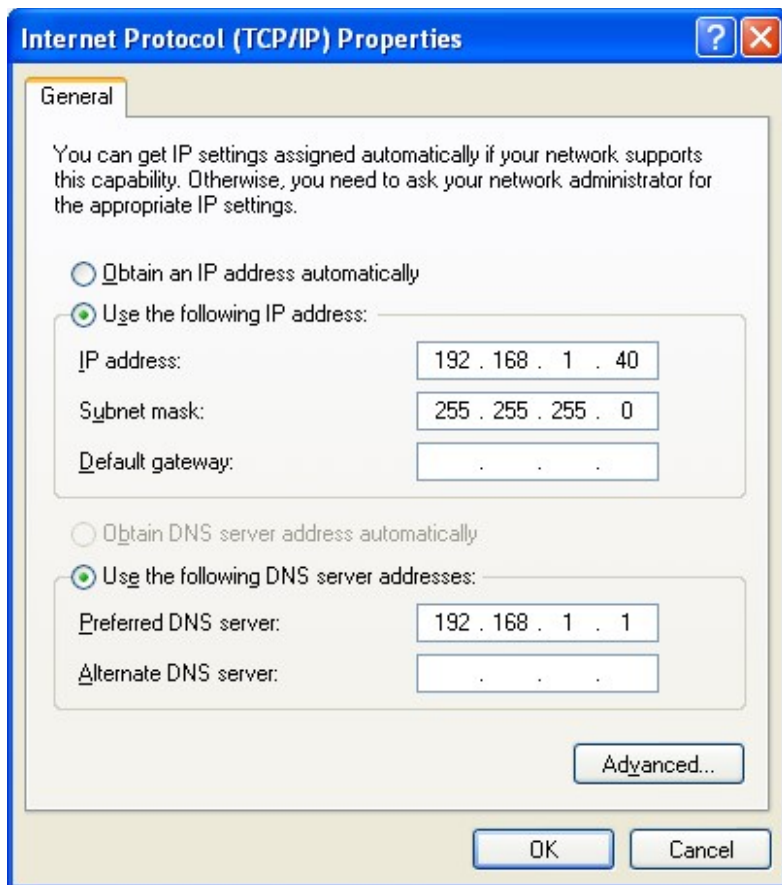
The default IP address for the XP130 is 192.168.1.17. If your network has some other network address like 192.168.2.x or 10.x.x.x, you have two options:

- Connect the XP130 directly to a spare Ethernet socket on your PC
- Change the default address of the XP130

### Connecting XP130 directly to your PC

Connect the XP130 Ethernet cable to a spare network socket on your PC, and check that the green link LED built into the XP130 Ethernet socket lights up. If the LED is not turned on, you need a crossed Ethernet cable for the connection.

When the connection is ok, open windows network connections (Start->Connect To->All Connections). Right-click on the connection you are using, click "Internet Protocol (TCP/IP) from the list, and click the "Properties" button. You should now see the window below, where you can enter an IP address.



## Changing IP address of XP130

If your network is not a 192.168.1.x network, or the default IP address 192.168.1.17 is used for something else, you will have to change the IP address of the XP130.

This is done using an external tool from Lantronix called **DeviceInstaller**. You can download it from [the Lantronix homepage](#).

The tool will find the XP130 on your network, and let you change the IP address. Follow the instructions in the Device Installer software.

## Discovering modules on the Conbus

After launching the XP service tool, you need to open a connection to a XP130 module. This is done by entering the IP address or hostname of the XP130 in the "XP130 address" box, and pressing the "Connect" button. When connection succeeds, the "Connect" button will change to "Disconnect".

To detect modules on the Conson bus, press the "Discover" button. This will detect the modules on the bus and show them in the "Module List" window.

You might need to press the "Discover" button a couple of times to get all modules and information in the module list. If you do not get any modules in the list, try "Disconnect" and "Connect" again.

## Detecting Consonbus

The pushbutton panels, connected via the Consonbus to the actor modules (relaymodules and lightdimmers), are also detected using the "Discover" function, and the module will show up below the actor module.

## Module Data

In the module data field, information about the module selected will be read out each time you select a module from the list.

## Output State

Output state can be read and set directly using check boxes in the "Output State" group. Note that support for setting output states over the Conbus is not available in all modules and firmware versions.

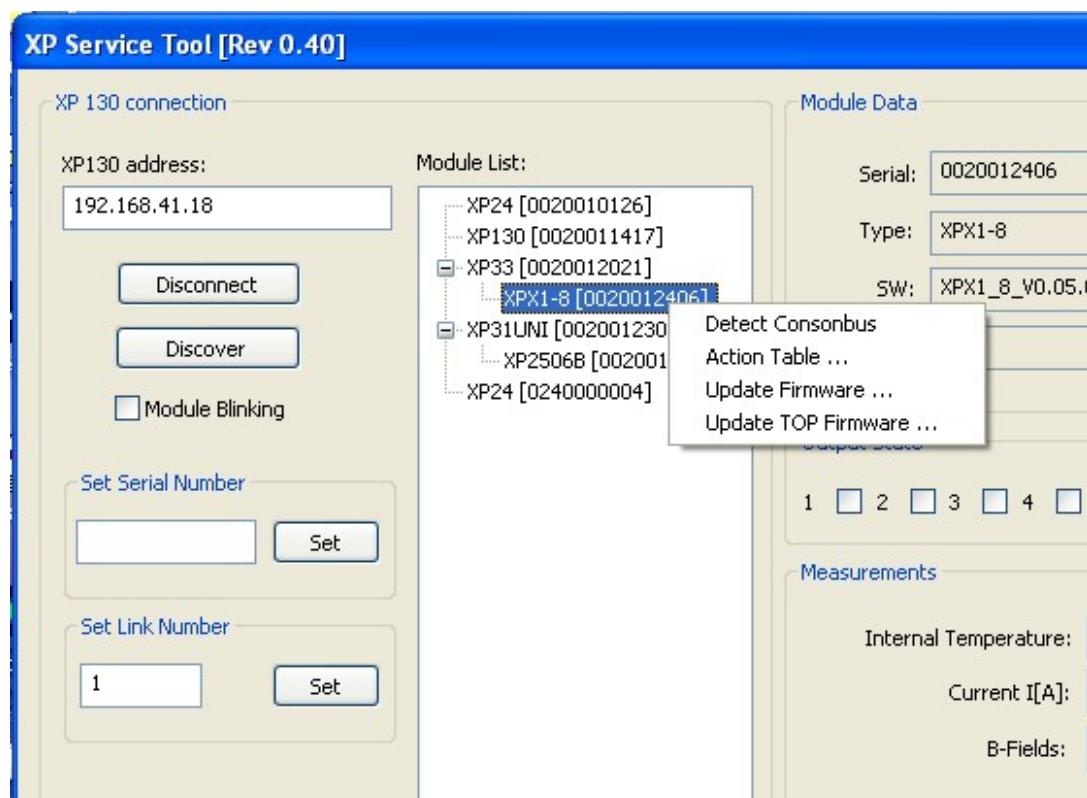
## Measurements

Some parameters can be read out from each module. Use the "Refresh Buttons" to get updated information from the selected module.

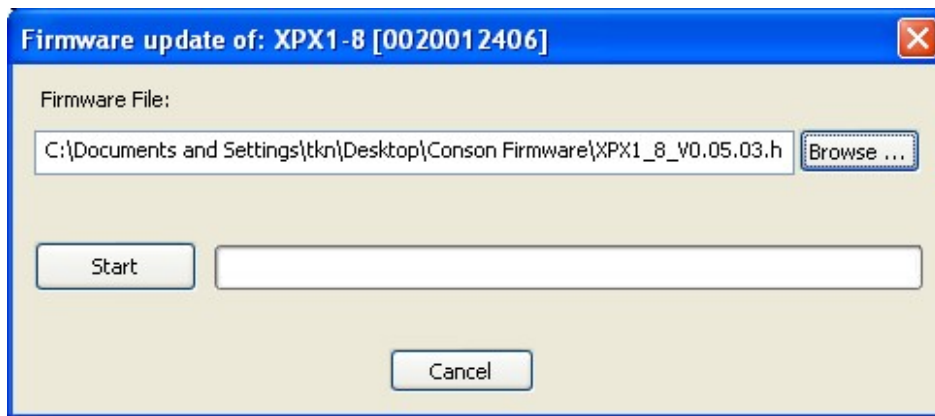
The Bfield measurements and the "Show Curve" button has to do with B field measurements from the built in HAL sensors. At the time of writing, this is only supported by XP24. The "Show Curves" button bring up a window with B field (current) waveforms from the selected module. It is updated each 10s.

# Updateing firmware

For updating firmware in the Conson modules and pushbutton panels, rightclick the module in the module list, and select "Update firmware" from the menu as shown below.



Selecting "Update firmware" will bring you to the dialog box shown below.



Here you select the firmware file (use the browse button to find the file), and when found press the "Start" button to start the firmware upload. Updateing af push button panel on the Consonbus will take a while longer then a module on the Conbus, because the communication speed is lower.

Some modules, like the XP31UNI, XP31BC and XP33 contains two internal processors, one located at the "bottom" PCB, which handles inputs and Conbus/Consonbus communication, and one located on a "top" PCB, handling the direct light dimming control. Updateing the top software is done selecting the "Update TOP firmware" menu item.

**Note!** Please be carefull to select the right software for download before pressing the "Start" button.

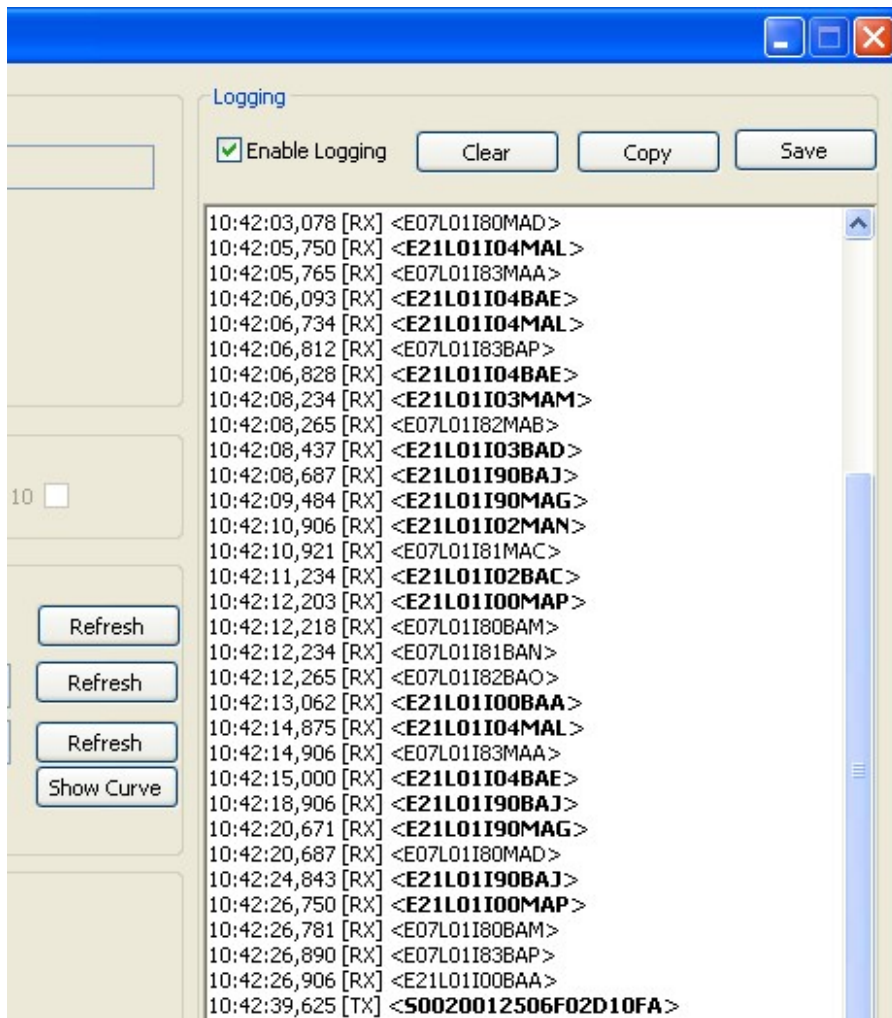
**Note!** If both top and bottom firmware should be updated for a light dimmer, you should start with the top. (If you have started with the bottom software, you might loose contact with the top software. In this case rewind bottom software.)

If the update fails, the module will be left in it's firmware update state, indicated by continous blinking. Just try updating again.

## Log Facillities

To hunt down configuration errors in an electrical installation with Conson XP modules, the XP service tool provides a log facillity.

Logging is enabled by checking he "Enable Logging" check box. Alle communication on the Conbus will henceforth be logged in the log window on the left of the XP servicetool. Se figure below.



The log window shows all data (telegrams) communicated on the Conbus. Each telegram starts with a "<" and ends with a ">". Note that telegrams going to or from the selected module are shown in bold in the log window.

There are three types of telegrams: [Event](#), [System request and Reply](#).

## Event Telegram

Event telegrams identifies an event in the system, e.g. user pressing a button on. Below is shown an example of an event telegram

```
<E14L00I02MAK>
```

The "E" identifies the telegram as being of the event type, the number following (14) is the module type code (see [more on module types](#)). The two digits after the "L" designates linknumber from 00 to 99, and the two digits after the "I" designates the input number. Finally the "M" indicates a button press (make), and a "B" designates a button release (break). The last two characters before the > is a checksum of the telegram.

The input numbers extends beyond push buttons. If the push button panel is equipped with an IR receiver, IR remote channels is mapped into the input space from 10 to 89.

The proximity sensor ("Sesame") on the Conson push button panels also emits an event. Input number 90 is used for sesam events.

# Generating System Report

To get an overview of your system modules, you can generate a system report, displaying data of the modules and buses.

The report is saved on your computer as a text file, that can easily be mailed together with e.g. a problem description.

To generate a report select File->System Report from the main menu. Then enter a name for your report file and press ok. The report is now generated.

## Contents of the report file

The report has the following sections:

- Header with date
- Summary of the modules in the system
- Detailed module report

The detailed module section contains information for each module:

- Module type name
- Serial number
- Link number
- Module number
- Internal temperature
- Firmware revision
- Top software revision if applicable

## System and Reply telegrams

System and reply telegrams are used to system related information like updateing firmware and reading the temperature from a module.

The system telegram shown below is used to request the temperature from a module.

```
<S0020012521F02D18FN>
```

System telegrams are identified by the "S", and followed by the receiver serial number. The two digits after "F" designates the system function (here "return data"), and the two digits after "D" is the data point ID (here temperature).

The system telegram above is answered using a reply telegram like shown below.

```
<R0020012521F02D18+26,0$CIL>
```

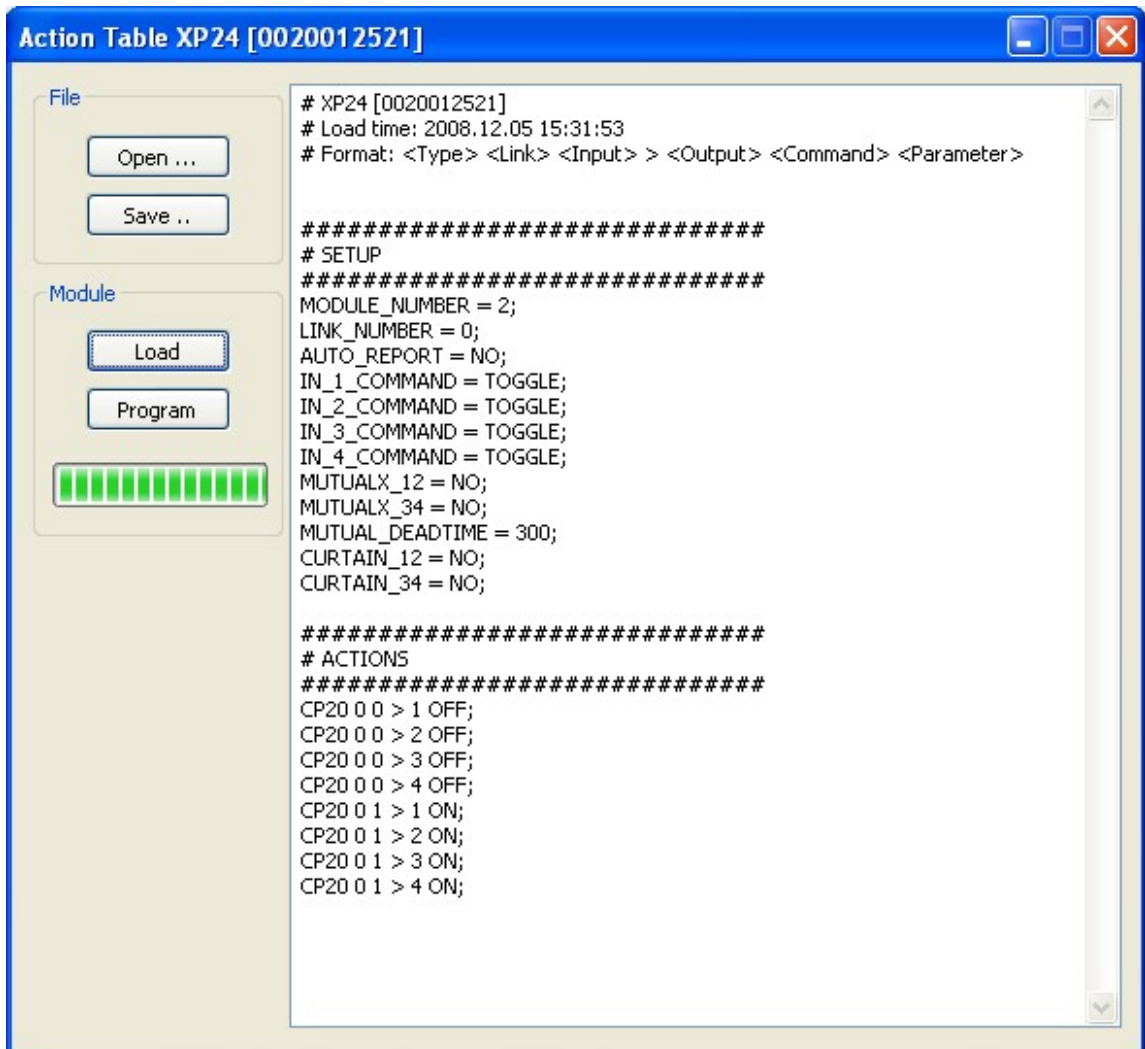
The "R" is for reply telegram, and is followed by the serial number of the sender of the telegram.

## Loading and Programming action tables

An action table inside each module defines how an output of the module is supposed to

react to a certain event. To load the action table into a human readable format, right-click the module in the module list, and select the "Action Table" menu entry.

Pressing the "Load" button should give you a result like the one shown below.



The Action table dialog consists of a code editor, for editing setup and action table, "Open" and "Save" buttons for saving action tables on your computer, and "Load" and "Program" buttons for loading from or programming to the module.

Old actions can be altered and new actions added by writing them in the code editor.

Read more about [coding action tables](#).

## Coding Action Tables

The action table is coded in the following way:

```
<Module Type> <Link Number> <Input Number> > <Output Number>
[Parameter] ;
```

Example: The center pushbutton on an XP2606 with link number 31, shall turn on relay 4 for 10 seconds.

Solution: The center pushbutton has input number 0, so we get the following line in the action table:

```
XP2606 31 0 > 4 ON 10s;
```

[See the complete list of actions](#), [and the list of module types to use](#).

Actions are normally run when receiving the "make" event from a push-button. (AUXRELAY is an exception, since it will turn the relay ON on a MAKE event, and OFF again on the break event). You can "invert" this functionality using the tilde sign ~ in front of the action:

```
XP28 1 0 > 1 ~ON 10s;
```

The action line above will trigger when the signal on the first input of XP28 with linknumber 1 is REMOVED - that is a BREAK is received from the XP28

[See advanced examples here](#)

## Auto reporting from Actor modules

Each module can send out auto report [events](#), when an output changes state. The auto report event telegram uses the module type and link number of the actor module, with output 1 corresponds to event input number 80, output 2 to event inputnumber 81 and so forth.

When an output turns ON, a MAKE event is send, when an outputs turns OFF a BREAK event is send. The autoreport events is both send out on the Conbus, and handled internally by the module - that is, it handles it's own report events.

Auto report events can be used to control a LED in a push-panel for indication of light state.

Auto reporting is not enabled by default. To enable it, set the setup token "AUTO\_REPORT" to "YES".

## Module typecodes

The following module type codes exists in the XP system:

Module	Code	Description
NOMOD	0	No module
ALLMOD	1	Code matching every moduletype
CP20	2	CP switch link module
CP70A	3	CP 38kHz IR link module
CP70B	4	CP B&O IR link module
CP70C	5	CP UHF link module
CP70D	6	CP timer link module
XP24	7	XP relay module
XP31UNI	8	XP universal load light dimmer
XP31BC	9	XP ballast controller, 0-10V



XP31DD	10	XP ballast controller DSI/DALI
XP33	11	XP 33 3 channel lightdimmer
CP485	12	CP RS485 interface module
XP130	13	Ethernet/TCPIP interface module
XP2606	14	5 way push button panel with sesam, L-Team design
XP2606A	15	5 way push button panel with sesam, L-Team design and 38kHz IR receiver
XP2606B	16	5 way push button panel with sesam, L-Team design and B&O IR receiver
XP26x1	17	Reserved
XP26x2	18	Reserved
XP2506	19	5 way push button panel with sesam, Conson design
XP2506A	20	5 way push button panel with sesam and 38kHz IR, Conson design
XP2506B	21	5 way push button panel with sesam and B&O IR, Conson design
XPX1-8	22	8 way push button panel interface
XP134	23	Junctionbox interlink

## Actions

Listed below are the possible output actions in an action table.

Action name	Description
VOID	No (empty) actions
TURNON	Turn output on. Can be given a time parameter to limit the on period
TURNOFF	Turn output off. Can be given a time parameter to limit the off period
TOGGLE	Toggle function (and regulate for dimmers)
BLOCK	Block further actions on output. The block is active between the make and break event
AUXRELAY	Turns on the output between a make and a break event
MUTUALEX	Mutual exclusion between outputs
LEVELUP	Regulate lightlevel up between make and break
LEVELDOWN	Regulate lightlevel down between make and break event
LEVELINC	Increment lightlevel
LEVELDEC	Decrement lightlevel
LEVELSET	Set lightlevel
FADETIME	Set fadetime
SCENESET	Set light scene
SCENENEXT	Change to next light scene
SCENEPREV	Change to previous light scene
CTRLMETHOD	Reserved

RETURNDATA	Return data from module
DELEAYEDON	Delay on action with time as parameter
FADE	Fade light
LEARN	Enter learn state
EVENTTIMER1 to EVENTTIMER4	Four event timers, that will send out an event message after the time specified in the parameter. The event will use the host module type (if the function is used in a XP24 module, the event module type will be XP24), the host module link number, and "inputs" 70 to 73 for EVENTTIMER1 to EVENTTIMER4 respectively.
STEPCTRL	Used for controlling multispeed ceiling fans. The STEPCTRL command will use releay 1 and 2 or 3 and 4 in pairs, and will cycle through 4 states (off-off, on-on, on-off, off-on)
STEPCTRLUP	Like STEPCTRL, except that it stops at the on-on state
STEPCTRLDOWN	Like STEPCTRLUP, except that it walks the states in the opposite direction, and stops at off-off

## Advanced Examples

### Fan control

A bathroom fan control should start some time after (ex. 2 minutes) the light is switched on, and it should turn off e.g. 5 minutes after the bathroom light is turned off.

If we have an XP24 (linknumber 3), where relay 1 is used for bathroom light, and relay 2 is used for the fan, the following two action lines will get you there:

```
XP24 3 80 > 2 DELAYEDON 2M;
XP24 3 80 > 2 ~OFF 5M;
```

The event "XP24 3 80" is an [auto report event](#) (remember to enable auto reporting for the module).

The event used in the example above comes from the bathroom light (relay 1 remember). Turning on the bathroom light will result in the "DELAYEDON .." action beeing started, which will turn on releay 2 in 2 minutes.

When the bathroom light is switched off, the "XP24 3 80" is send again, now as a BREAK event. Using the tilde (~) in front of an action, tells the module to activate the action on a BREAK event instead of a MAKE event.

Note that the bathroom light could be connected to any actor module (relay or dimmer) on the Conbus, just change the event specification.

If, for instance, the bathroom light is controlled by a XP31UNI having linknumber 6, the action lines will simply change to:

```
XP31UNI 6 80 > 2 DELAYEDON 2M;
XP31UNI 6 80 > 2 ~OFF 5M;
```