

IsoLOG Control

The IsoLOG 3D ARRAY is currently controlled by regular HTTP requests. Commands are transferred by requesting cgi-scripts with arguments, the current device state can be accessed by reading the „status.xml“ file.

Currently there are three commands:

select.cgi

Arguments:

- `switch=<num>` - num is the two-digit decimal value of the element to be activated on the selected sector, starting at 0
- `port=<num>` - num is the decimal value of the radial sector to be selected, starting at 0
- `amp=<1|0>` - enable or disable the preamplifier when switching the given port (optional)

Example:

`http://1.2.3.4/select.cgi?switch=01&port=4&=0` - Activate the second element in the fifth sector, and disable the preamplifier

There is no return value specified at this time.

Note: The preamplifier state is a global setting. If the preamplifier state is not specified it will keep the previous state even though the antenna might have multiple preamplifiers for different ports.

autorotate.cgi

Arguments:

- `automode=<1|0>` - `automode=1` to enable hardware rotation, `automode=0` to disable hardware rotation („chopper mode“)

There is no return value specified at this time.

Note: Current versions of the IsoLOG 3D ARRAY will always cycle clockwise between all antennas in hardware rotation mode. Later Resale versions may offer customizable patterns (like only horizontal orientation, only a single sector or counter-clockwise-rotation).

Note 2: When disabling chopper mode the currently selected port is undefined, the values in status.xml will reflect the state before chopper mode was activated.



amplifier.cgi

Arguments:

- amp=<1|0> - enable or disable the preamplifier

There is no return value specified at this time.

Note: This setting applies to all preamplifiers attached to the antenna, even if only one is going to actually be active.

IsoLOG status

Antenna information

The status.xml file contains several fields describing the configuration of the connected IsoLOG model. These fields are static, their content might only change after a firmware update.

<model>	The base model of the IsoLOG
<serial>	Individual serial number of the IsoLOG
<options>	Which optional components are included in this specific IsoLOG model
<fwversion>	Firmware version used on this IsoLOG model
<sectors>	Number or directional sectors, usually 8 or 16
<elements>	Maximum number of connected elements per directional sector, typically 2, 3 or 4
<element_labels>	A semicolon-separated list of labels for the different elements in each sector, this should match the <elements> value. Generally this will contain the antenna type connected.
<disabled_ports>	A semicolon-separated list of ports that are not actually connected to an antenna and therefore useless. Ports are specified as pair of sector and element separated by underscore

Example:

<disabled_ports>1_3;3_3;5_3;7_3</disabled_ports> disables the last element on every second sector on a 8-sector configuration.



Antenna status

The following fields are dynamically updated by the firmware when executing commands:

<switch00> ... <switch03>	The sector index currently selected
<switch16>	The element index currently selected
<chopper>	1 if chopper mode is currently active, 0 if it is not active (manual port selection)
<amplifier>	1 if the preamplifiers are active, 0 if they are not active (bypass mode)

IP Discovery

The IsoLOG obtains automatically an IP address using DHCP, or if no DHCP server is present in the connected network using a hardwired address.

The userspace application can detect this by sending a UDP broadcast packet on port 30303 with the payload

“Discovery: Who is out there?\n”

at the connected subnet, wait for replies on the same port and evaluate the IP addresses.

The content of the reply is not defined.



Programming Examples

Address discovery using UDP broadcast in python (examples are the minimum required without error checking or output):

```
import socket as S
myhost = S.gethostbyname(S.gethostname())
sock = S.socket(S.AF_INET, S.SOCK_DGRAM, S.IPPROTO_UDP)
sock.bind((myhost, 30303))
sock.setsockopt(S.SOL_SOCKET, S.SO_BROADCAST, True)
sock.sendto("Discovery: Who is out there?\n", ("<broadcast>", 30303))
data, addr = sock.recvfrom(1024)
```

Enabling first port (assuming 8-sector, 2-element configuration):

```
import urllib
# example, replace with actual IP address as determined in example above
address = "192.168.0.55"
# switch address must be passed with two digits
urllib.urlopen("http://" + address + "/select.cgi?port=0&switch=00")
```

Enabling last port (assuming 8-sector, 2-element configuration) and enable preamp:

```
# first comes the 8-way switch
urllib.urlopen("http://" + address + "/select.cgi?port=7&switch=01&=1")
```

Enabling chopper mode:

```
urllib.urlopen("http://" + address + "/autorotate.cgi?automode=1")
```

Reading current status:

```
url = urllib.urlopen("http://" + address + "/status.xml")
status = url.read()
# 'status' now contains the state as XML string.
# This can be processed by the various xml modules
```

