RFXCOM

RFXLAN
xPL

www.rfxcom.com

Contents

1. Introduction

The RFXLAN transceiver is a 10/100Mb LAN interface with a 433.92MHz receiver and RFXmitter.
The RFXLAN can also have an optional Visonic receiver installed.

The LAN interface controller can be flashed with TCP/IP firmware or xPL firmware.
With the TCP/IP firmware loaded you can connect to the receiver on port 10001 and to the RFXmitter on port 10002.
With the xPL firmware loaded the receiver messages are transmitted on the LAN using the standard UDP port 3865 for xPL.
xPL cmnd messages received from the LAN interface are translated to RF commands or to I/O commands to control I/O output lines.

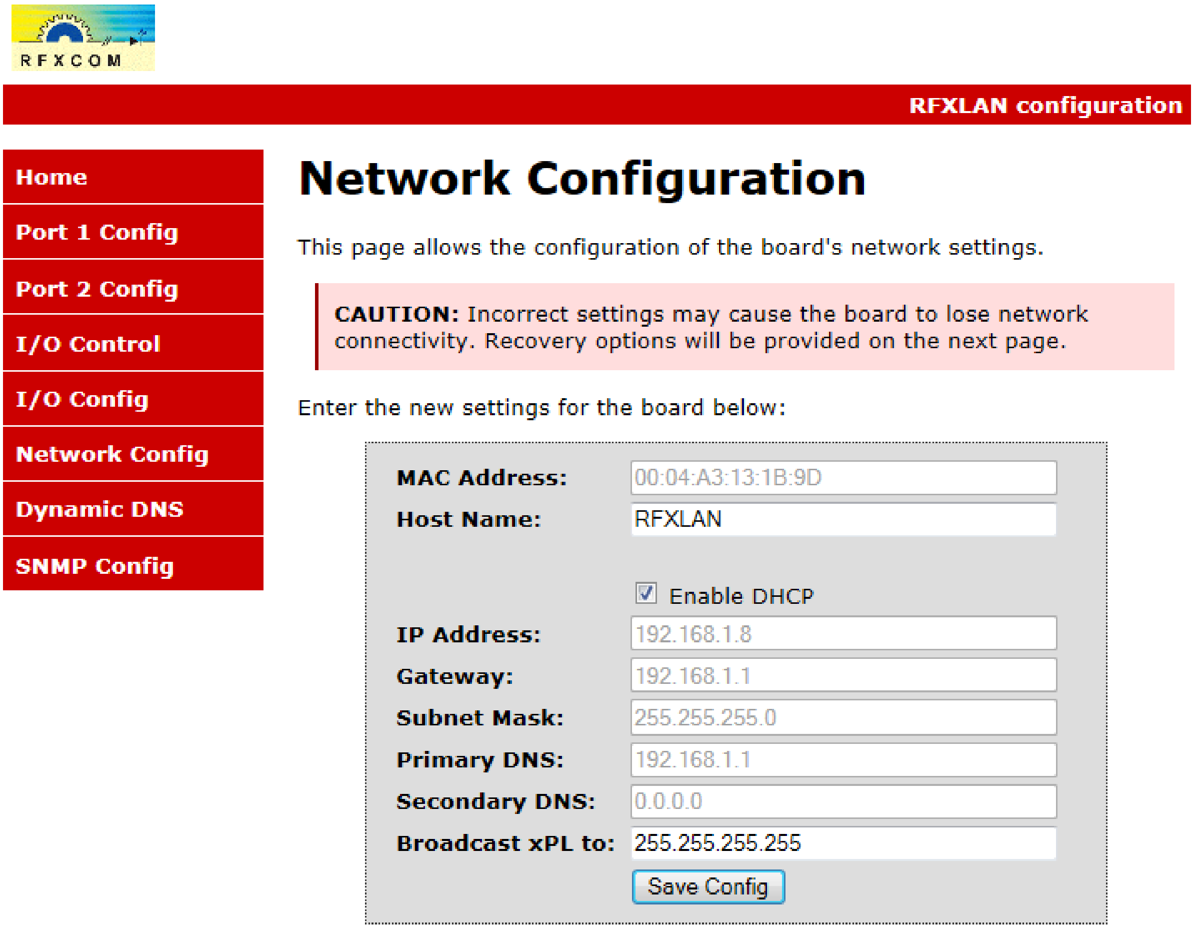
Important: it is advised to configure unused I/O lines to input with pull-up or as output to avoid unwanted sensor.basic messages from unused I/O lines.

Note: New schemas and additions to existing schemas are marked in light green.

With thanks to Mal Lansell for allowing me to copy the text below from his web site.

2. RFXLAN configuration

Configure the network settings in the RFXLAN if required:



Network Configuration

This page allows the configuration of the board's network settings.

CAUTION: Incorrect settings may cause the board to lose network connectivity. Recovery options will be provided on the next page.

Enter the new settings for the board below:

MAC Address:	00:04:A3:13:1B:9D
Host Name:	RFXLAN
	☑ Enable DHCP
IP Address:	192.168.1.8
Gateway:	192.168.1.1
Subnet Mask:	255.255.255.0
Primary DNS:	192.168.1.1
Secondary DNS:	0.0.0.0
Broadcast xPL to:	255.255.255.255

3. Quick start

3.1. On Linux

- Download xpl-perl https://github.com/beanz/xpl-perl/
  Follow the README on that page how and what to install.

- Start the xpl-hub:
  ```
  xpl-hub -i eth0 -v --define broadcast=0.0.0.0 &
  ```

- Start xpl-logger to check if the RFXLAN heart-beat messages are received. The heart-beat is transmitted 5 seconds after power up and after that every 5 minutes.

  ```
  root@server:~ # xpl-logger -i eth0 -v --define broadcast=255.255.255.255
  Listening on 192.168.0.1:48216
  Sending on 255.255.255.255
  Adding client: 192.168.0.1:48216 "bnz-listener.server" 192.168.0.1:35701
  xpl-stat/hbeat.app: bnz-listener.fi... 192.168.0.1:35701
  xpl-stat/hbeat.basic: rfxcom-lan.00...
  ```

3.2. *on Windows*

- Connect the RFXLAN with RFXxPL firmware flashed.

- Download and run xPL infrastructure check package
  http://www.thijsschreijer.nl/blog/?page_id=150

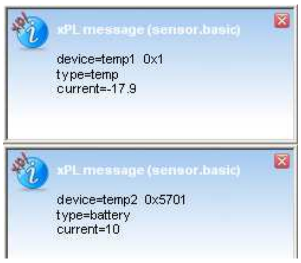
- Download and install the xPLHAL Manager for Windows
  http://xplproject.org.uk/?page_id=9&did=1

- Download and install xPL Monkey HUB (if not installed by xPL infrastructure check package)
  http://www.xplmonkey.com/downloads/xplhub.msi

- Download and install Medusa Desktop Balloon Popup
  http://xplproject.org.uk/?page_id=9&did=3

- Enable UDP port 3865 in the firewall or disable the firewall.

- Start xPLBalloon. You should see now xPL messages from the RFXLAN.



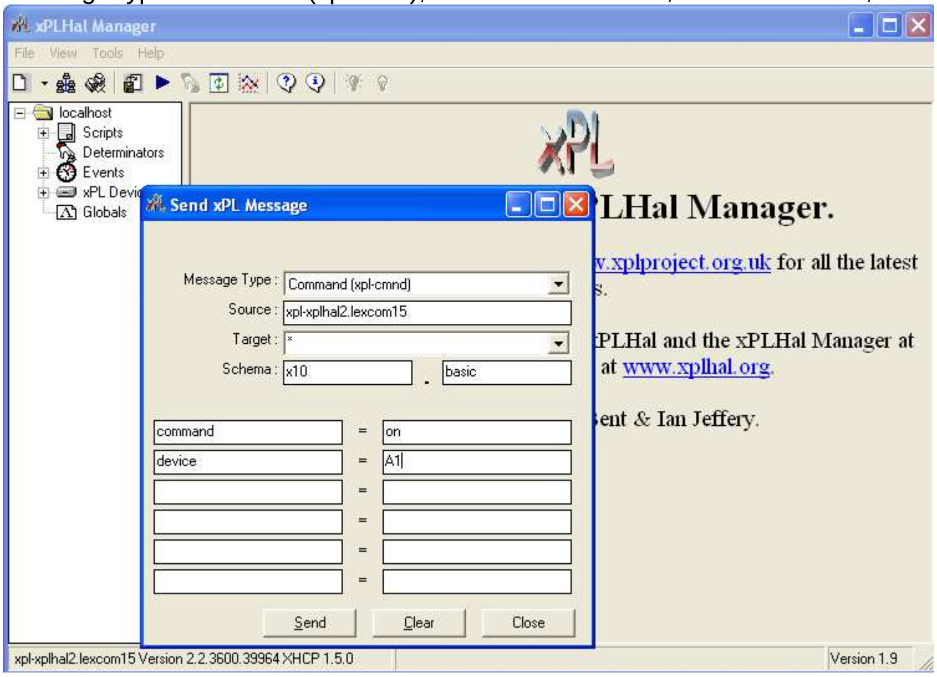
xPL message (sensor.basic)

device=temp1 0x1
type=temp
current=-17.9

xPL message (sensor.basic)

device=temp2 0x5701
type=battery
current=10

- Start xPLHAL Manager and select: Tools – Send xPL Message.
  Send an xPL message for example:
  Message type: Command(xpl-cmd), Schema: x10.basic, command = on, device = A1



Message Type : Command (xpl-cmnd)
Source : xpl-xplhal2.lexcom15
Target : *
Schema : x10 . basic

command = on
device = A1

xpl-xplhal2.lexcom15 Version 2.2.3600.39964 XHCP 1.5.0
Version 1.9

Press Send and the RFXmitter will transmit the X10 A1-On RF command.

- See also: http://xplproject.org.uk/wiki/index.php?title=Get_Started_on_Windows
  xplhub installation instructions:
  http://xplproject.org.uk/wiki/index.php?title=Install_the_xPL_Monkey_Windows_xPL_Hub

4. xPL Message Schema

Based on http://xplproject.org.uk/wiki/index.php?title=XPL_Message_Schema

4.1. xPL protocol core schemas used by the RFXLAN

4.1.1. hbeat.basic - Device heartbeat signaling

```
xpl-stat
{
hop=1
source= rfxcom-lan.<hex MAC address>
target=*
}
hbeat.basic
{
interval=5
version=<version info>
ip=<IP address>
}
```

source is rfxcom-lan.<hex MAC address> Hex MAC address is the 12 hex characters MAC address, for example: source=rfxcom-lan.0004a3131b9d

4.1.2. hbeat.request – Heartbeat response on device discovery

If the RFXLAN receives this message

```
xpl-cmnd
{
hop=1
source=<source address of sending xPL device>
target=*| rfxcom-lan.<hex MAC address>
}
hbeat.request
{
command=request
}
```

It will respond with an hbeat.basic

4.1.3. status.request

If the RFXLAN receives this message:

```
xpl-cmnd
{
hop=1
source=xpl-xplhal.myhouse
target=rfxcom-lan.<hex MAC address>
}
sensor.request
{
request=current
device=(io0-io7)
}
```

It will respond with an xpl-stat sensor.basic reporting the RFXLAN I/O status. See the sensor.basic for RFXLAN I/O description in chapter 4.3.10

4.2. xPL protocol core schemas for the RFXLAN transmitter

4.2.1. xpl-cmnd - xPL message type cmnd

Each xPL message for the RFXLAN transmitter will start with:

```
xpl-cmnd
{
hop=1
source=<source address of sending xPL device>
target=*|rfxcom-lan.<hex MAC address>
}
```

A target of * is send to all xPL devices.
To address a specific RFXLAN use the RFXLAN address which is: rfxcom-lan.<hex MAC address>
For example: target =rfxcom-lan.0004a3131b9d

IMPORTANT:
A maximum of 40 transmit messages are buffered. Messages will be ignored if more messages are transmitted. Each command transmitted by the RFXmitter will send an xpl-trig for the transmitted RF message.

4.2.2. Transmit buffer log messages

If the transmit buffer reaches a threshold level with space for 5 messages a log.basis – type=wrn is send.
If the buffer is back again below this threshold level it will send a log.basis – type=inf.
If no space is available in the transmit buffer a log.basis – type=err is send. Messages are lost now.

```
log.basic
{
type=wrn
text=transmit buffer can accept <n> messages.
}
```

```
log.basic
{
type=err
text= transmit buffer full!
}
```

```
log.basic
{
type=inf
text= transmit buffer below threshold.
}
```

4.2.3. x10.basic - X10 lighting, ARC, Flamingo, IKEA Koppla and Waveman

This message is used to control X10, ARC devices, Flamingo, IKEA Koppla, Waveman and Harrison. ARC devices have code wheels (A-P/1-16) and are delivered under several brand names using the same protocol. (See the list below)

```
x10.basic
{
device=<house code[device code]>
command=on|off|dim|bright|all_lights_on|all_lights_off
[level=(0-100)]
[protocol=arc|flamingo|koppla|waveman|harrison|he105|rts10]
}
```

device=<house code[device code]>
house code = A to P or lower case a to p
note: trig messages as a response on this command will always have the house code in upper case.

device code = 1 to 16. Device code is not used in case of all_lights_on/off.
For he105 only house codes A and B are allowed. See the HE105 address conversion in the appendix.

command=on|off|dim|bright|all_lights_on|all_lights_off
The X10 lighting command to be transmitted.

level=(0-100)
set dim level is only used by IKEA Koppla in steps of 10.

protocol=arc|flamingo|koppla|waveman|harrison|he105|rts10
Default protocol is X10.

protocol=arc is used for units with address wheels (A-P/1-16) of the following brands:
KlikAanKlikUit,
Chacon,
HomeEasy,
DomiaLite,
Domia,
ByeByeStandBy,
ELRO AB600,
NEXA
Proove
Intertechno
Duwi
Commands allowed: on|off|all_lights_on|all_lights_off

protocol=flamingo
Address allowed: A-P / 1-16 See the Flamingo, AB4xx address conversion in the appendix.
The Flamingo is an AB400D unit so other AB4xx units may operate as well using this protocol.
Commands allowed: on|off

protocol=koppla
Address allowed: A-P / 1-10
Commands allowed: on|off|dim|bright|all_lights_on|all_lights_off

protocol=waveman
Commands allowed: on|off|all_lights_on|all_lights_off
Note:
The trig message as a response on a cmnd with protocol=waveman will be protocol=arc

protocol=he105
Address allowed: A-B /1-16
Commands allowed: on|off
The HE105 is a HomeEasy relay with selectable: dry contact or 230V contact.

protocol=rts10

Address allowed: A-P /1-16.
Commands allowed: on|off|all_lights_on|all_lights_off
all_lights_on/off is program the address in the RTS10 that is set to program mode.
Note:
The trig message as a response on a cmnd with protocol=rts10 will be a sensor.basic digimax message.

protocol=harrison
Commands allowed: on|off|dim|bright|all_lights_on|all_lights_off
on is open,
off is close,
dim or bright is stop,
all_lights_on/off is program the address switches in the curtain motor that is set to program mode.

The address (A-P/1-16) is converted to the Harrison address in the curtain motor. See the appendix for details.

4.2.4. ac.basic - control of AC code devices

This message is used to control AC type devices (with program/learn button):

- ANSLUT,
- Chacon,
- DI.O,
- KlikAanKlikUit,
- NEXA,
- Proove,
- Intertechno,
- Düwi,
- HomeEasy UK,
- HomeEasy EU.

This schema is partly comparable with the homeeasy.basic schema. The ac.basic schema does not support he105 and only a hex address.

```
ac.basic
{
address=(0x1-0x3ffffff)
unit=(1-16)|group
command=on|off|preset
[level=(0-15)]
[eu=true]
}
```

address=(0x1 – 0x3ffffff)
The address used by the AC device. In the RF protocol this is a 26bit value, which should be specified as a hexadecimal number with a leading 0x. Use the same address as received from an already paired remote or select any value you want and pair the module like you do with a remote. (push the learn button on the module and transmit an ON command). Never use an address with all zeroes!

unit=(1-16)|group
The unit number of the device. This is a value between 1 and 16, and can also be a special reserved unit called "group". The "group" unit code is especially useful with computer control, since two consecutive group on commands will not cause the device to go into its dimming cycle.

command=on
Turns the device on. An AC device is paired with the transmitter by putting the device into setup mode and sending an on command.

command=off
Turns the device off. If the device is in setup mode, sending an off command will unpair it from the transmitted address+unit.

command=preset
level=(0-15)
Sets the device to specific dim level. Level 0 is OFF and 15 is the brightest level.

eu=true
This will transmit RF codes for HomeEasy EU.

4.2.5. x10.security

This message is used to control an X10 alarm panel or to transmit an X10 security sensor signal that can be received by an X10 alarm panel.

```
x10.security
{
command=alert|normal|motion|arm-home|arm-away|disarm|panic|lights-on|lights-off
device=<device id>
[delay=min|max]
}
```

command=alert|normal|motion|arm-home|arm-away|disarm|panic|lights-on|lights-off
The X10 security command to be transmitted.

command=alert
The security sensor has been triggered.

command=normal
The cause of the alert state has been removed.

command=motion
Movement has been detected.

command=arm-home
Sent to arm the system while the house is still occupied - at night, for example. The usual method is to arm "perimeter" sensors such as door, window and glass-break detectors, while ignoring motion sensors.

command=arm-away
Sent to arm the system while the house is. In this mode all sensors will be armed.

command=disarm
Sent by security remote controls to disarm the system.

command=panic
Sent by security remote controls to immediately trigger the alarm.

command=lights-on
Sent by security remote controls to turn on the set of lights that are controlled by the alarm console.

command=lights-off
Sent by security remote controls to turn off the set of lights that are controlled by the alarm console.

device=<device id>
Unique identifier of the device sending the message. The id is a hex number between 0x0 and 0xffffff. It is also referred to as the device address.

delay=max
Some sensors have a switch which changes their alert message so that the alarm console knows to allow a delay period before triggering the alarm. If not present, the delay should be assumed to be "min".

4.2.6. remote.basic

This message is used to transmit RF commands as if they were transmitted by an X10 PC Remote, ATI remote or Medion remote.

```
remote.basic
{
keys=0x<hex keycode>
device=pc|ati|medion (0x0-0xF)
}
```

keys=<0x hex keycode>
0x followed by the 2 digits hex keycode: see appendix.

device=rf|ati|medion
device=pc is an X10 PC Remote, address is optional
device=ati is an ATI remote
device=medion is for the medion remote

address is the remote hex channel from 0x0 to 0xF. Optional for device=pc and only used for cursor commands, default is 0xE
Note: ati_plus is not supported by the transmitter but only by the receiver.

4.2.7. control.basic – X10 Ninja/Robocam (not yet implemented)

This message is used to control the X10 Ninja/Robocam.

```
control.basic
{
device=<house code>
type=ninja
current=left|right|up|down|p1|prog_p1|p2|prog_p2|p3|prog_p3|p4|prog_p4|center
|prog_center|sweep|prog_sweep
}
```

device=<house code>
ninja address is a house code A to P

type=ninja
The X10 Ninja/Robocam is a camera positioner.

current=left|right|up|down|p1|prog_p1|p2|prog_p2|p3|prog_p3|p4|prog_p4|center|prog_center|sweep|prog_sweep

4.2.8. control.basic – Mertik-Maxitrol (not yet supported)

This message is used to control the Mertik Maxitrol gas fires.

```
control.basic
{
device=(0x0-0xF)
type=mertik
current=on|off|step_up|step_down|run_up|run_down|stop
}
```

device=(0x0-0xF)
mertik address is a hex code 0x0 to 0xF

type=mertik

current=on|off|step_up|step_down|run_up|run_down|stop

A stop command will be send 12 seconds after the **run_up** or **run_down** has been send.
Important: only one Mertik can be controlled by the RFXLAN or do not transmit a run_up/down for another Mertik within 12 seconds.

4.2.9. control.basic – RFXLAN I/O line

This message is used to control the 8 I/O lines of the RFXLAN.

```
control.basic
{
device=(io0-io7)
type=output
current=high|low|toggle
}
```

device=(io0-io7)
IO line to control

type=output

current=high|low|toggle
high – set I/O line high
low – set I/O line low
toggle – set the opposite of the current state

4.3. xPL protocol core schemas used by the RFXLAN receiver

4.3.1. xpl-trig –xPL message type trig

Each xPL message from the RFXLAN receiver will start with:

```
xpl-trig
{
hop=1
source=rfxcom-lan.<hex MAC address>
target=*
}
```

source=rfxcom-lan.<hex MAC address>
Hex MAC address is the 12 hex characters MAC address, for example:
source=rfxcom-lan.0004a3131b9d

4.3.2. log.basic

This is used for reporting of an error status of the RFXLAN or unknown sensor messages received

```
log.basic
{
type=inf|wrn|err
text=<text>
[code=<received code>]
}
```

type=inf|wrn|err
inf = information message
wrn = warning message, used if transmit buffer has reached the minimum threshold.
err = error message, used to indicate an error message.

text=<text>
type=inf - text=unknown sensor
This message is send if an unknown RF sensor message has been received. In most cases this is a disturbed RF packet or an unimplemented sensor.
type=inf – text=transmit buffer below threshold
This message is send if the transmit buffer is below threshold and can accept a new transmitter command without the risk on buffer overflow.
type=wrn - text=transmit buffer can accept x messages.
This message indicates the number of free slots for transmit messages in the transmit buffer.
type=err - text=transmit buffer full!
Too many messages are send and the transmit buffer has been overflowed.

For the RFXSenor the next type=inf messages are send:
text=address incremented");
text=no 1-wire device connected");
text=1-wire rom crc error");
text=not a DS1820 or DS2438");
text=no end of read received");
text=1-wire scratchpad crc error");
text=unknown error

code=<received code>
This field is displayed with type=inf and it concerns an unknown sensor message is received. Received code is the received RF packet data in hex

4.3.3. x10.basic - X10 lighting

Received X10 lighting or ARC messages.

```
x10.basic
{
device=<house code[device code]>
command=<x10 command>
}
```

device=<house code[device code]>
house code = A to P
device code = 1 to 16. Device code is not present in case of all_lights_on/off.

command=on|off|dim|bright|all_lights_on|all_lights_off
The X10 lighting command received.

Note: Flamingo or AB4xx remotes are received but the information received is useless.

4.3.4. ac.basic – AC code remotes and wall switches

Received messages from AC type remotes or AC type RF wall switches (with program/learn button):

- ANSLUT,
- Chacon,
- DI.O,
- KlikAanKlikUit,
- NEXA,
- Proove,
- Intertechno,
- Düwi,
- HomeEasy UK,
- HomeEasy EU.

This schema is partly comparable with the homeeasy.basic schema. This ac.basic schema does not support he105 and accepts only a hex address.

```
ac.basic
{
address=(0x1-0x3ffffff)
unit=(1-16)|group
command=on|off|preset
[level=(0-15)]
}
```

address=(0x1 – 0x3ffffff)
The address used by the AC device. In the RF protocol this is a 26bit value, which should be specified as a hexadecimal number with a leading 0x. Never use an address with all zeroes!

unit=(1-16)|group
The unit number of the device. This is a value between 1 and 16, and can also be a special reserved unit called "group". The "group" unit code is especially useful with computer control, since two consecutive group on commands will not cause the device to go into its dimming cycle.

command=on
Turns the device on. An AC device is paired with the transmitter by putting the device into setup mode and sending an on command.

command=off
Turns the device off. If the device is in setup mode, sending an off command will unpair it from the transmitted address+unit.

command=preset
level=(0-15)
Sets the device to specific dim level. Level 0 is OFF and 15 is the brightest level.

4.3.5. x10.security

Received X10, Visonic PowerCode/CodeSecure security or Chacon, Avidsen, NEXA smoke detector message received.

```
x10.security
{
command=alert|normal|motion|light|dark|arm-home|arm-away|disarm|panic|lights-
on|lights-off
device=<device id>
[type=cn|ds10|ds90|sd90|ms10|ms20|ms90|kr10|sh624|mct234|mct302|mct550|mcw]
[tamper=true]
[low-battery=true]
[delay=max]
}
```

command=alert|normal|motion|light|dark|arm-home|arm-away|disarm|panic|lights-on|lights-off
The command received from an X10 security sensor or remote.

command=alert
The security sensor has been triggered.

command=normal
The cause of the alert state has been removed.

command=motion
Movement has been detected.

command=light
Sent by some motion detectors when the light level has risen above a preset threshold.

command=dark
Sent by some motion detectors when the light level has fallen below a preset threshold.

command=arm-home
Sent by security remote controls to arm the system while the house is still occupied - at night, for example. The usual method is to arm "perimeter" sensors such as door, window and glass-break detectors, while ignoring motion sensors.

command=arm-away
Sent by security remote controls to arm the system while the house is. In this mode all sensors will be armed.

command=disarm
Sent by security remote controls to disarm the system.

command=panic
Sent by security remote controls to immediately trigger the alarm.

command=lights-on
Sent by security remote controls to turn on a set of lights that are controlled by the alarm console.

command=lights-off
Sent by security remote controls to turn off the set of lights that are controlled by the alarm console.

device=<device id>
Unique identifier of the device sending the message. The id is a hex number between 0x0 and 0xffffff. It is also referred to as the device address.

type=cn|ds10|ds90|sd90|ms10|ms20|ms90|kr10|sh624|mct234|mct302|mct550|mcw
cn is a Chacon, Avidsen, NEXA smoke detector
mct and mcw are Visonic PowerCode sensors
other types are X10
These messages are possible:
type=cn
type=ds10,ds90
type=sh624,mct234
type=sh624
type=ds10,ds90,sd90,mct302,mct550
type=kr10
type=kr10,sd90,mct234
type=ms10,ms20,ms90,mcw
type=mcw
type=ms10,mcw
type=ms20
type=ds90
type=ds90,mct302,mct550
type=ms90,mcw
type=sd18,c018
type=hp564
type=dm10
type=unknown

tamper=true
Optional item that indicates whether the security device has been tampered with (usually triggered by opening the case). If not present, the tamper state should be assumed to be false.

low-battery=true
Optional item that indicates whether the security device needs to have its battery replaced. If not present, the low-battery state should be assumed to be false.

delay=max
Some sensors have a switch which changes their alert message so that the alarm console knows to allow a delay period before triggering the alarm. If not present, the delay should be assumed to be "min".

4.3.6. remote.basic

Received X10 PC Remote, Medion remote, ATI Remote or ATI Remote Plus

```
remote.basic
{
keys=<0x hex keycode>
device=pc|ati|medion|ati_plus [<address>]
}
```

keys=<0x hex keycode>
0x followed by the hex keycode: see appendix.

device=rf|ati|medion|ati_plus
device=pc is an X10 PC Remote, address is only used at cursor commands.

device=ati is an ATI remote

device=medion is for the medion remote

device=ati_plus is an ATI-Plus remote
Note: the high order bit of the ATI-PLUS remote indicates an odd or even commend so that multiple pushes are detectable. To find the correct key code and the received hex code with 0x7F.

address is the remote hex channel from 0x0 to 0xF.

4.3.7. sensor.basic – Digimax

```
sensor.basic
{
device=digimax 0x<hex device id>
type=temp
current=<degrees Celsius>
units=c
}
```

```
sensor.basic
{
device=digimax 0x<hex device id>
type=setpoint
current=<degrees Celsius>
units=c
}
```

```
sensor.basic
{
device=digimax 0x<hex device id>
type= demand
current=heater_on|heater_off|cooling_on|cooling_off
}
```

4.3.8. sensor.basic – RFXSensor

```
sensor.basic
{
device=rfxsensor 0x<hex device id>
type=temp
current=<degrees Celsius>
units=c
}
```

```
sensor.basic
{
device=rfxsensor 0x<hex device id>
type=voltage
current=<volt>
}
```

Two voltage sensors can be transmitted.
The sensor with an odd device id is the A/D voltage sensor.
The sensor with an even device id is the system voltage (regulated battery voltage).
A system voltage < 4.72Volts means battery empty.

The A/D sensor can have a humidity or barometric sensor connected.
The formula to calculate the relative humidity RH:

RH = (((A/D voltage / supply voltage) - 0.16) / 0.0062) / (1.0546 - 0.00216 * temperature)

Formula to calculate the barometric pressure in hPa:

hPa = ((A/D voltage / supply voltage) + 0.095) / 0.0009

4.3.9. sensor.basic – RFXMeter

```
sensor.basic
{
device=rfxmeter 0x<hex device id>
type=count
current=<value>
}
```

current=<value>
This value is the counter value of the RFXMeter. The maximum value before roll over is 16777215

4.3.10. sensor.basic – RFXLAN I/O lines

This message is received in case of a state change of one of the 8 I/O lines of the RFXLAN or as answer on a status request.

```
sensor.basic
{
device=(io0-io7)
type=input
current=high|low
}
```

device=(io0-io7)
IO line to control

type=output

current=high|low
high – I/O line is high
low – I/O line is low

4.3.11. sensor.basic – Mertik-Maxitrol (not yet supported)

This message is received on a control command for the Mertik Maxitrol gas fires.

```
sensor.basic
{
device=(0x0-0xF)
type=mertik
current=on|off|step_up|step_down|run_up|run_down|stop
}
```

device=(0x0-0xF)
mertik address is a hex code 0x0 to 0xF

type=mertik

current=on|off|step_up|step_down|run_up|run_down|stop

4.3.12. sensor.basic - Oregon sensors

Currently supported Oregon Scientific sensors are:

ID	name	type
Oregon-THR128 Oregon-THR138 Oregon-THC138	temp1	Inside Temperature
Oregon-THC238/268 Oregon-THN122N/132N Oregon-THWR288A Oregon-THRN122N Oregon-AW129 Oregon-AW131	temp2	Outside/Water/BBQ Temp
Oregon-THWR800	temp3	Water temperature
Oregon-RTHN318	temp4	Outside Temperature
Oregon-THGN122N/123N Oregon-THGR122NX Oregon-THGR228N Oregon-THGR238/268	th1	Inside Temp-Hygro
Oregon-THGR810	th2	Inside Temp-Hygro
Oregon-RTGR328N	th3	Outside Temp-Hygro
Oregon-THGR328N	th4	Outside Temp-Hygro
Oregon-WTGR800	th5	Outside Temp-Hygro
Oregon-THGR918 Oregon-THGRN228NX Oregon-THGN500	th6	Outside Temp-Hygro
Huger - BTHR918	thb1	Inside Temp-Hygro-Baro
Oregon-BTHR918N Oregon-BTHR968	thb2	Inside Temp-Hygro-Baro
Oregon-RGR126 Oregon-RGR682 Oregon-RGR918	rain1	Rain Gauge
Oregon-PCR800	rain2	Rain Gauge
Oregon-WTGR800	wind1	Anemometer
Oregon-WGR800	wind2	Anemometer
Huger-STR918 Oregon-WGR918	wind3	Anemometer
Oregon-UVN128 Oregon-UV138	uv1	UV sensor
Oregon-UVN800	uv2	UV sensor
Oregon-RTGR328N	dt1	Date & Time
Oregon-BWR102	WEIGHT1	Body Weight Monitor
Oregon-GR101	WEIGHT2	Body Weight Monitor
cent-a-meter Electrisave OWL CM113	elec1	Ampere meter
OWL CM119, CM160	elec2	Power meter
OWL CM180	elec3	Power meter
OWL CM180i	elec4	Ampere+Power meter

```
sensor.basic
{
device=(temp1-temp4|th1-th6|thb1|thb2) 0x<hex sensor id>
type=temp
current=<degrees Celsius>
units=c
}
```

```
sensor.basic
{
device=(th1-th6|thb1|thb2) 0x<hex sensor id>
type=humidity
current=(0-100)
description=normal|comfort|dry|wet
}
```

```
sensor.basic
{
device=(th1-th6|thb1|thb2) 0x<hex sensor id>
type=status
current=normal|comfort|dry|wet
}
```

```
sensor.basic
{
device=thb1|thb2 0x<hex sensor id>
type=pressure
current=<hPa>
units=hpa
forecast=sunny|partly cloudy|cloudy|rain
}
```

```
sensor.basic
{
device=rain1|rain2 0x<hex sensor id>
type=rainrate
current=<mm/hr>
units=mmh
}
```

```
sensor.basic
{
device=rain1|rain2 0x<hex sensor id>
type=raintotal
current=<mm>
units=mm
}
```

```
sensor.basic
{
device=wind1|wind2|wind3 0x<hex sensor id>
type=gust
current=<m/sec>
units=mps
}
```

```
sensor.basic
{
device=wind1|wind2|wind3 0x<hex sensor id>
type=average_speed
current=<m/sec>
units=mps
}
```

```
sensor.basic
{
device=wind1|wind2|wind3 0x<hex sensor id>
type=direction
current=(0-359)
}
```

```
sensor.basic
{
device=uv1|uv2 0x<hex sensor id>
type=uv
current=(0-12)
description=low|medium|high|very high|dangerous
}
```

```
sensor.basic
{
device=(temp1-4|th1-6|thb1-2|rain1-2|wind1-3|uv1-2|elec1-4) 0x<hex sensor id>
type=battery
current=(0-100)
}
```

Note: for sensors with an OK - Not OK battery indication, a value 10 is reported if battery empty.

```
sensor.basic
{
device=weight1|weight2 0x<hex sensor id>
type=weight
current=<kg>
units=kg
}
```

```
datetime.basic
{
datetime=<date and time as yyyymmddhhmmss>
date=<date as yyyymmdd>
time=<time as hhmmss>
}
```

```
sensor.basic
{
device=elec1_1 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec1_2 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec1_3 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec2 0x<hex sensor id>
type=power
current=<kilowatt>
units=kw
}
```

```
sensor.basic
{
device=elec2 0x<hex sensor id>
type=energy
current=<kWh>
units=kwh
}
```

```
sensor.basic
{
device=elec4_1 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec4_2 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec4_3 0x<hex sensor id>
type=current
current=<ampere>
}
```

```
sensor.basic
{
device=elec4 0x<hex sensor id>
type=energy
current=<kWh>
units=kwh
}
```

5. Appendix

5.1. *RFXLAN schemas*

The RFXLAN service sends and acts on messages using a number of xPL schemas. These are outlined in the table below:

Protocol	Transmitter Schema	Receiver Schema	Comments
Standard X10 RF	x10.basic	x10.basic	
Units with code wheels(A-P/1-16) BBSB Chacon Domia Lite ELRO AB600 HomeEasy KlikAanKlikUit NEXA	x10.basic	x10.basic	Commands for these devices are mapped to standard X10 house and unit codes, and so appear as X10 devices in xPL. The transmitter schema must have set: protocol=arc
IKEA Koppla	x10.basic	N/A	IKEA have now discontinued the Koppla range. Commands for Koppla devices are mapped to standard X10 house and unit codes. The transmitter schema must have set: protocol=koppla
Flamingo	x10.basic	N/A	Flamingo AB400 devices are mapped to X10 house and unit codes. They appear as standard X10 devices to xPL. The transmitter schema must have set: protocol=flamingo
Harrison	x10.basic	N/A	Harrison devices are mapped to X10 house and unit codes. They appear as standard X10 devices to xPL. The transmitter schema must have set: protocol=harrison
HE105	x10.basic	N/A	HE105 appear as a control device to xPL.
RTS10	x10.basic	N/A	RTS10 appear as a control device to xPL.
Units with program button: Chacon HomeEasy KlikAanKlikUit NEXA	ac.basic	ac.basic	These devices use an AC protocol, translated to ac.basic in xPL.

Security X10	x10.security	x10.security	The sensor ID is 3 bytes
Visonic	N/A	x10.security	Visonic CodeSecure and PowerCode commands are mapped to X10 security commands, and so appear as X10 security devices to xPL.

X10 PC Remote ATI Remote Medion Remote	remote.basic	remote.basic	
ATI Remote Wonder Plus	N/A	remote.basic	

Thermostat: Digimax 210 TH10	N/A	sensor.basic	Thermostat data is converted to three sensor.basic messages - two temperatures (current temperature and the thermostat setpoint) and an output representing the demand for heating or cooling
RFXSensor	N/A	sensor.basic	Sensor data is sent in up to three sensor.basic messages - the current temperature, the sensor battery voltage and the measured sensor value. The measured A/D sensor value must be converted by the application to humidity or barometer or can be used as voltage value.
RFXMeter	N/A	sensor.basic	The counter value is sent as a sensor.basic message.
Oregon Scientific	N/A	sensor.basic datetime.basic	Oregon weather sensors typically report more than one value. Each value is sent in xPL using its own sensor.basic message. The datetime.basic schema is used to report the date and time provided by certain Oregon sensors that include an "Atomic Clock" feature.
Cent-a-meter Electrisave CM113	N/A	sensor.basic	The CM113 device can have up to three current transformers attached.
CM119 CM160 CM180	N/A	sensor.basic	
CM180i	N/A	sensor.basic	The CM180i device can have up to three current transformers attached.

5.2. *Flamingo, AB4xx address conversion to switch settings*

HC	switch	DC	switch	switch
	1 2 3 4		5 6 7 8	9 10
A	0 0 0 0	1	0 0 0 0	0 0
B	0 0 0 1	2	0 0 0 1	
C	0 0 1 0	3	0 0 1 0	
D	0 0 1 1	4	0 0 1 1	
E	0 1 0 0	5	0 1 0 0	
F	0 1 0 1	6	0 1 0 1	
G	0 1 1 0	7	0 1 1 0	
H	0 1 1 1	8	0 1 1 1	
I	1 0 0 0	9	1 0 0 0	
J	1 0 0 1	10	1 0 0 1	
K	1 0 1 0	11	1 0 1 0	
L	1 0 1 1	12	1 0 1 1	
M	1 1 0 0	13	1 1 0 0	
N	1 1 0 1	14	1 1 0 1	
O	1 1 1 0	15	1 1 1 0	
P	1 1 1 1	16	1 1 1 1	

0 = toggle switch off, 1 = toggle switch on

Examples:

xPL	1 2 3 4 5 6 7 8 9 10
A1	0 0 0 0 0 0 0 0 0 0
A15	0 0 0 0 1 1 1 0 0 0
N2	1 1 0 1 0 0 0 1 0 0
N11	1 1 0 1 1 0 1 0 0 0

5.3. Harrison address conversion to switch settings

The address used is converted to the address selected in the Harrison curtain motor using the table below.

```
switch 1 2 3 4         5 6 7 8
       H H H H         X X X X
   A   0 1 1 0      1  0 0 0 0
   B   0 1 1 1      2  0 0 0 1
   C   0 1 0 0      3  0 0 1 0
   D   0 1 0 1      4  0 0 1 1
   E   1 0 0 0      5  0 1 0 0
   F   1 0 0 1      6  0 1 0 1
   G   1 0 1 0      7  0 1 1 0
   H   1 0 1 1      8  0 1 1 1
   I   1 1 1 0      9  1 0 0 0
   J   1 1 1 1     10  1 0 0 1
   K   1 1 0 0     11  1 0 1 0
   L   1 1 0 1     12  1 0 1 1
   M   0 0 0 0     13  1 1 0 0
   N   0 0 0 1     14  1 1 0 1
   O   0 0 1 0     15  1 1 1 0
   P   0 0 1 1     16  1 1 1 1

H H H H = House code
X X X X = device code

Switch position in the motor:
Up    = 1
Middle      = not used!!!!
Down  = 0
```

Examples:
If you assign the X10 address E7 (1000 0110) to the curtain motor then set the switches to: 1=up, 2=down, 3=down, 4=down, 5=down, 6=up, 7=up, 8=down

If you assign the X10 address A2 (0110 0001) to the curtain motor then set the switches to: 1=down, 2=up, 3=up, 4=down, 5=down, 6=down, 7=down, 8=up

5.4. HomeEasy Heater HE105 address conversion to switch settings

```
 xPL  | HE105 switches
======================
HC  DC  unit 1 2 3 4 5
A   1    0   0 0 0 0 0
A   1    1   0 0 0 0 0
A   2    2   0 0 0 0 1
A   3    3   0 0 0 1 0
A   4    4   0 0 0 1 1
A   5    5   0 0 1 0 0
A   6    6   0 0 1 0 1
A   7    7   0 0 1 1 0
A   8    8   0 0 1 1 1
A   9    9   0 1 0 0 0
A   10   10  0 1 0 0 1
A   11   11  0 1 0 1 0
A   12   12  0 1 0 1 1
A   13   13  0 1 1 0 0
A   14   14  0 1 1 0 1
A   15   15  0 1 1 1 0
A   16   16  0 1 1 1 1
B   1    17  1 0 0 0 0
B   2    18  1 0 0 0 1
B   3    19  1 0 0 1 0
B   4    20  1 0 0 1 1
B   5    21  1 0 1 0 0
B   6    22  1 0 1 0 1
B   7    23  1 0 1 1 0
B   8    24  1 0 1 1 1
B   9    25  1 1 0 0 0
B   10   26  1 1 0 0 1
B   11   27  1 1 0 1 0
B   12   28  1 1 0 1 1
B   13   29  1 1 1 0 0
B   14   30  1 1 1 0 1
B   15   31  1 1 1 1 0
B   16   32  1 1 1 1 1
```

5.5. Mertik Maxitrol address conversion to switch settings

Address	switch position			
	1	2	3	4
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

5.6. Remote codes

5.6.1. PC Remote Code table

Dec	Hex	Button
2	02	0
18	12	8
34	22	4
56	38	Rewind
58	3A	Info
64	40	CHAN+
66	42	2
82	52	Ent
96	60	VOL+
98	62	6
99	63	Stop
100	64	Pause
112	70	Cursor-left
113	71	Cursor-right
114	72	Cursor-up
115	73	Cursor-down
116	74	Cursor-up-left
117	75	Cursor-up-right
118	76	Cursor-down-right
119	77	Cursor-down-left
120	78	left mouse
121	79	left mouse-End
123	7B	Drag
124	7C	right mouse
125	7D	right mouse-End
130	82	1
146	92	9
160	A0	MUTE
162	A2	5
176	B0	Play
182	B6	Menu
184	B8	Fast Forward
186	BA	A+B
192	C0	CHAN-
194	C2	3
201	C9	Exit
209	D1	MP3
210	D2	DVD
211	D3	CD
212	D4	PC / Shift-4
213	D5	Shift-5
214	D6	Shift-Ent
215	D7	Shift-Teletext
216	D8	Text
217	D9	Shift-Text
224	E0	VOL-
226	E2	7
242	F2	Teletext
255	FF	Record

5.6.2. ATI Remote Wonder Code table

Dec	Hex	Button
0	00	A
1	01	B
2	02	power
3	03	TV
4	04	DVD
5	05	?
6	06	Guide
7	07	Drag
8	08	VOL+
9	09	VOL-
10	0A	MUTE
11	0B	CHAN+
12	0C	CHAN-
13	0D	1
14	0E	2
15	0F	3
16	10	4
17	11	5
18	12	6
19	13	7
20	14	8
21	15	9
22	16	txt
23	17	0
24	18	snapshot ESC
25	19	C
26	1A	^
27	1B	D
28	1C	TV/RADIO
29	1D	<
30	1E	OK
31	1F	>
32	20	<-
33	21	E
34	22	v
35	23	F
36	24	Rewind
37	25	Play
38	26	Fast forward
39	27	Record
40	28	Stop
41	29	Pause
44	2C	TV
45	2D	VCR
46	2E	RADIO
47	2F	TV Preview
48	30	Channel list
49	31	Video Desktop
50	32	red
51	33	green
52	34	yellow
53	35	blue
54	36	rename TAB
55	37	Acquire image
56	38	edit image
57	39	Full screen
58	3A	DVD Audio
112	70	Cursor-left
113	71	Cursor-right
114	72	Cursor-up
115	73	Cursor-down
116	74	Cursor-up-left
117	75	Cursor-up-right
118	76	Cursor-down-right
119	77	Cursor-down-left
120	78	V
121	79	V-End
124	7C	X
125	7D	X-End

5.6.3. ATI Remote Wonder Plus Code table

Dec	Hex	Button
0	00	A
1	01	B
2	02	power
3	03	TV
4	04	DVD
5	05	?
6	06	Guide
7	07	Drag
8	08	VOL+
9	09	VOL-
10	0A	MUTE
11	0B	CHAN+
12	0C	CHAN-
13	0D	1
14	0E	2
15	0F	3
16	10	4
17	11	5
18	12	6
19	13	7
20	14	8
21	15	9
22	16	txt
23	17	0
24	18	Open Setup Menu
25	19	C
26	1A	^
27	1B	D
28	1C	FM
29	1D	<
30	1E	OK
31	1F	>
32	20	Max/Restore Window
33	21	E
34	22	v
35	23	F
36	24	Rewind
37	25	Play
38	26	Fast forward
39	27	Record
40	28	Stop
41	29	Pause
42	2A	TV2
43	2B	Clock
44	2C	TV
45	2D	VCR
46	2E	RADIO
47	2F	TV Preview
48	30	Channel list
49	31	Video Desktop
50	32	red
51	33	green
52	34	yellow
53	35	blue
54	36	rename TAB
55	37	Acquire image
56	38	edit image
57	39	Full screen
58	3A	DVD Audio
112	70	Cursor-left
113	71	Cursor-right
114	72	Cursor-up
115	73	Cursor-down
116	74	Cursor-up-left
117	75	Cursor-up-right
118	76	Cursor-down-right
119	77	Cursor-down-left
120	78	Left Mouse Button
121	79	V-End
124	7C	Right Mouse Button
125	7D	X-End

5.6.4. Medion Remote Code table

Dec	Hex	Button
0	00	Mute
1	01	B
2	02	power
3	03	TV
4	04	DVD
5	05	Photo
6	06	Music
7	07	Drag
8	08	VOL-
9	09	VOL+
10	0A	MUTE
11	0B	CHAN+
12	0C	CHAN-
13	0D	1
14	0E	2
15	0F	3
16	10	4
17	11	5
18	12	6
19	13	7
20	14	8
21	15	9
22	16	txt
23	17	0
24	18	snapshot ESC
25	19	DVD MENU
26	1A	^
27	1B	Setup
28	1C	TV/RADIO
29	1D	<
30	1E	OK
31	1F	>
32	20	<-
33	21	E
34	22	v
35	23	F
36	24	Rewind
37	25	Play
38	26	Fast forward
39	27	Record
40	28	Stop
41	29	Pause
44	2C	TV
45	2D	VCR
46	2E	RADIO
47	2F	TV Preview
48	30	Channel list
49	31	Video Desktop
50	32	red
51	33	green
52	34	yellow
53	35	blue
54	36	rename TAB
55	37	Acquire image
56	38	edit image
57	39	Full screen
58	3A	DVD Audio
112	70	Cursor-left
113	71	Cursor-right
114	72	Cursor-up
115	73	Cursor-down
116	74	Cursor-up-left
117	75	Cursor-up-right
118	76	Cursor-down-right
119	77	Cursor-down-left
120	78	V
121	79	V-End
124	7C	X
125	7D	X-End

5.7. RFXCOM

5.7.1. RFXCOM xPL changes/additions

1. Added to the x10.basic schema:
   protocol= arc, he015 and rts10

2. ac.basic schema added

3. The device id in the x10.security is 1 to 3 bytes instead of 1 byte.

4. control.basic schema used for the X10 Ninja/Robocam

5. control.basic schema used for the Mertik Maxitrol

6. Added to sensor.basic for the Digimax:
   demand=heater_on|heater_off|cooling_on|cooling_off

7. Added to sensor.basic for Oregon humidity sensors:
   description=normal|comfort|dry|wet

8. Added to sensor.basic for Oregon barometric sensors:
   forecast=sunny|partly cloudy|cloudy|rain

9. Types added to sensor.basic:
   rainrate
   raintotal
   gust
   average_speed

10. For future use add this type to sensor.basic:
    lightlevel=<lux or lumen>

6. Handy tools

To monitor network traffic:
http://www.wireshark.org/download.html

Thijs Schreijer's xPL protocol analyzer for Wireshark
http://www.thijsschreijer.nl/blog/?p=479

Thijs Schreijer's xPL infrastructure check package
http://www.thijsschreijer.nl/blog/?page_id=150

Several useful tools:
http://xpl.lhopital.org/download/

Revision history.

Version 0.0 – April 30, 2011
  Initial version.

Version 1.0 – May 3, 2011
  sensor.basic schemas added.

Version 2.0 – May 4, 2011
  control.basic

Version 3.1 – May 5, 2011
  he105 and rts10 added to x10.basic
  sensor.basic follows the xPL standard
  rts10 address changed
  hbeat.basic remote-ip changed to ip

Version 3.2 – May 6, 2011
  Stop cmd send for Mertik cont_up/down
  Buffer for max 12 RFXmitter messages

Version 3.3 – May 7, 2011
  Quick Start added.

Version 4.0 – May 12, 2011
  Approved xPL schemas

Version 4.1 – May 15, 2011
  ATI-PLUS received key code.
  X10 security Lights-on/off added for the transmitter.

Version 4.2 – May 16, 2011
  Chacon, Avidsen, NEXA smoke detectors added

Version 4.3 – May 30, 2011
  I/O lines added

Version 5.0 – June 7, 2011
  RF Remote renamed to PC Remote
  In remote.basic: device=rf changed to device=pc
  PC Remote codes 70 and 72 changed to 63 and 64
  eu=true removed from received ac.basic command (only used for transmit)
  units= added to sensors with ambiguous values
  transmit buffer information, warning and error log.basic messages added

Version 6.0 – June 14, 2011
  xPL infrastructure check package added to Quick start.
  sensor.basic Mertik added

Version 6.1 – July 18, 2011
  AC address starts at 1.

Version 6.2 – August 21, 2011
  log.basic added

Version 7.0 –September 15, 2011
  xPL broadcast IP address added
  Quick start for Linux added

Version 7.1 –September 17, 2011
  type= added for all received security sensors

Version 7.2 –September 28, 2011
  type= messages expanded.

Version 7.3 – October 17, 2011
  Flamingo address code updated and AB4xx added.

Version 7.4 – October 28, 2011
  description=low|medium|high|very high|dangerous added to UV sensors

Version 7.5 – November 9, 2011
  Notes added at x10.basic for trig responses on cmnd’s

Version 7.6 – December 5, 2011
  ac.basic unit numbers changed from 0-15 to 1-16
  (add 1 to all your configured ac.basic devices!!!)

Version 7.7 – February 29, 2012
  Max. transmit commands changed from 12 to 40

Version 7.8 – January 4, 2013
  elec3 and elec4 added

7. Copyright notice