# D:\Hobby\Prosjekter\MultiTRX\Bilder\multitrx_batch.jpg

MultiTRX, firmware 1.0

Operating manual

Second revision

Jon Petter Skagmo 2011 - skagmo.com

Contents

[Introduction and overview 1](#_Toc304102914)

[Getting started 1](#_Toc304102915)

[Receiving data 1](#_Toc304102916)

[Sabotage control 1](#_Toc304102917)

[Controlling power-switches 2](#_Toc304102918)

[Gao-specific 2](#_Toc304102919)

[Nexa-specific 2](#_Toc304102920)

[Complete command list 3](#_Toc304102921)

[Commands transmitted from MultiTRX (RF-reception) 3](#_Toc304102922)

[Commands transmitted to the MultiTRX (RF-transmission) 3](#_Toc304102923)

[Other commands transmitted to the MultiTRX 3](#_Toc304102924)

[Compatible hardware 4](#_Toc304102925)

[Sensors 4](#_Toc304102926)

[Power switches 4](#_Toc304102927)

[Hardware specification 4](#_Toc304102928)

# Introduction and overview

Tons of wireless sensors, remote power switches, alarms etc. use the 433.92 MHz ISM band for wireless transmission of data. Unfortunately, not many of these products can be easily controlled with a computer. The MultiTRX aims to solve the problem of easily decoding and encoding of signals from cheap consumer electronics from stores like Clas Ohlson. It does a lot of things not present in any other device. Among other exiting features it decodes and encodes the unique ID of remote power switches, enabling you to keep the pairing with your existing remote control.

# Getting started

Once plugged in, the MultiTRX will appear as a serial port on your computer. On Windows systems, check device manager to find out which COM-port number has been assigned to the device. On Linux based systems, the device will appear in */dev/* as *ttyUSBx*.

The MultiTRX will light both leds for 2.5 seconds, which indicates that the MultiTRX is ready to accept a firmware update. If a boot loader is not detected, both leds will flash rapidly. This indicates a successful startup.

To test the device in Windows, download a serial terminal software like PuTTY, and start a serial session at 9600 baud (8N1). Go to *Change Settings...*, *Terminal* and click *Implicit CR in every LF*. Send *$P* to the device and you should receive the version number. Data received from RF should now appear in the terminal as they are received.

# Receiving data

The MultiTRX can decode data from Esic temperature sensors, Doro Secur sensors and Nexa and GAO remote controls. See the complete command list on the next page.

## Sabotage control

To make sure this device will not be used to sabotage other users of remote power switches, a software limit on reception of Nexa and GAO devices has been added. This limit disables printing of these devices until at certain number of equal packets has been received. This means you have to hold the button on your remote a few seconds before the full packet is shown. It is however still possible to let the MultiTRX print your remote control ID at the first packet received if you have stored it's unique ID in the MultiTRXs EEPROM ID-memory. Doing so is described in a these steps:

1. Hold your remote close to the MultiTRX and transmit until a full packet is shown in your terminal. Pay attention to the unique ID (GAO devices have a three hex character ID, and Nexa devices have a seven character unique ID, check the command list for details).

Let's say you receive the Nexa-string *$N1234ABC030*. The unique ID is then *1234ABC.*

1. Save the data in the ID-memory by writing *$S01234ABC0*. The first *0* after the *S* is the memory location. Memory 1-8 are all valid choices. The next seven digits is the unique ID, and the leading zero is a padding character. This command needs the ID to be exactly eight characters, so if the ID is shorter than eight characters, add extra zeros until the MultiTRX prints *$OK*.
2. Type *$R* to read all eight positions of the ID-memory, just to check that your recently added ID is there

# Controlling power-switches

The easiest way of controlling your remote power switch is to keep your (already paired) remote control close to the MultiTRX and copy the string received, eg. *$GABC2F0.* This way you can switch on and off this device by sending *$GABC2F1* and *$GABC2F0* , respectively.

## Gao-specific

If you want to pair a GAO device with the MultiTRX directly, send the string above, but replace the power on/off parameter with the number two: *$GABC2F2.*

## Nexa-specific

Nexa dimmers has the great feature of being able to dim to an absolute level in 16 steps. To do so, replace the power on/off parameter with the number two, and add an extra hexadecimal character (upper case) with the dim level: *$N1234ABC030 -> $N1234ABC032F* (will set the device to maximum brightness).

# Complete command list

All communication is done at 8N1 9600 baud, ASCII-formatted data. Data from the board is terminated with LF (ASCII character 10). Data sent to the board does not need any kind of termination, as the commands are fixed length. All hexadecimal characters must be in upper case!

## Commands transmitted from MultiTRX (RF-reception)

|  |  |  |
| --- | --- | --- |
| Output | Description | Example(s) |
| $Dddd | Doro Secur sensor. d = dip switch settings (hex). | $DA07 |
| $Eh,c,t,H | Esic temperature sensor. h = house code,  c = channel, t = temperature (degrees Celsius),  H = relative humidity (percent). | $E5,1,12.3,71 |
| $Giiiiip | GAO remote power switch. i = unique ID (hex),  p = power setting (0/1) and pairing mode (2). | $GABC2F0 $GABC2F1 |
| $Niiiiiiigcp | Nexa remote power switch. i = unique ID (hex), g = group enable (0/1), c = channel (hex), p = power setting (0/1). Packets with dim level will not be received! | $N1234ABC010 $N1234ABC011 |

## Commands transmitted to the MultiTRX (RF-transmission)

|  |  |  |
| --- | --- | --- |
| Input | Description | Example(s) |
| $GiiicCp | GAO remote power switch. i = unique ID (hex), c = channel, C = checksum, p = power setting (0/1) and pairing mode (2) | *$GABC2F0 $GABC2F1* |
| $Niiiiiiigcp(l) | Nexa remote power switch. i = unique ID (hex), g = group enable (0/1), c = channel (hex),  p = power setting (0/1, absolute dim level = 2), l = dim level (hex). Dim level only applies if p = 2, so the length of this string varies. | $N1234ABC010 $N1234ABC011 $N1234ABC012F |
| $Whuup | Waveman remote power switch. h = house code (A-P), uu = unit code (01-16), p = power setting (0/1) | $WA011 |

## Other commands transmitted to the MultiTRX

|  |  |  |
| --- | --- | --- |
| Input | Description | Example(s) |
| $P | The MultiTRX will respond with version number | *-* |
| $Sniiiiiiii | Save unique IDs in the EEPROM ID-memory to avoid the sabotage control in the MultiTRX. n = EEPROM location (1-8), i = unique ID of remote control. The ID-field must be eight digits of length, so the ID might need to be padded with zeros! | $S11234ABC0 (ID = 1234ABC) $S212300000 (ID = 123) |
| $R | Read the ID-memory | $R12345678,ABC12340, 00000000,00000000... |
| $X | Force restart. Primarily used to upload new firmware with bootloader. |  |

# Compatible hardware

## Sensors

* All wireless Doro Secur sensors (available from Alarm-Agenturet)
* Esic temperature sensor (available from Clas Ohlson)

## Power switches

* Self learning Nexa (available from Clas Ohlson and Jula)
* Waveman (available from Clas Ohlson)
* GAO/CO-tech (available from Clas Ohlson)

# Hardware specification

|  |  |
| --- | --- |
| Supported devices (RX) | Esic temperature sensor, Doro Secur sensors and Gao and Nexa remote power switches |
| Supported devices (TX) | GAO, Nexa and Waveman remote power switches |
| Modulation | OOK / AM |
| Output power | 10 dBm |
| Receiver sensitivity | -112 dBm |
| Frequency stability | +/- 50kHz over -40 to 85 degrees Celsius |