PHANTOMTRONICS
MPX OVER IP = MicroMPX

General: MicroMPX User Manual
Reference: Release V.2.0

Dear Customer,

Thank you for buying this MicroMPX device.
MicroMPX is a revolutionary method to transport your broadcast processors multiplex (MPX) signal via LAN or WAN.
Particularly the lossy codec is specially made to transport this kind of complex signals.
The codec is developed by ThimeoSoft and in this case PhantomTronics has developed the hardware.

This user manual is intended to get you started with this system.

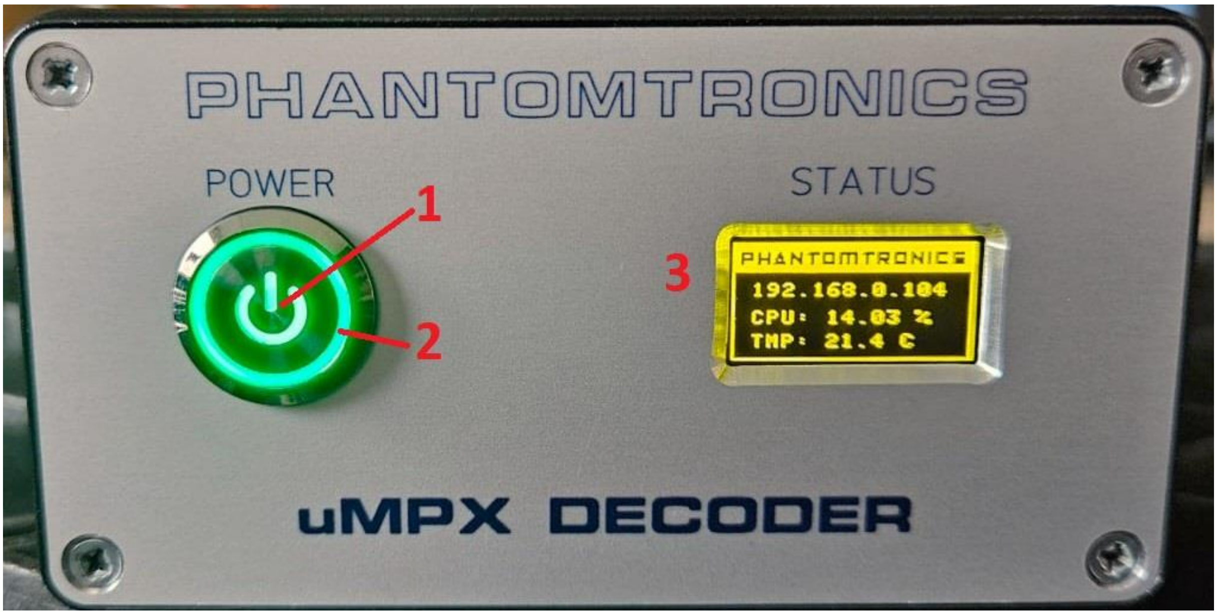
INDEX

1. Unpacking and connecting your MicroMPX device.

Package contents:

- Power supply 9VDC 2A	1x
- MicroMPX EN/Decoder	1x

Front panel



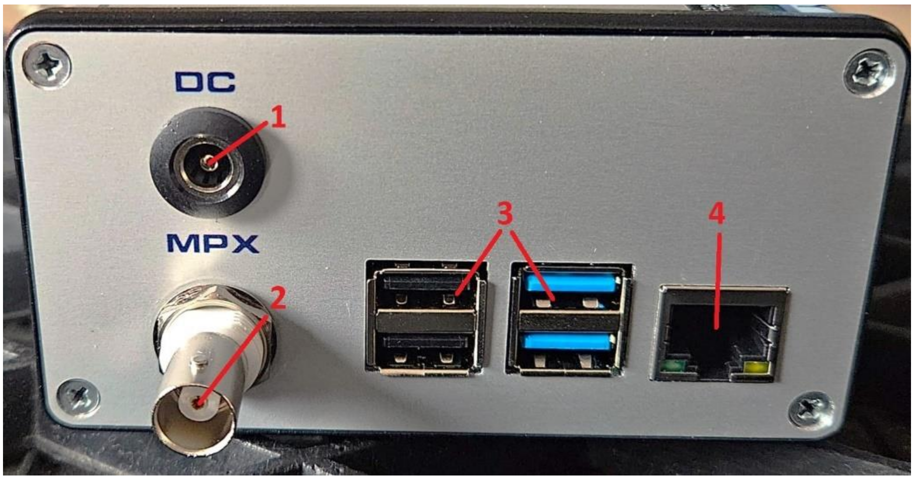
1. Start-up / Shutdown button
2. Power light
3. Status Display

Start-up / Shutdown button
When a power supply is connected to your MicroMPX device, it should start-up automatically.
Reason for this, is that in a rare case of power failures the device will restart automatically when the power comes back.
Shutting down; the device can be shut down by pressing the shutdown button [1], not by disconnecting the power source! *(this can result in a corrupt operating system)*
When the shutdown button [1] is pressed it might take 5-10 seconds for the system to be switch off.
Start-up; When the device is off; it can be started by pressing the Start-up button [1]

Power light
The power light is to indicate the unit is in *start-up, shut-down*, or *running mode*
When the device is fully shut-down the light will go off, and then power supply can be disconnected.

Status Display
The status display is to indicate the devices IP-Address, CPU load and CPU temperature.
When the device is not connected to a network, the display will show: *no IP-ADR*
CPU load should not exceed 50% Temperature should not exceed 70 degrees Celsius.

Back panel



1. Power connector
2. MPX connector
3. USB connector
4. LAN connector

Power connector
The MicroMPX device is build based on a ARM micro PC, and has a Power Supply management board inside.
This means the input voltage is wide range and can be between 6 and 24VDC.
A 9V 2A power supply comes standard with the PhantomTronics device.

MPX connector
The MPX connection is a BNC Balanced type connector.
MicroMPX Encoder: your sound processors analog MPX (Stereo+R(B)DS) signal should be connected.
MicroMPX Decoder: Analog MPX output, should be connected to your transmitters MPX input.

LAN connector (1 Gb/s)
The LAN RJ45 connection, is a gigabit connection and is used to send (Encoder) or receive (decoder) the uMPX Stream. It is also used for remote control of the device.

USB Connectors
MicroMPX Decoder: The USB Connectors can be used to connect a storage device with your backup MPX file, in case of a network failure the Decoder will play the stored MPX file, and when the network comes back it will automatically switch back to the MicroMPX composite stream.
MicroMPX Encoder: The USB Connectors are not used.

2. Default Ports and Passwords.

Default Ports:

• Default Port 8080	User (WEB) Interface
• Default Port 22	SSH Interface
• Default Port 8854	Default MicroMPX streaming port

Default ports can be changed, see Chapter 5,6 and 7

MicroMPX user interface.
The user interface is based on HTML5 and can be accessed by typing your devices IP-address in a web browser. *e.g. http://192.168.0.100:8080*

MicroMPX SSH Interface.
SSH connection is used for advanced settings, such as setting a fixed IP-Address, add IP-addresses to the whitelist or connecting to the Power management board.

MicroMPX Stream port.
This is the port used by the MicroMPX Application for the digitalised MPX stream.

Default SSH login and password:

When connecting to the device via SSH port 22;
Default username: thimeo
Default Password: thimeo

Default password can be changed; see chapter 7

3. Preparations before setting up your MicroMPX stream.

Below is only applicable when devices are on different LAN's
If both the Encoder and Decoder are on the same network you can skip port forwarding.

In order to get your MicroMPX stream running you need to do some preparations.
Depending on your needs you need to forward the ports you would like use at the **Decoder** site.
So port forwarding has to be done at the transmitter sites (Decoders) and at least one port has to be port forwarded (MPX stream port)

If you would like to control your MicroMPX device remotely, you need to forward the WEB interface port.

If you would like to be able to remote update and reboot your device, you need to port forward the SSH port.

Example:
*Your **Encoder** is located in your studio, and your pc is on the same LAN, you can access your encoder via the PC. Your LAN is connected to internet via a router. All functionality's can be used.*

*Your **Decoder** is at a transmitter site on a mountain and is connected to internet via a router at the transmitter site.*
The Decoder's LAN address: 192.168.0.100
The WAN address of the transmitter site is: 98.765.43.21
You would like to have full control of the decoder;
MPX stream at port 8700, User interface (WEB) at port 8701, SSH connection at port 8702

In the router at the transmitter site, you need to add the following:

MPX Stream:	98.765.43.21 : 8700 → 192.168.0.100 : 8700
User Interface (WEB):	98.765.43.21 : 8701 → 192.168.0.100 : 8080
SSH control:	98.765.43.21 : 8702 → 192.168.0.100 : 22

See also: https://www.noip.com/support/knowledgebase/general-port-forwarding-guide/

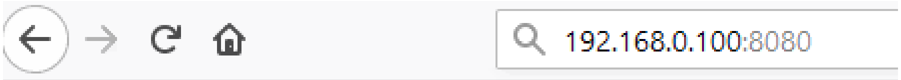
Now you will have to configure your Encoder and Decoder setting via the WEB interface.
For this see chapter 5. [ENCODER] or 6. [DECODER]

PLEASE NOTE! While it may be possible to use the public internet with MicroMPX, there are no guarantees that performance, bandwidth or up-time will be adequate for your needs.
Low budget outfits that have few or no STL choices may consider the internet to be perfectly adequate. Major operators with real revenue on the line may consider the Internet a last resort backup link only. Results will vary. The risk is on you

4. Controlling your device.

As mentioned above your device can be controlled several ways.
The main user interface is done by a WEB browser connecting to the build in webserver.
Most browsers are compatible, as long as it supports HTML5; Mozilla firefox is recommended.

Enter the MicroMPX device IP-Address in the Address line of your browser followed by the webservers port number [default 8080]



192.168.0.100:8080

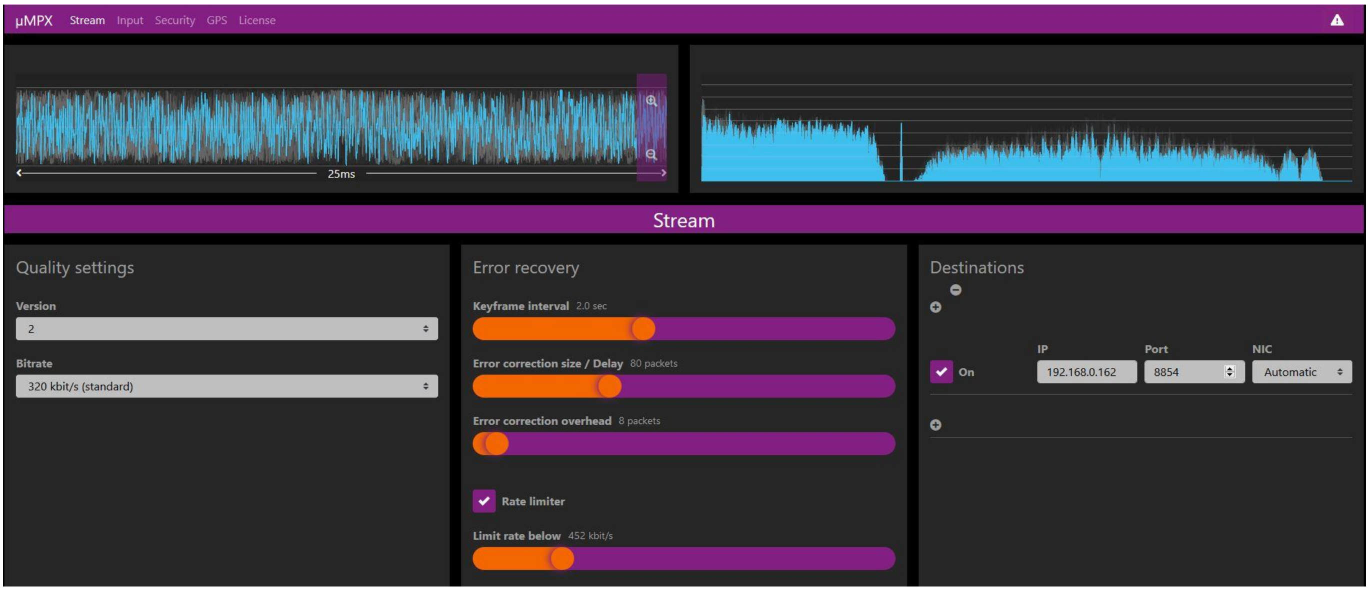
5. MicroMPX Encoder settings.

Basic settings encoder to get you started:	
Bitrate	384 Kbit/s
Keyframe Int	4 sec.
Span	90
Overhead	10
Rate limiter	ON → 500 Kbit/s

When connecting to MicroMPX Encoders web interface you will see fig 1.

Fig 1. Main page

Now Click on the Stream tab.
You will see something like fig 2



Stream

Quality settings

Version: 2

Bitrate: 320 kbit/s (standard)

Error recovery

Keyframe interval 2.0 sec

Error correction size / Delay 80 packets

Error correction overhead 8 packets

☑ Rate limiter

Limit rate below 452 kbit/s

Destinations

	IP	Port	NIC
☑ On	192.168.0.162	8854	Automatic

Fig 2. Stream

Quality settings

Settings that control the MicroMPX audio quality.

- Version -- The type of codec used by the Encoder; 1-3 (Evolution of the codeq)
- Bitrate -- The bitrate that MicroMPX uses.

320 Kbit/s is normally sufficient, but if Allow L-R spectrum asymmetry if louder (may deteriorate reception) is used, a higher value (384 Kbit/s) is recommended.

Error Recovery

Settings that control the MicroMPX error recovery overhead

- Keyframe interval -- Sets the time between keyframes.

If packets are lost (even after using Error correction), audio can only resume after reception of a key frame.
The value to be set is depending on the network stability, default setting is: 4 seconds.
If your network is unstable, you could decide to set it lower, this will result in more data but a quicker resume of audio.

- Error correction span (delay) -- The number of packets over which the Error correction overhead packets are send.

This is best explained using some numbers. Let's say that Span is set to 64 packets, and Error correction overhead is set to 4 packets. Then that means that for every 64 normal packets, 4 extra recovery packets will be sent. This causes 6.25% (4/64) overhead. With these settings, of the total of 68 (64 + 4) packets, any set of up to 4 packets can be recovered if they are lost.
With 32 and 2, which has the same amount of overhead, any set of 2 packets of the total 34 can be recovered if lost.

This may seem to be very similar, but packets often are dropped in burst, so typically 64/4 will recover far more lost packets than 32/2, and even 32/4 will only recover slightly more packets than 64/4, at double the overhead.

*The only problem with larger Span values is that they require **more delay at the decoder end**. If the first of 64 packets is lost, in order to recover it, at least one recovery packet is needed to recover it. And the recovery packets are sent after sending all the normal packets in the Span. So, with 96 packets per second, around 700mS of latency is needed to enable recovery for 64 packets. For 32 it's only half that.*

- Error correction overhead -- The number of recovery packets per Error correction span (delay) are sent.
- Rate limiter -- Limits how fast (recovery) packets can be send to the network.

Recovery packets can in some cases be a bit bigger than normal packets.
So a few Kbit/s overhead is recommended. If no overhead is available above the (calculated) total bitrate based on normal plus recovery packets, then any excess recovery packets are dropped.

- Limit rate below – Limits the bandwith

Many networks have limitations in how many packets they can process, if large numbers of Error correction overhead packets are sent in burst, some of them (plus some real packets that come afterwards) can get dropped, actually deteriorating reception on the other end.

Some people have asked for the possibility to send the same stream twice over the same connection, with a 1 second delay, as a means of redundancy.
Instead of that, you can set both Error correction span (delay) and Error correction overhead to 120 packets (the number of packets that are sent in about 1.5 seconds).
That way, any drop that takes less than 0.7 seconds will be fully recovered.

Note: The total of the two sliders may not exceed 255, that's why we can only go to 120

Destinations

All the target (decoder) addresses.

- ON - Enables this output line. [Tikked purple box = ON]
- IP - The IP address of a Decoder must be entered here.

The IP address must be reachable from encoder to decoder.
It is possible to send multiple streams to the same encoder for redundancy.
(See: Error correction panel yellow section above).

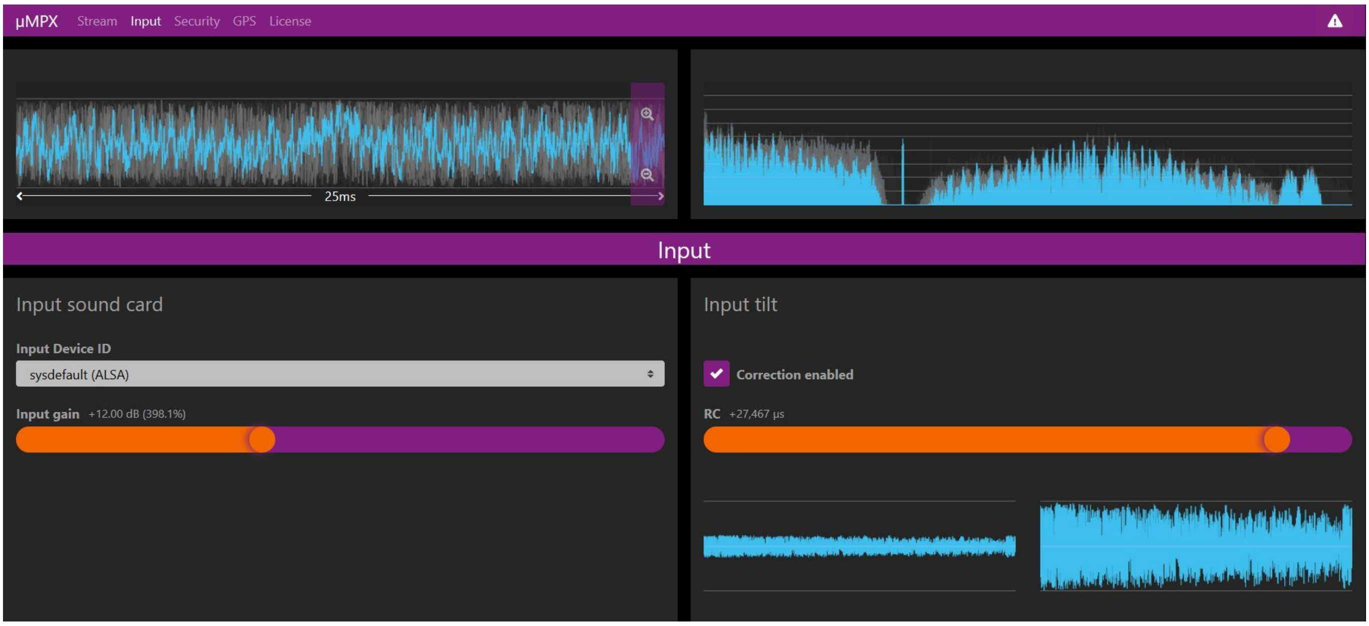
- PORT - The port number on which the digitalised MPX stream is being streamed

This port has to be reachable from encoder to decoder.
You will probably need to setup port forwarding in the router on the decoder site. (See: chapter 3 / p5)

- NIC - The network interface on the pc to send the data to.

This option is not used on the hardware versions of MicroMPX Encoder.
MicroMPX is also developed for use on PC, and in case of multiple network interfaces (NIC), then this option will let you choose on which NIC this particular configuration line is being streamed.

Now Click on the Input tab.
You will see something like fig 3.



Input sound card

Input Device ID
sysdefault (ALSA)

Input gain +12.00 dB (398.1%)

Input tilt

☑ Correction enabled

RC +27,467 μs

Fig 3. Input

Input Device ID -- Dropdown menu of the sound devise used for MPX input encoding.
This should be set at: ***sysdefault (ALSA)*** *or* ***(hw0.0)(ALSA)***

Input level correction -- *Slider to set the optimum streaming level.*
In order to get the best result, the MicroMPX stream should be set at a certain level, this optimum level will ensure the loudest signal with clipping control.
The error section will help you set the best level.

Tilt correction -- Slider to correct for soundcard low pass filtering.
This RC should be set at: ± 27 uS

Soundcards who do not have a flat frequency response (mostly caused by a low pass filter) can be compensated to get a flat frequency response.
Calibration can be done by connecting a function generator to the MPX input and apply a 300Hz square wave. In the WEB interface the square should have a flat top; if not, correct with the RC slider.

Now Click on the Security tab.
You will see something like fig 4.

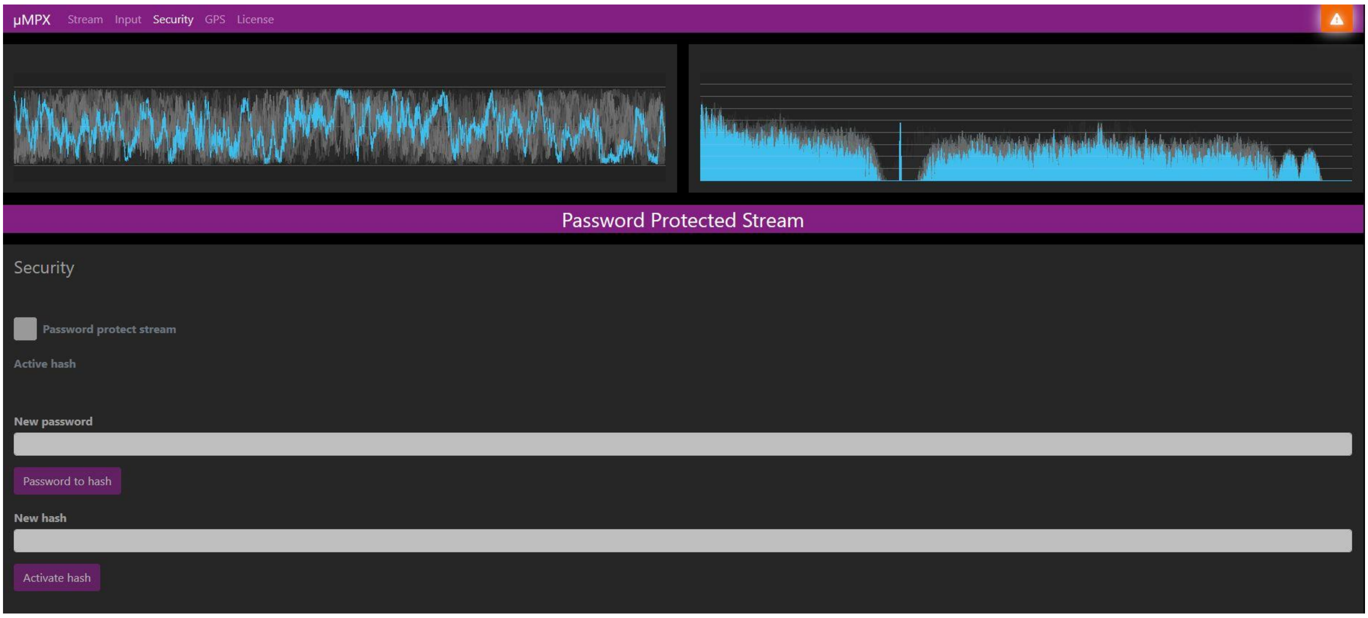


Fig 4. Security

Security panel --Settings that protect the MicroMPX stream against external interference.

Password protect stream --Encodes the stream to make it hard for external parties to take over the stream.
(We say "make it hard" because in the end, anything can be hacked).

Active hash -- Hash code used for encoding the stream.
Hash is a method of encrypting a data stream by splitting the stream and index the parts in a certain way.
Only the ones who know the index can decode it.
This code must be the same on the encoder and all decoders.

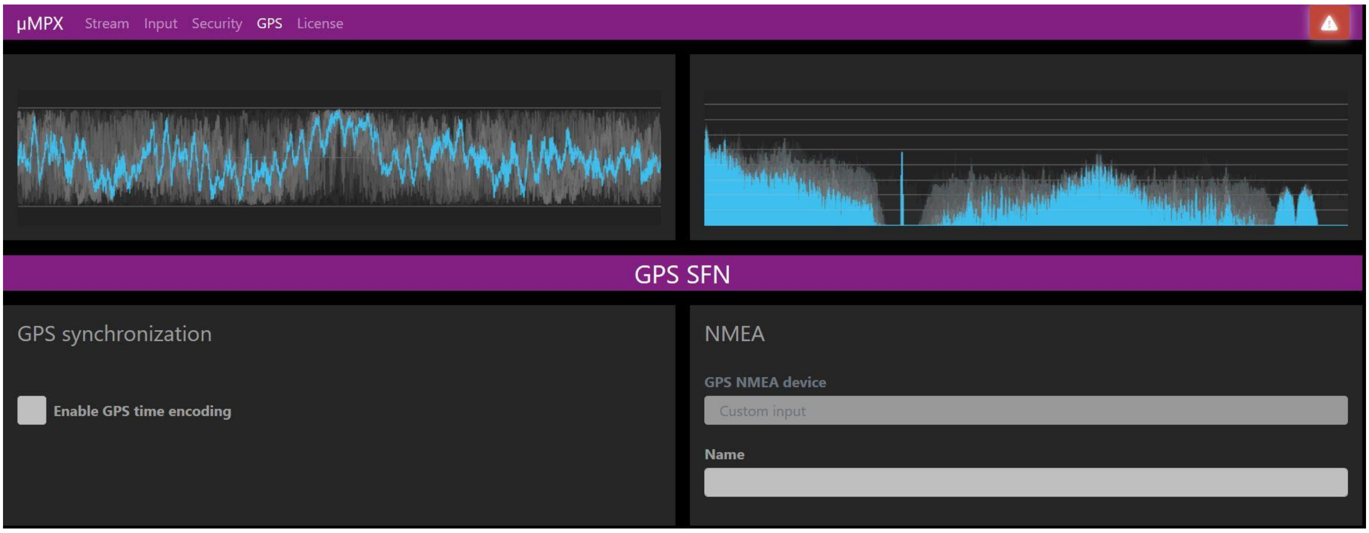
New password
Enables entering a password, which is transcoded to a hash code by means of the build in password to hash code converter. This action will be carried out when clicked on Password to hash

Password to hash
Converts the entered password to a 'hash' code.

New hash
You can enter a hash code here, to be put in Active hash.

Activate hash
Activate the hash code.

Now Click on the GPS tab.
You will see something like fig 5



GPS SFN

GPS synchronization

□ Enable GPS time encoding

NMEA

GPS NMEA device

Custom input

Name

Fig 5. GPS

GPS SFN – GPS input device for use with Single Frequency Networks.
MicroMPX has synchronisation between streams automatically, for this GPS SFN is not necessary.
So what can be done with GPS SFN ?

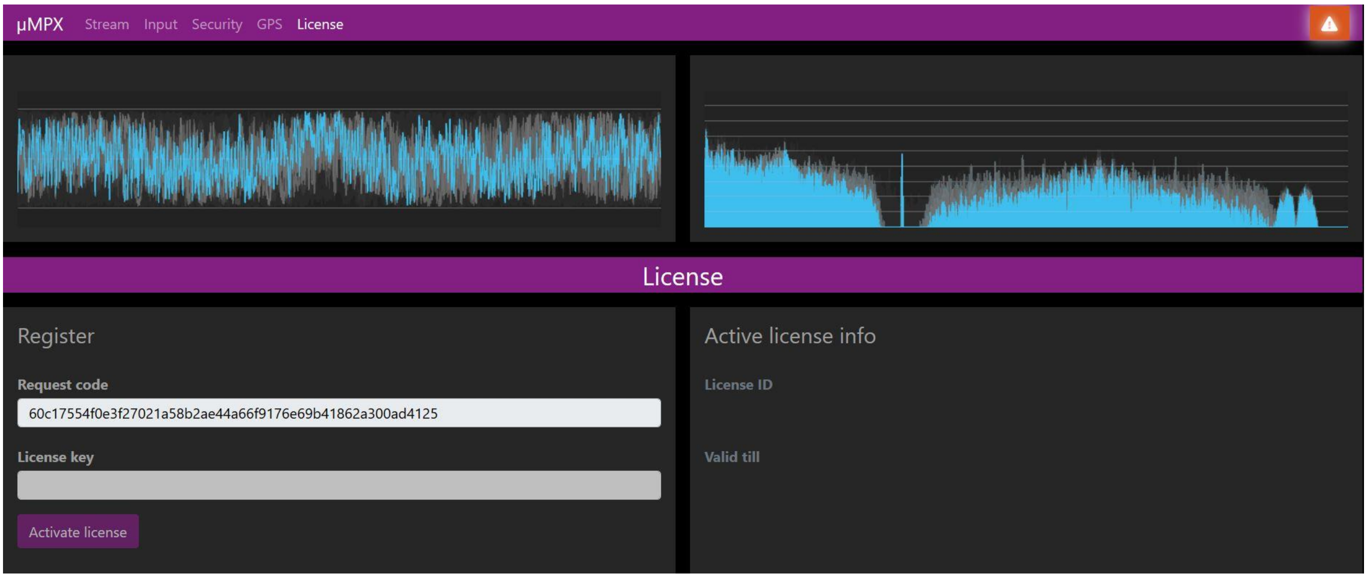
GPS SFN
Only applicable when:

1. *2 transmitters with the same program are on the same frequency &*
2. *If they have an overlap.*

** When above applies to your situation the optional GPS receiver is needed **

On FM in general this generates audio interference because of the runtime differences and jitter in the overlapping area, in this area the audio timing is critical. So with GPS SFN you can adjust the audio phase in order to adjust the area of interference.
The optional GPS is used to synchronise the MicroMPX device with the 1pps, in this way you can accurately set audio delay in micro-second precision.

Now Click on the LICENSE tab.
You will see something like fig 6



License

Register

Request code

60c17554f0e3f27021a58b2ae44a66f9176e69b41862a300ad4125

License key

Activate license

Active license info

License ID

Valid till

Fig 6. License

License – Field to import your purchased licence code.
MicroMPX hardware does NOT come with a software license.
Software has been developed by Thimeo Audio Technology and a licence can be purchased here:

https://www.stereotool.com/products

No license limitations
MicroMPX can be used without a license, then occasionally a beep or audio tone sweep will be generated.
Time between the sweeps will be random.
This audio sweep or tone will not exceed the set deviation level.

6. MicroMPX Decoder settings.

Basic settings encoder to get you started:	
Port	Use port as configured in encoder.
Delay	5 sec.
Security	Use hash as configured in encoder.

When connecting to MicroMPX Decoders web interface you will see fig 7.

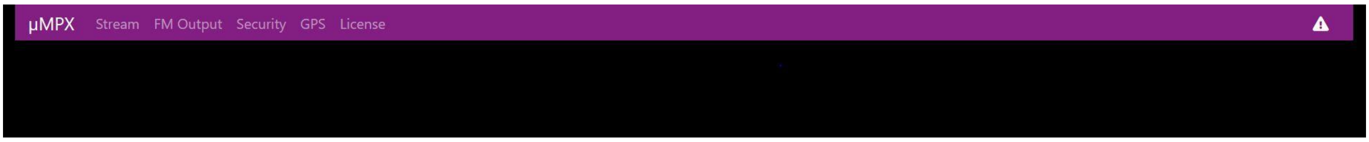
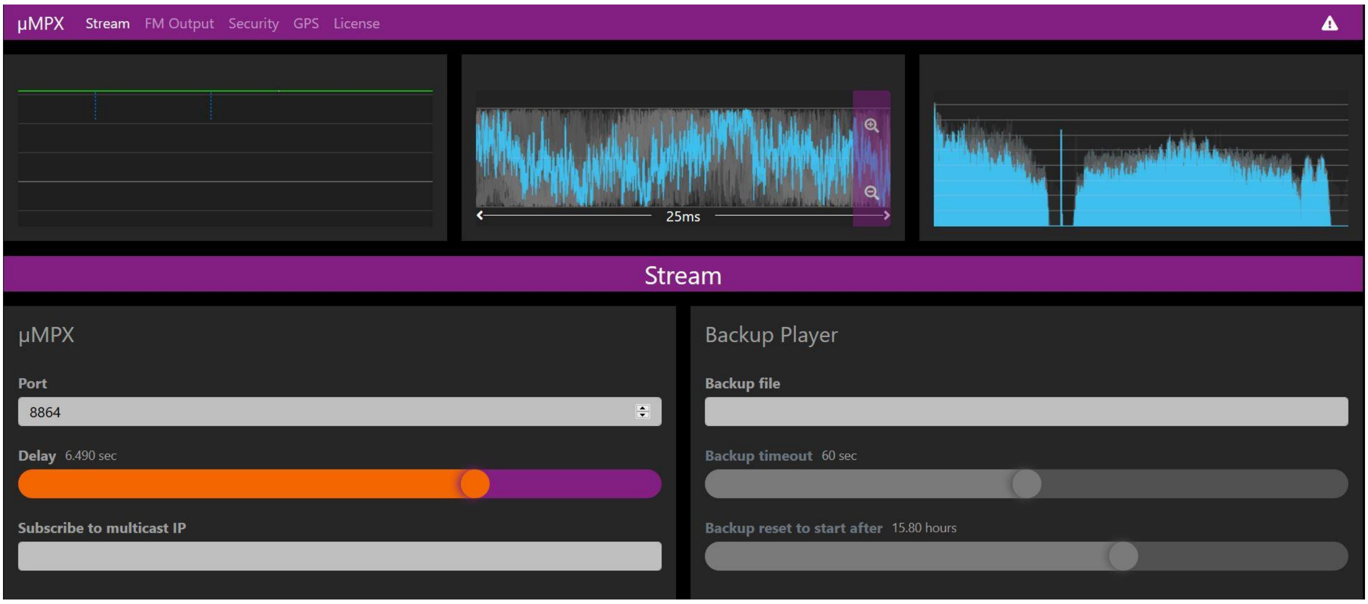


Fig 7. uMPX Decoder: Main page

- At the top you can choose which settings you would like to see or edit .
- The triangle icon will take you to the error log.

Now Click on the Stream tab.
You will see something like fig 8.



µMPX

Port
8864

Delay 6.490 sec

Subscribe to multicast IP

Backup Player

Backup file

Backup timeout 60 sec

Backup reset to start after 15.80 hours

Fig 8. uMPX Decoder: Stream

Stream -- Settings that configures the MicroMPX stream

- *Port -- MPX streaming port; should be the same as configured in the Encoder.*
- *Delay -- Delay time [S] in which the decoder has time to reconstruct lost packets.*
  Note: Delay in all Decoders should be the same to ensure AF [RDS] synchronised audio.
- *Subscribe to multicast IP – Used when the MPX stream is configured as multicast stream*

Backup player - a Build-in Silence detector; plays a pre-recorded MPX file in case the MPX stream fails.

- *Backup file -- Recorded WAV or FLAC file which contains MPX recording.*
  By default the USB ports are checked at startup, so when inserting an USB device containing the MPX recording a reboot of the MPX Decoder should be done.
  MPX files can be recorded using Thimeo Watchcat or MPXtool (available through optimizedmedia.net)

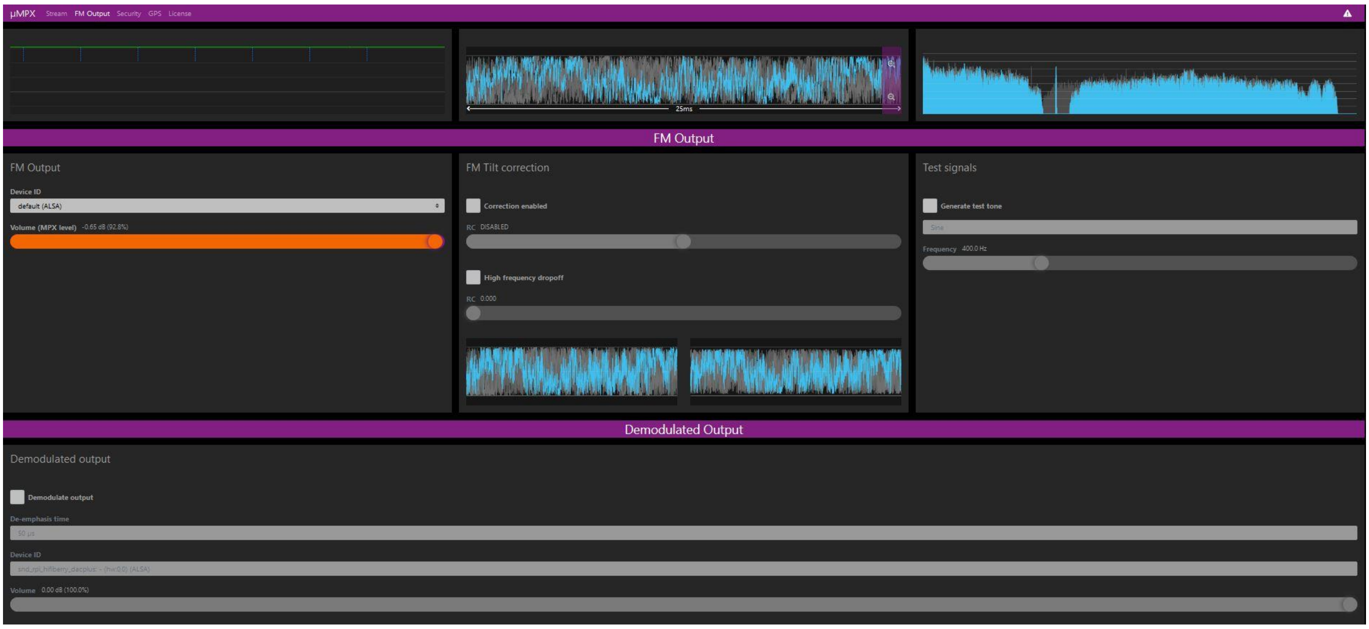
  Example:
  You have recorded a MPX file called 'backup.flac' and you have put it in the ROOT directory of an USB stick. Now your Backup file line should look something like this:

  /media/usb0/backup.flac

  Since your MicroMPX decoder has multiple USB ports usb0 can also be replaced by usb1.....8

- *Backup timeout -- Time before starting the recorded MPX file in case of MPX stream loss.*
- *Backup reset to start after -- Time in which the recorded file will start again from the beginning*

Now Click on the SOUNDCARDS tab.
You will see something like fig 8.



FM Output

Device ID: default (ALSA)

Volume (MPX level) -0.65 dB (92.8%)

FM Tilt correction

□ Correction enabled

RC DISABLED

□ High frequency dropoff

RC 0.000

Test signals

□ Generate test tone

Sine

Frequency 400.0 Hz

Demodulated output

□ Demodulate output

De-emphasis time: 50 µs

Device ID: snd_rpi_hifiberry_dacplus: - (hw:0,0) (ALSA)

Volume 0.00 dB (100.0%)

Fig 8. uMPX Decoder: Soundcards

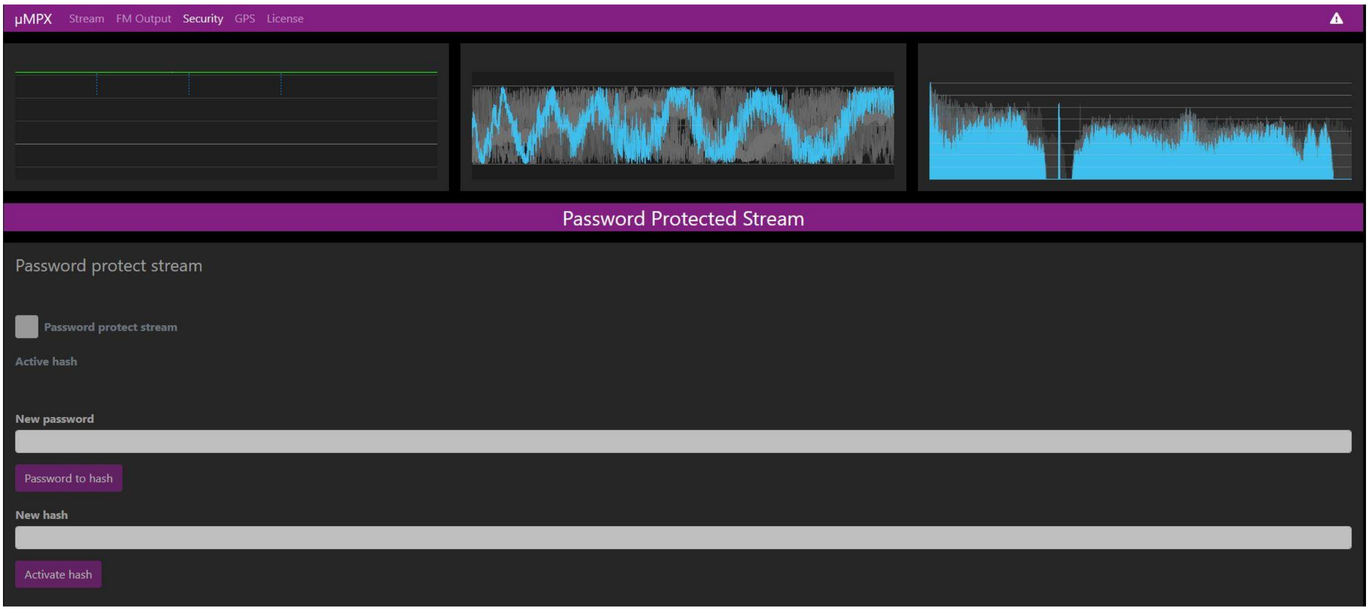
FM output -- Dropdown menu of the sound devise used for MPX output.
This is fixed by the software and depends on you MicroMPX device.

Volume -- Slider to set the MPX output level.
Set this slider to your country's deviation regulations.

FM Tilt correction -- Slider to correct for soundcard low pass filtering.
This RC should be set at: 0

> *Soundcards who do not have a flat frequency response (mostly caused by a low pass filter) can be compensated to get a flat frequency response.*
> *Calibration can be done by connecting a oscilloscope to the MPX output and set a 300Hz square wave via the build in function generator. [TEST SIGNAL]*
> *On the oscilloscope the square should have a flat top; if not, correct with the RC slider.*

Now Click on the SECURITY tab.
You will see something like fig 9.



Password Protected Stream

Password protect stream

□ Password protect stream

Active hash

New password

Password to hash

New hash

Activate hash

Fig 9. uMPX Decoder: Security

Password protect stream
Tick the ‘Password protected stream’ box when security is enabled. (see encoder)

Decodes the stream by use of a cryptographic method, if a hash is used at the Encoder end, make sure the Decoder uses the same hash

Active hash
Hash code used by the encoder for encoding the stream, you need the same hash in the Decoder. (see Encoder section for more information regarding hash encryption)

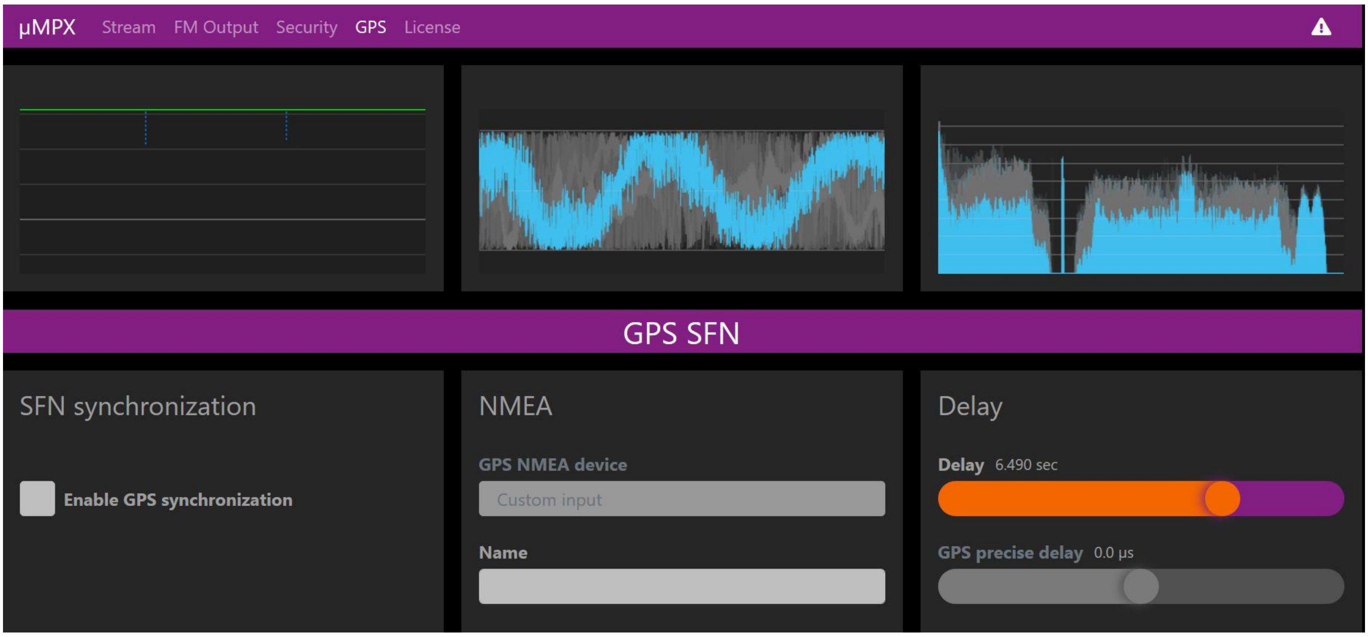
New password
Enables entering a password, which is transcoded to a hash code by means of the build in password to hash code converter. This action will be carried out when clicked on Password to hash

Password to hash
Converts the entered password to a ‘hash’ code.

New hash
You can enter a hash code here, to be put in Active hash.

Activate hash
Activate the hash code.

Now Click on the GPS tab.
You will see something like fig 10.



GPS SFN

SFN synchronization

□ Enable GPS synchronization

NMEA

GPS NMEA device: Custom input

Name

Delay

Delay 6.490 sec

GPS precise delay 0.0 µs

Fig 10. uMPX Decoder: License

<u>GPS SFN</u> – GPS input device for use with Single Frequency Networks.
MicroMPX has synchronisation between streams automatically, for this GPS SFN is not necessary.
So what can be done with GPS SFN ?

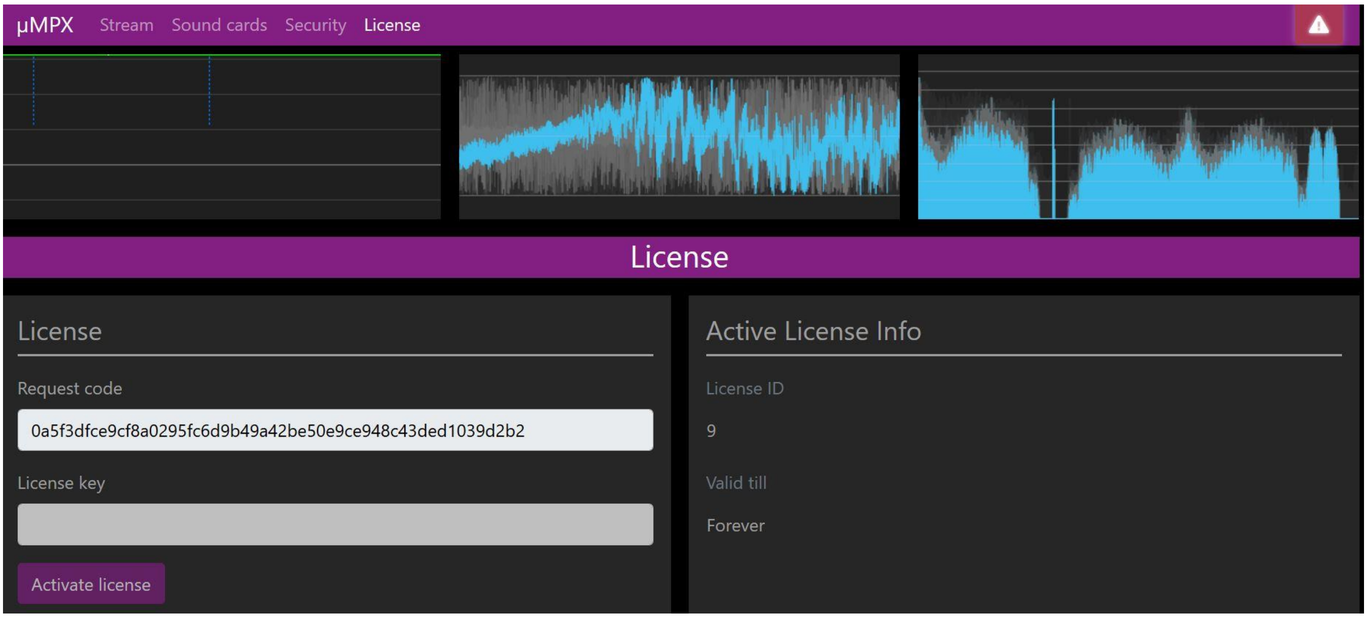
GPS SFN
<u>Only applicable when:</u>

1. *2 transmitters with the <u>same program</u> are on the <u>same frequency</u> &*
2. *If they have an <u>overlap</u>.*

If above is not the case, you don’t need the optional GPS receiver.

On FM in general this generates audio interference because of the runtime differences and jitter in the overlapping area, in this area the audio timing is critical. So with GPS SFN you can adjust the audio phase in order to adjust the area of interference.
The optional GPS is used to synchronise the MicroMPX device with the 1pps, in this way you can accurately set audio delay in micro-second precision.

Now Click on the LICENSE tab.
You will see something like fig 11.



License

License

Request code

0a5f3dfce9cf8a0295fc6d9b49a42be50e9ce948c43ded1039d2b2

License key

Active License Info

License ID

9

Valid till

Forever

Fig 11. uMPX Decoder: License

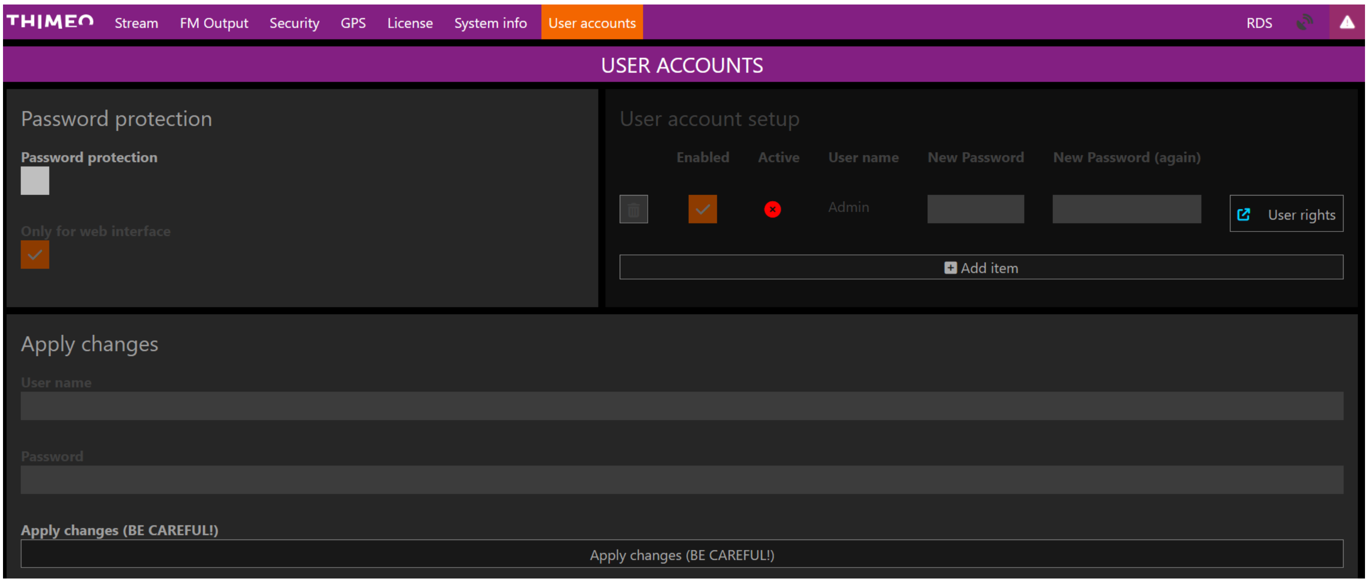
License

MicroMPX hardware does NOT come with a software license.
Software has been developed by Thimeo Audio Technology and a licence can be purchased here:

https://www.thimeo.com/micrompx/buy/

No License limitations
MicroMPX can be used without a license, then occasionally a beep or audio tone sweep will be generated.
Time between the sweeps will be random.
This audio sweep or tone will not exceed the set deviation level.

Now Click on the User Account tab.
You will see something like fig 12.



USER ACCOUNTS

Password protection

Password protection
□

Only for web interface
☑

User account setup

	Enabled	Active	User name	New Password	New Password (again)	
	☑	●	Admin			User rights

Apply changes

User name

Password

Apply changes (BE CAREFUL!)

Fig 12. uMPX Decoder: User Accounts

In this tab web interface user accounts and their rights can be configured.
When an ADMIN account has been configured web access is restricted, and in order to be able to make changes to the device you have to be logged in to an account with the appropriate rights.

Password protection
This checkbox enables the password protection for the GUI and enlightens the User account setup section.

Only for web interface
Needs to be checked.

User account setup
Default user name "Admin" is always used, and has all rights, please set an Admin password.
Please fill in you Admin password below:

Admin password: …………………………………………………………..

Note1: Whenever an user account is setup you will have to Apply Changes by using the Admin account details and click "Apply changes (BE CAREFULL!)" bar.

Note2: Use the 'Thimeo' logo on the left to refresh the page.

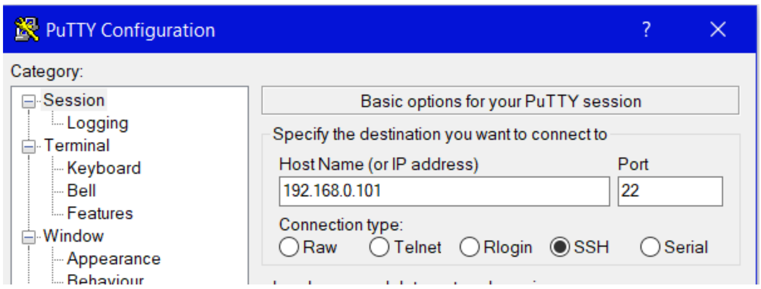
7. Apendix A.

In this case we are using a free software program called PuTTY to remote configure all settings below.
PuTTY can be downloaded via: http://www.putty.org

- **Set the whitelist.**
  By default all IP-addresses can access the MicroMPX WEB interface, so everybody can change settings and levels. It should probably be a good idea to alow web interface access to some known destination IP-addresses. This can be accomplished by using the whitelist.

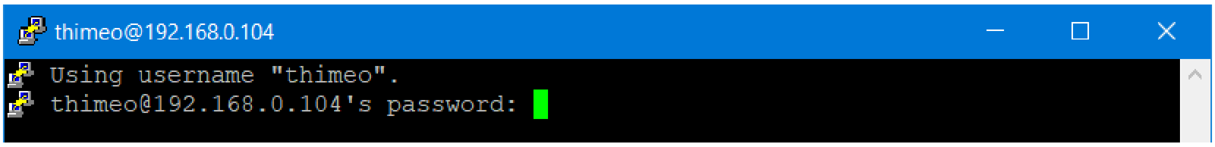
The whitelist option can be found in the *.MicroMPX_Decoder.rc* file on the MicroMPX devices storage.
** When changed, store all changes by: [CTRL]+[X] and [Y] and [ENTER] followed by *sudo reboot* **

1. Open the PuTTY program and set the ip adres, port is 22 by default, now click [OPEN]



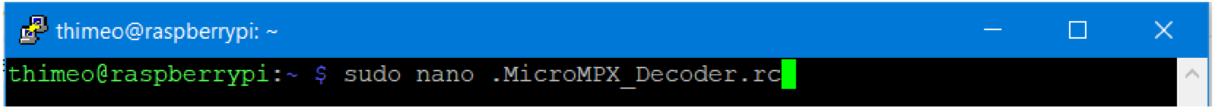
Host Name (or IP address): 192.168.0.101
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

2. login as: *thimeo [ENTER]* and use your password (default password: thimeo) [ENTER]



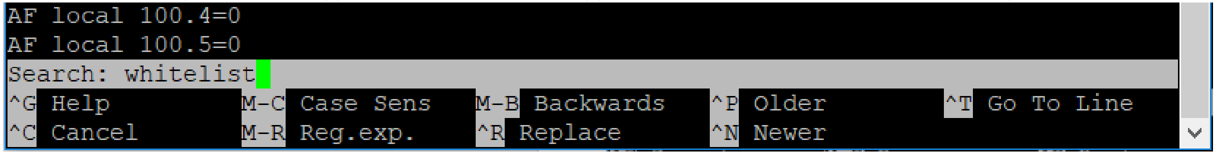
```
Using username "thimeo".
thimeo@192.168.0.104's password:
```

3. type: *sudo nano .MicroMPX_Decoder.rc* [ENTER]



```
thimeo@raspberrypi:~ $ sudo nano .MicroMPX_Decoder.rc
```

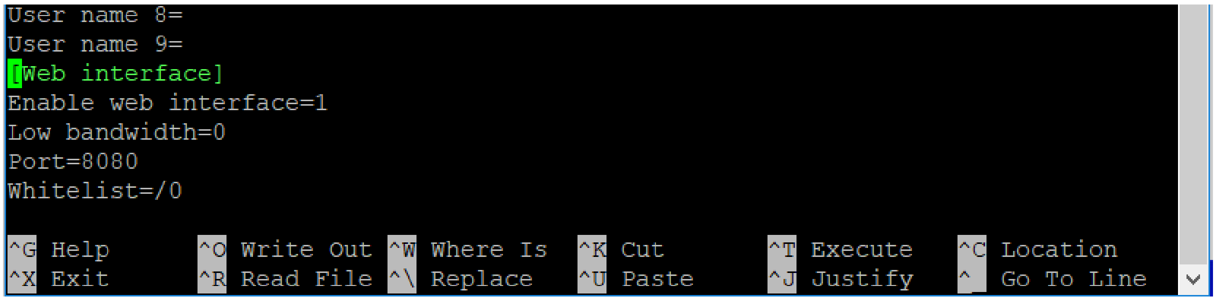
4. A file opens, now type: *[CTRL]+[W]* and type: *whitelist* in the search bar and press [ENTER]



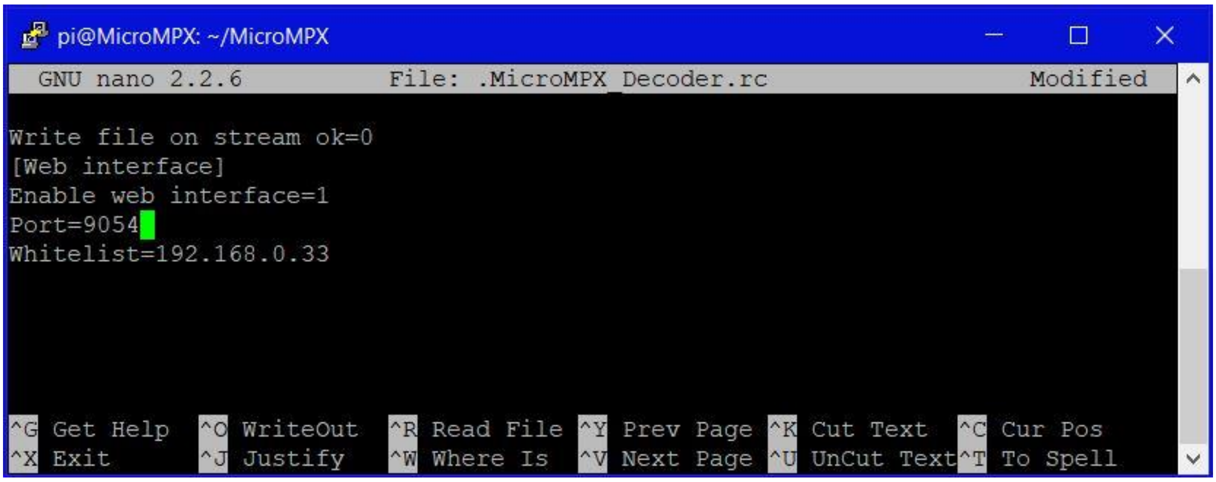
```
AF local 100.4=0
AF local 100.5=0
Search: whitelist
^G Help       M-C Case Sens   M-B Backwards   ^P Older        ^T Go To Line
^C Cancel     M-R Reg.exp.    ^R Replace      ^N Newer
```

5. Use arrowkeys to navigate to the Whitelist line.
   Change /0 to the IP-Address who is allowed to have access to the webinterface.

** When changed, store all changes by: [CTRL]+[X] and [Y] and [ENTER] followed by *sudo reboot* **



```
User name 8=
User name 9=
[Web interface]
Enable web interface=1
Low bandwidth=0
Port=8080
Whitelist=/0

^G Help       ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^  Go To Line
```



```
pi@MicroMPX: ~/MicroMPX

  GNU nano 2.2.6            File: .MicroMPX_Decoder.rc                 Modified

Write file on stream ok=0
[Web interface]
Enable web interface=1
Port=9054
Whitelist=192.168.0.33






^G Get Help   ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

- **Change WEB interface port number.**
  By default MicroMPX has its WEB interface port fixed at port 8080.
  In case you would like to change this port, please set the correct port number in the same file mentioned in above section.

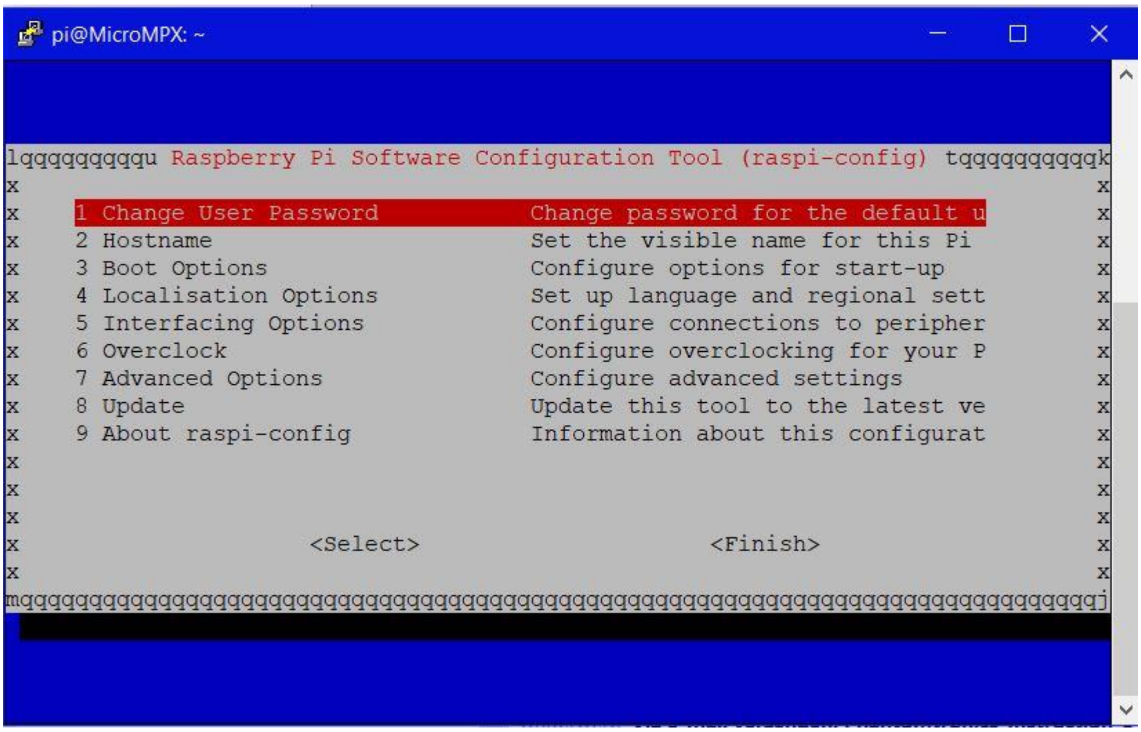
  ** When changed, store all changes by: [CTRL]+[X] and [Y] and [ENTER] followed by *sudo reboot* **

- **SET fixed IP in your MicroMPX device.**
  By default your MicroMPX device has DHCP enabled, so it gets an IP address from the DHCP server of the network it is connected to.
  Best thing to do is to configure your DHCP server (most of the time your router) to always give the same ip address to your device, this is done by entering the MAC address of the MicroMPX device in your DHCP server. (see manual of your DHCP server / Router)

  If you would like to manually set the IP address of the MicroMPX device please follow the steps below:
  1. Login to your MicroMPX device using the steps 1 and 2 (page 19)
  2. Type: *sudo nano /etc/dhcpcd.conf*
  3. A file is opened, now use arrow keys to scroll to : *interface eth0* ;remove the #
  4. Set the following: *static ip_address=192.168.0.100 [set the IP-address you want]* ;remove the #

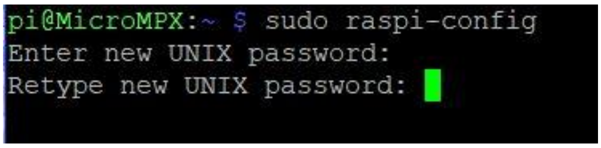
** When changed, store all changes by: [CTRL]+[X] and [Y] and [ENTER] followed by *sudo reboot* **

- **Change SSH password.**
  By default the MicroMPX SSH Password is*: thimeo*
  In case you would like to change this, please connect to the MicroMPX device
  (see Restriction of WEB access)
  When connected, enter the following: *sudo raspi-config [enter]*



```
lqqqqqqqqqu Raspberry Pi Software Configuration Tool (raspi-config) tqqqqqqqqqqk
x                                                                              x
x    1 Change User Password          Change password for the default u        x
x    2 Hostname                      Set the visible name for this Pi         x
x    3 Boot Options                  Configure options for start-up           x
x    4 Localisation Options          Set up language and regional sett        x
x    5 Interfacing Options           Configure connections to peripher        x
x    6 Overclock                     Configure overclocking for your P        x
x    7 Advanced Options              Configure advanced settings              x
x    8 Update                        Update this tool to the latest ve        x
x    9 About raspi-config            Information about this configurat        x
x                                                                              x
x                                                                              x
x                                                                              x
x                     <Select>                     <Finish>                    x
x                                                                              x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Now the configuration menu will open; use arrow keys to select 1 and press [enter];
Now select OK and press [enter]



```
pi@MicroMPX:~ $ sudo raspi-config
Enter new UNIX password:
Retype new UNIX password:
```

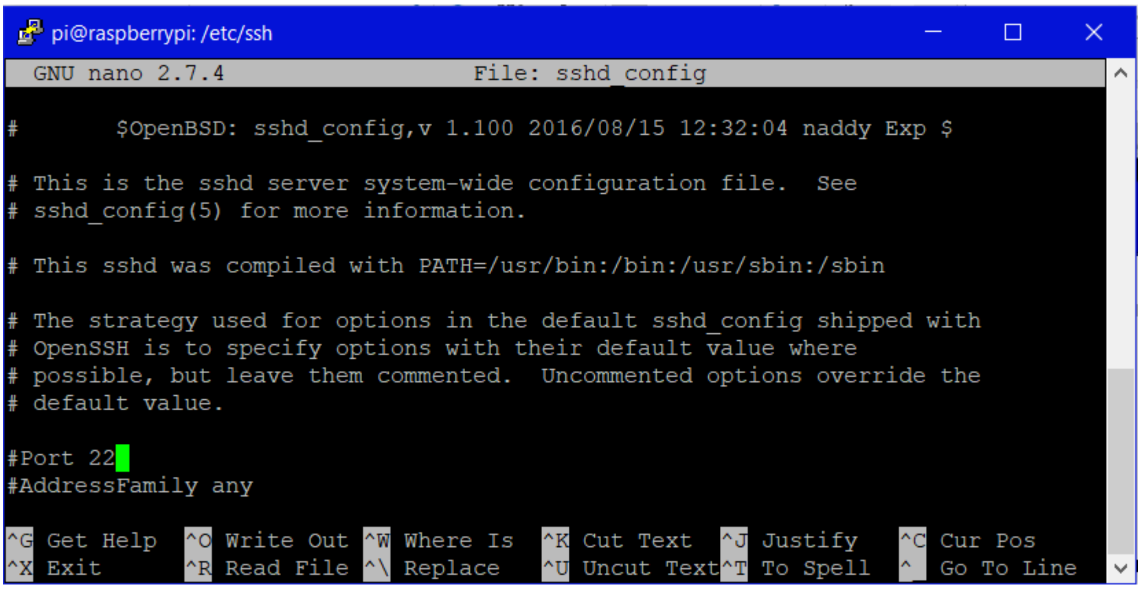
Now type your new password and press [enter] 2x
Your password has now been changed.

ENTER YOUR NEW PASSWORD HERE:____

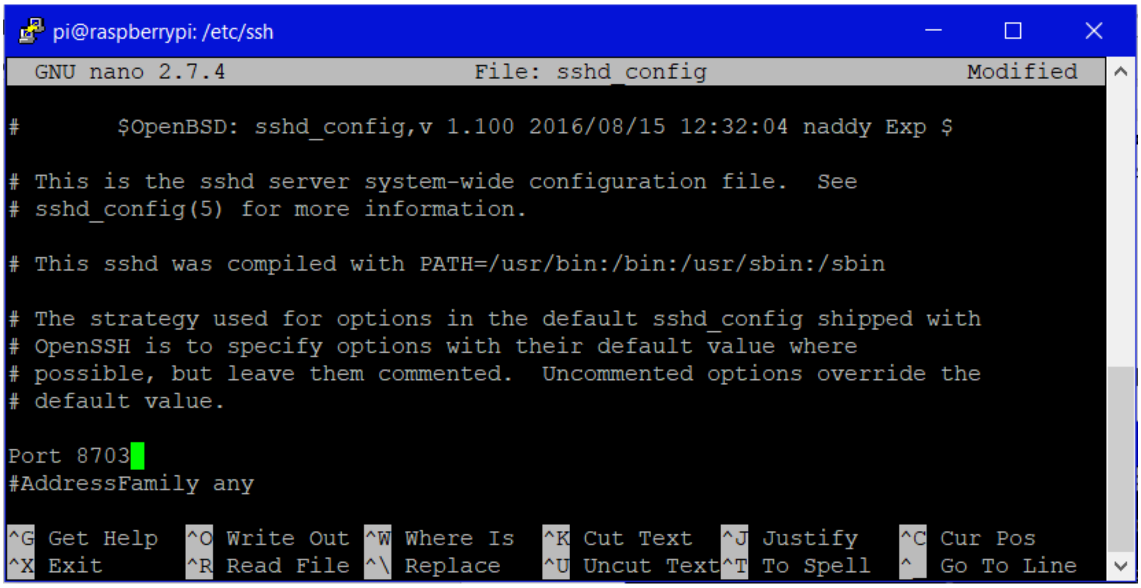
- **Change SSH port number.**
  By default the SSH port number is set at 22.
  It could be useful to change this, since it’s a common used port number for SSH.
  Remember to configure port forwarding in your router to remain access !
  Login to your MicroMPX device using the steps 1 and 2 (page 19)
  When connected, enter the following:

  cd /etc/ssh/ [enter]
  sudo nano sshd_config [enter]

  Now use arrow keys to select and change the port number, remove the # at this line



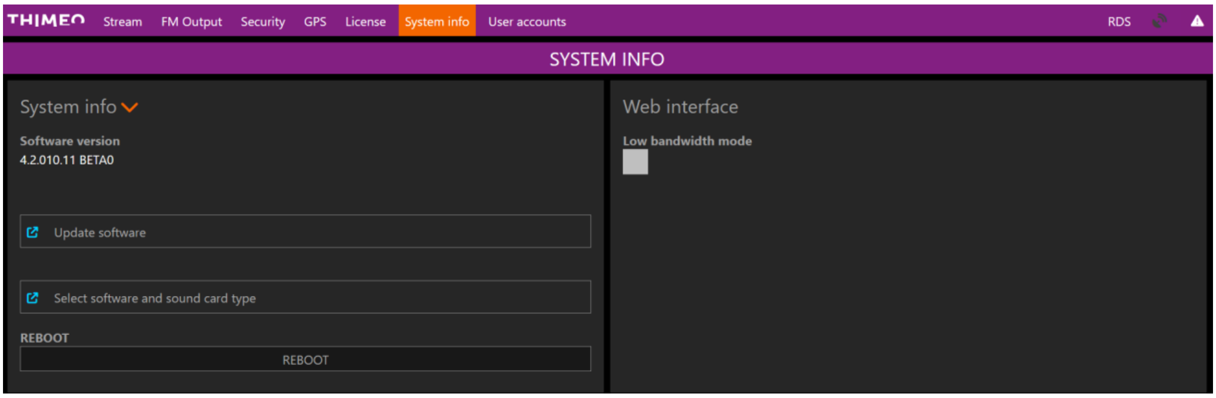
```
 GNU nano 2.7.4                  File: sshd_config

#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

#Port 22
#AddressFamily any

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```



```
 GNU nano 2.7.4                  File: sshd_config                  Modified

#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 8703
#AddressFamily any

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

** When changed, store all changes by: [CTRL]+[X] and [Y] and [ENTER] followed by *sudo reboot* **

- **Remote reboot your device.**
  There are 2 ways to reboot your MicroMPX device.
  1. Via it's web interface [Admin] and the 'system info' tab. [see image below]
  2. Via SSH:
     - Login to your MicroMPX device using steps 1 and 2
     - When connected, enter the following: *sudo reboot [enter]*



Software version
4.2.010.11 BETA0

Low bandwidth mode
□

8. Apendix B Trouble shooting.

- Make sure you use a correct PSU; this is often the cause of multiple undefinable faults.
  The correct PSU: 9Vdc – 2AMP
- Timing; faults can also occur when timing on the devices are incorrect; you can check the time by logging in with PuTTY and type: *sudo date*
- A Wifi connection could be a problem, sometimes this can cause frequent dropouts in the MPX stream.
- Drop of packets can happen when your network is not stable enough, this should not be a direct problem, when it happens all the time please check your network stability.
- 'Robot' like modulation; this can happen in combination with Stereotool, please check soundcard buffer settings in stereotool.

9. Software Updating your MicroMPX Device

- Updating the MicroMPX software is not recommended when you have a stable setup.
  However, future new features will be available via software updates they can be downloaded via the thimeo website.

  https://download.thimeo.com/ThimeoPi64Updater_AllProducts.zip

- Download the Thimeo updater ZIP file, and login your MicroMPX device via it's web interface [Admin]
- Select the System info tab and click the update software button
- Browse to the downloaded ZIP file and click on the upload icon on the right
- The new image is now uploaded and your MicroMPX device will automatically reboot when the update is done.

Any comments ?
please sent mail to: info@MicroMPX.nl

Jeroen Platenkamp
MicroMPX.nl