

openSPOT User Manual



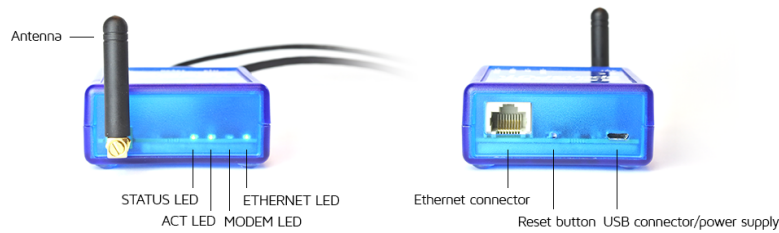
Note that this user manual is for the stable firmware version 0072. It's updated with new information when a new stable firmware is released.

Components

Accessories

The openSPOT package comes with an AC/USB power supply, an antenna, a USB and an Ethernet cable. These are all you need to operate openSPOT, however, you will need a web browser on a computer, tablet or mobile phone for initializing openSPOT settings, but after that, openSPOT can work standalone.

Interfaces



LED states

- The ETHERNET LED is wired to the Ethernet controller, so it's always green when an Ethernet cable is plugged in, and always blinking green, when it's communicating. It works even when openSPOT is running the bootloader.
- The STATUS LED shows openSPOT's connector status:
 - Continuously red: connector not set.
 - Blinking red: connector is trying to connect.
 - Continuously green: connector is connected.
- ACT LED states:
 - Slow blinking green: modem CPU found and is ready.
 - Fast blinking green: modem CPU communicating with main CPU.
 - Fast toggling between red and green: modem CPU firmware upgrade is in progress.
 - Slow blinking red: modem CPU version mismatch. Firmware upgrade needed.
 - Fast blinking red: modem CPU not found. Try to reboot openSPOT. If it doesn't help, try to upgrade firmware.
- MODEM LED states:
 - Unlit: modem is idle, or searching for sync.
 - Green: modem is receiving.
 - Red: modem is transmitting.

- Fast toggling between red and green: modem CPU firmware upgrade is in progress.

If the ACT LED is unlit, and both the STATUS and MODEM LEDs are blinking fast green, openSPOT is in bootloader mode. If main CPU firmware upgrade is in progress, the STATUS LED is toggling fast between red and green states, if modem CPU firmware upgrade is in progress, the MODEM LED is toggling fast between red and green states. If you reboot openSPOT and the ACT and STATUS LEDs are blinking red by turns, you've uploaded a firmware file with an incorrect hardware version. Retry firmware upgrade with the correct firmware upgrade file. For further information, refer to the *Firmware upgrade* section below.

Note that the ACT LED is also acting as a "heartbeat", if it's blinking very slowly that means the main CPU is very busy at the moment. It can happen when the web interface is loading, and openSPOT is in a call simultaneously. Calls always have higher priority, so the loading of the web interface can be slow during an ongoing call.

First steps

1. Screw on the antenna.
2. Connect the USB cable to a computer or to the AC/USB power supply.
3. Connect openSPOT to your local area network with the Ethernet cable.
4. Open <http://openspot.local> or <http://openspot/> in your browser. Note that you may have to enter openSPOT's IP address, if you can't open the page (for ex. on Android and iOS phones, which don't support MDNS or LLMNR – see the note below).
5. Log in with the default password *openspot* (all letters are lower case).
6. Follow instructions in the Connectors section below to set up connecting to a server/reflector.
7. If you want to use openSPOT with a DMR radio,

determine the best DMR demodulation mode for your transceiver by using DMR AutoCal. See it's usage in the Connectors section below.

On Android and iOS phones, you can use MDNS/Bonjour discovery applications to find openSPOT's IP address. Recommended apps for Android are [Zentri Discovery](#) or [ZeroConf Browser](#).

Here's a tutorial video on connecting openSPOT to the BrandMeister network:



The method is the same for other connectors.

Firmware upgrade

Be sure to always use the latest firmware on your openSPOT. For firmware upgrade files, see [this](#) page. Make sure that you use the correct firmware file for your openSPOT's hardware version. The hardware version is printed on the bottom of the device, it's in the model ID. For example, if you see SRF-OSP-1.1 there, you have hardware version 1.1. openSPOT uses a custom bootloader which shows up as a flash drive with our official USB ID (0x1fc9:0x8124).

Here's a tutorial video about upgrading the firmware:

openSPOT: How to upgrade the fi...



Entering bootloader mode

To start firmware upgrading, you'll have to connect openSPOT *to a computer* and *using the USB cable*, and you have to enter openSPOT into bootloader mode:

- The simplest way is to open the web interface, log in, go to the *Settings* page, and click on the *Reboot to bootloader* button.
- The other way is:
 - Disconnect openSPOT's USB cable, so it will be powered off.
 - Press it's reset button with a pen, and keep it pressed.
 - Plug openSPOT's USB cable in while keeping the reset button pressed. The ACT and STATUS red LEDs will be on while you keep pressing the reset button.
 - Release the reset button, if the ACT LED is off, and STATUS and MODEM green LEDs are blinking periodically, then openSPOT has successfully entered bootloader mode.

In bootloader mode, openSPOT will show up as a flash drive labelled BOOTLOADER on your

computer. There will be two 0 byte text files on this flash drive:

- SRF-OSP.TXT
- SRF-OSM.TXT

These two files signal that both the main and the modem CPUs are ready to start the firmware upgrade process.

Starting firmware upgrade

- Copy the firmware .bin file to the flash drive amongst the currently present text files.
- The STATUS LED will toggle between red and green states periodically. Firmware upgrade will last about 2 minutes. Do not power off, reboot or reset openSPOT or your computer until the upgrade process is finished!
- If the STATUS LED is constantly green for at least 5 seconds, the process is finished successfully, if it's constantly red for at least 5 seconds, the process is finished with errors. When the process is finished, openSPOT will reconnect it's flash drive.
- You can check the result of the firmware upgrade by looking at the flash drive. If you see SUCCESS.TXT, the process completed without errors. If there are no SUCCESS.TXT is present, please re-download the firmware file (it may got corrupted during the download), and retry the whole firmware upgrade process. If it fails after several times, take a note on what other files are present on the drive and write us a post in openSPOT [troubleshooting forum](#).
- Finally, you can reboot openSPOT, or just wait a minute, as the bootloader automatically reboots after 1 minute of inactivity.
- openSPOT will start in normal mode and upgrades the modem CPU automatically. This process lasts about 10 seconds, you can see the modem LED toggling between red and green states periodically.

- Now both the main and the modem CPU's firmware are upgraded. You can continue using openSPOT. See the new firmware's [changelog](#) for new features!

If you reboot openSPOT and the ACT and STATUS LEDs are blinking red by turns, you've uploaded a firmware file with an incorrect hardware version. Retry firmware upgrade with the correct firmware upgrade file.

Your browser may cache the previous version's web interface. Close and reopen your browser after a firmware upgrade, and press shift+F5 several times to clear the browser's cache.

Networking info

openSPOT's factory setting is to acquire an IP address using DHCP from the user's local router/DHCP server.

You can open the web interface by entering openSPOT's host name in your browser (you have to enter *http://* before the hostname, otherwise your browser will start a web search instead of opening the web interface). The factory default host name is *openspot*, that's why you have to open <http://openspot/> when opening the web interface for the first time. If it doesn't work, try opening <http://openspot.local>.

If none of the above works, you can always look at the DHCP clients list in your router's web interface to get openSPOT's IP address, which can be entered in your browser instantly. This may be the case if you want to open the web interface on Android and iOS phones, which don't support MDNS or LLMNR.

The default login password for the web interface is *openspot* (all letters are lower case).

You don't have to set any port forwards in your router for openSPOT. All used protocols are designed in a way that the initiate the connection, so your router will keep the used port open in both directions.

Resetting configuration

If you changed settings in a way you can't reach the web interface, you can restore factory default configuration by following the steps below. The procedure is the same as starting the bootloader, except that you have to press the reset button for longer time.

Here's a tutorial video on resetting the configuration:



Required steps:

- Power off the device.
- Keep the reset button pressed, and plug back the USB cable.
- Keep the reset button pressed for at least 10 seconds.
- After 10 seconds the STATUS and MODEM LEDs will flash quickly 5 times, signaling that default configuration has been restored.
- Release the reset button. The bootloader starts.
- Press the reset button once for a short time.
- That's all, openSPOT application starts with

factory default settings (DHCP client mode).

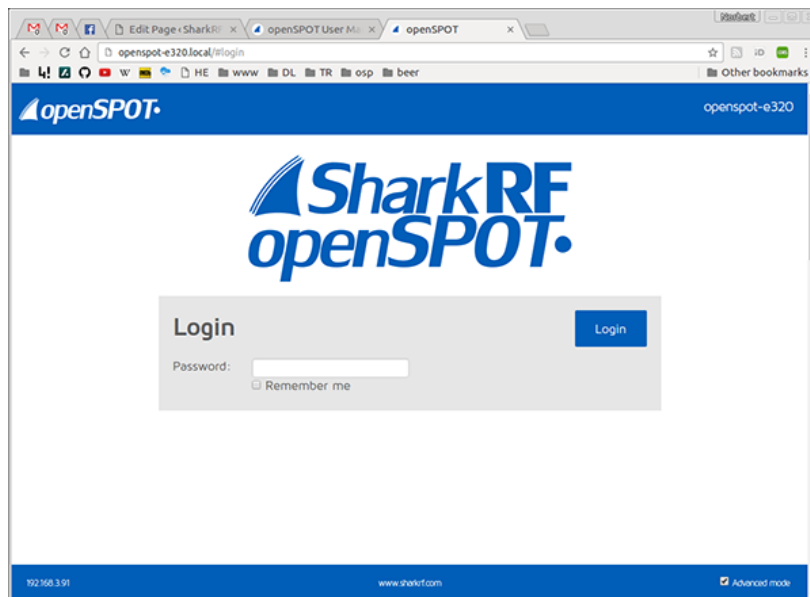
Web interface

When the web interface is loading data from openSPOT, a loader animation is displayed in the top right corner of the page.

Static information on all pages of the web interface

- openSPOT's hostname is in the upper right corner of the page. This is not visible on small displays.
- openSPOT's current IP address is in the lower left corner of the page. This is not visible on small displays.
- Advanced mode checkbox is in the lower right corner of the page. If this gets checked, all advanced mode settings show up. If you uncheck it, advanced mode settings will hide. Advanced mode is not needed to be turned on for normal operation.

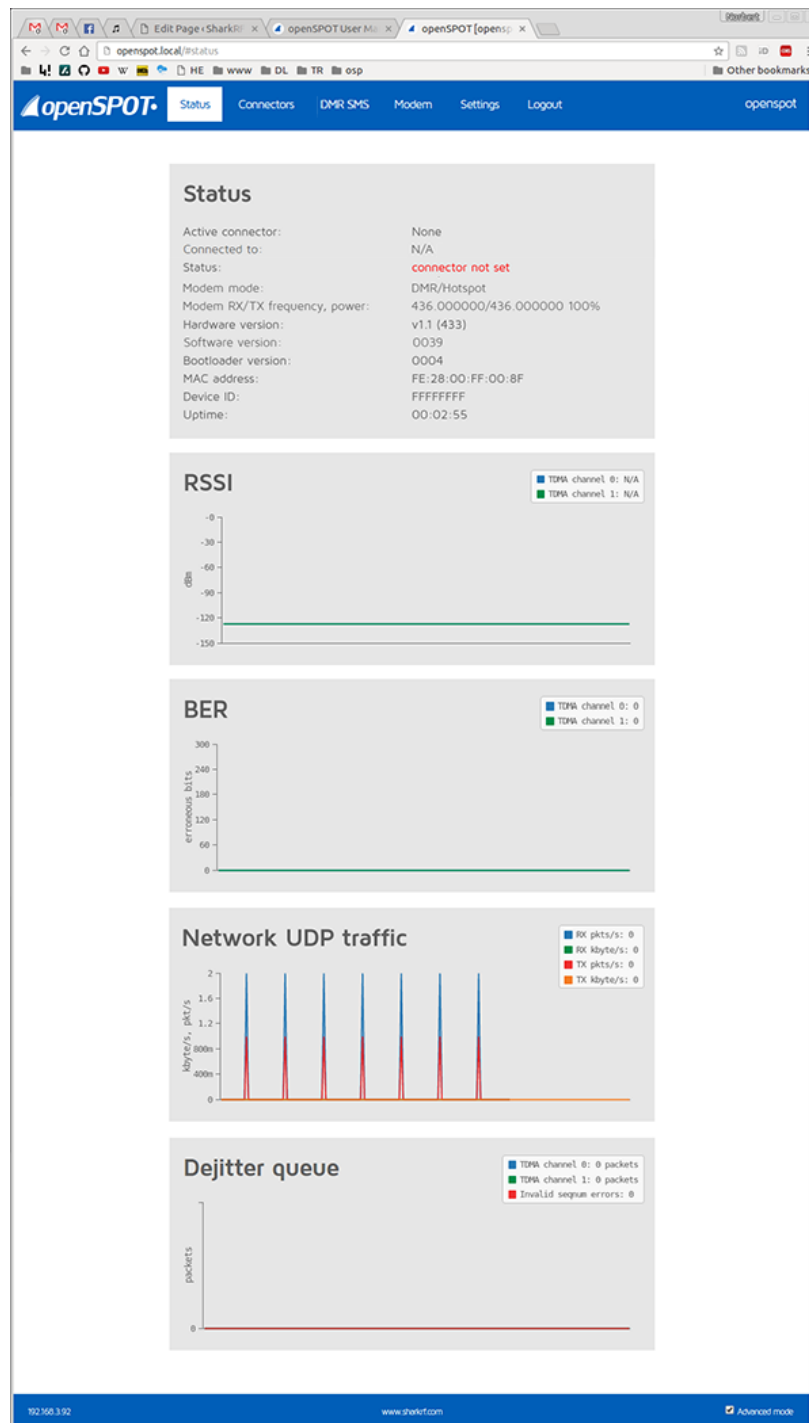
Login page



This is the page where you can log in to the web interface. Enter openSPOT's password, then click on the *Login* button. If you check the *Remember me*

checkbox, openSPOT will not ask for a login until it gets rebooted, or your browser forgets the login cookie.

Status page



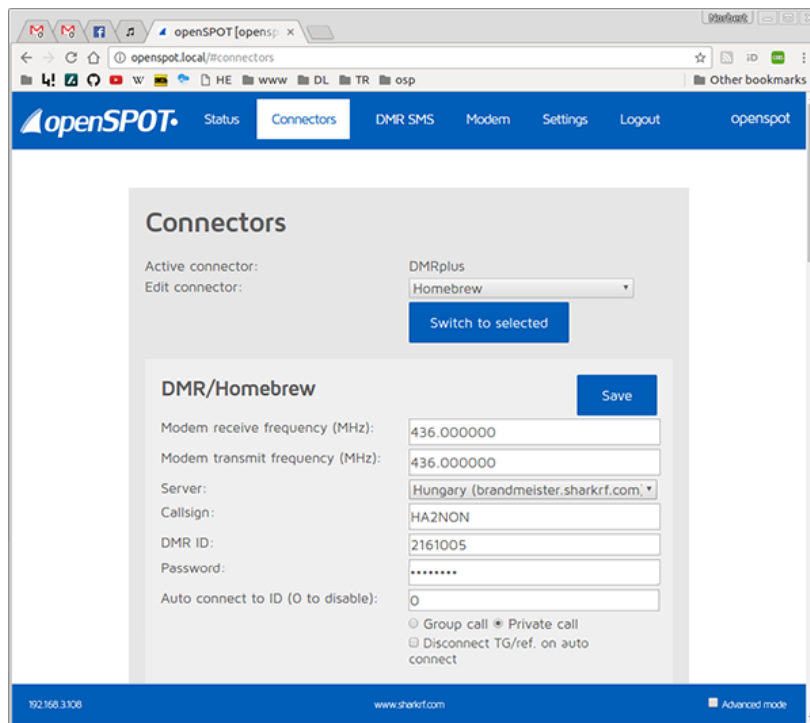
You can see the current status of openSPOT on this page. All dynamic information are refreshed every 500ms.

- The RSSI graph shows receive signal strength for

received packets **from the modem**. It is not updated when openSPOT modem is not synced to a station.

- The BER graph shows received Bit Error Ratio. BER is calculated also for packets received from connectors, not just from the modem.
- The Network UDP traffic graph shows transmit and receive UDP traffic for the currently active connector. Note that UDP+IP+Ethernet header overhead is not included in displayed values.
- Dejitter queue graph (only shown in advanced mode) shows the number of packets waiting in the dejitter queue. All packets **received from a connector** will first go to this queue to eliminate network jitter. If the dejitter queue length is too short, the graph's line can touch zero during a QSO and the call's audio can stop for a moment. If the length is too long, the delay can be unnecessarily high. You can use this graph to set a proper dejitter queue length. There should be at least 4-5 packets in the dejitter queue when receiving a call from a connector. You can set the dejitter queue length on the *Settings* page. As a general rule, if you have a good network connection to the selected connector, set dejitter queue length to 500ms, and if you have a 3G/4G internet connection or voice stutters a lot, try setting the length to at least 1000ms.

[Connectors page](#)



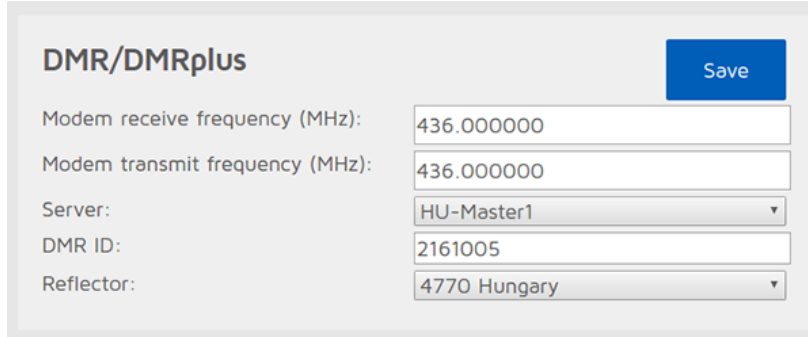
In openSPOT terms, a connector is the protocol which you use to connect your openSPOT to a server.

- openSPOT can connect to one connector at a time. You can edit a connectors settings by selecting it from the *Edit connector* list.
- The currently active connector can be changed by selecting a new one from the *Edit connector* list, and then clicking on the *Switch to selected* button.
- Don't forget to click on the *Save* button after changing a connector's settings.
- If the connector you switch to doesn't match the modem's current mode, the web interface will show a popup dialog, where you can switch the modem to the appropriate mode. For example, if the current mode is DCS, and the current modem mode is – obviously – D-STAR, and you switch to the Homebrew connector, the popup dialog will ask you if you want to switch the modem to DMR mode. If you miss the popup, you can do this manually on the *Modem* page of course.
- Some connectors have "Add server" and "Remove" buttons. These buttons can be used to add and remove custom servers to the server list. Note that custom servers are stored in your

browser's local storage, so if you switch browser or computer, you won't see your saved custom servers.

DMR connectors

DMRPlus connector

The screenshot shows a configuration window titled "DMR/DMRplus" with a blue "Save" button in the top right corner. Below the title, there are five input fields arranged in two columns. The first column contains labels: "Modem receive frequency (MHz):", "Modem transmit frequency (MHz):", "Server:", "DMR ID:", and "Reflector:". The second column contains the corresponding values: "436.000000", "436.000000", a dropdown menu showing "HU-Master1", a text field with "2161005", and a dropdown menu showing "4770 Hungary".

Label	Value
Modem receive frequency (MHz):	436.000000
Modem transmit frequency (MHz):	436.000000
Server:	HU-Master1
DMR ID:	2161005
Reflector:	4770 Hungary

You can use this connector to connect openSPOT to the DMRplus network.

Modem modes supported in this connector: DMR, C4FM

To use the DMRplus connector, follow these steps:

1. Select a server from the list, preferably the nearest to your current location.
2. Enter your DMR ID.
3. Select a DMRplus reflector from the list.
4. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server.
- You can set the UDP port of the connector. The default port for DMRplus is 8880.
- You can manually enter a reflector ID.
- You can set the keepalive interval and RX timeout in seconds. The default keepalive interval for DMRplus is 1 second, and RX timeout is 30 seconds.

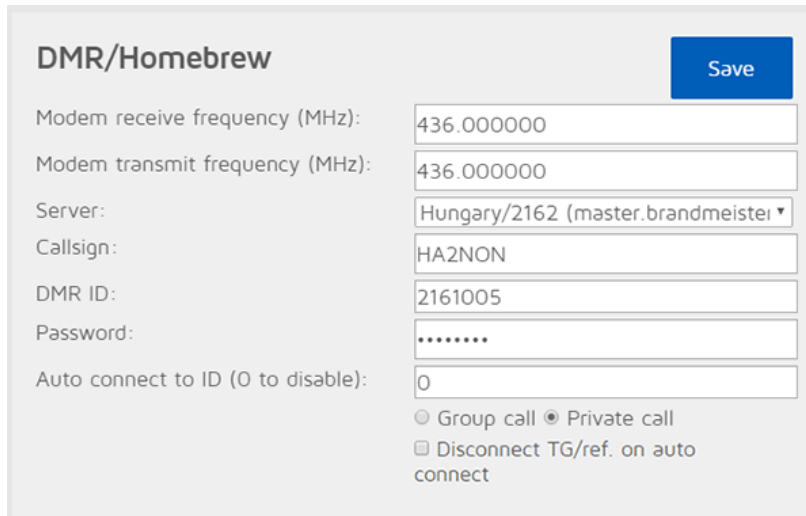
When this is the active connector, openSPOT's local echo service is available. Start a call to the local echo DMR ID (by default it's 9999, this can be changed on the *Settings* page), and openSPOT will play back the recorded voice. Calls started on the local echo DMR

ID won't get sent to the network.

You can change the current reflector without the web interface by starting a short call with your radio to a DMR ID between 4000 and 5000.

If you are using a C4FM radio, you can change the current reflector without the web interface by sending a DTMF code in the format *4004 for connecting to reflector 4004.

Homebrew connector



The screenshot shows a web form titled "DMR/Homebrew" with a "Save" button in the top right corner. The form contains several input fields and radio buttons. The fields are: "Modem receive frequency (MHz)" with value "436.000000", "Modem transmit frequency (MHz)" with value "436.000000", "Server:" with a dropdown menu showing "Hungary/2162 (master.brandmeister)", "Callsign:" with value "HA2NON", "DMR ID:" with value "2161005", "Password:" with masked characters ".....", and "Auto connect to ID (0 to disable):" with value "0". Below the "Auto connect to ID" field, there are two radio buttons: "Group call" (unselected) and "Private call" (selected). Below these, there is a checkbox labeled "Disconnect TG/ref. on auto connect" which is currently unchecked.

Modem receive frequency (MHz):	436.000000
Modem transmit frequency (MHz):	436.000000
Server:	Hungary/2162 (master.brandmeister ▼)
Callsign:	HA2NON
DMR ID:	2161005
Password:
Auto connect to ID (0 to disable):	0

☐ Group call ☒ Private call

☐ Disconnect TG/ref. on auto connect

You can use this connector to connect openSPOT to a DMR network which supports the Homebrew protocol, like **Brandmeister**.

Modem modes supported in this connector: DMR, C4FM

To use the Homebrew connector, follow these steps:

1. Select a server from the list, preferably the nearest to your current location.
2. Enter your callsign.
3. Enter your DMR ID.
4. Enter the server password. The default password for Brandmeister servers is *passwOrd* or *password* (note that some servers does not use the default password – you have to ask your server's admin for the password if openSPOT won't connect with the default one)
5. Enter the reflector or talkgroup ID you want to connect to in the *Auto connect to ID* field. (optional)
6. Set the *Auto connect to ID* call type. If you want

auto connect to a reflector, set it to Private call, if you want to auto connect to a talkgroup, set it to Group call. (optional)

7. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

8. Start a short **private** call to the reflector DMR ID you want to connect to, or a short **group** call to the group DMR ID you want to talk to.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server.
- You can set the UDP port of the connector. The default port for Homebrew is 62030.
- You can set the TDMA channel (0 or 1) where auto connect will try to link the given ID.
- You can set the auto connect interval in seconds. By settings this, you can stay connected to the given auto connect TG/reflector at all times. Setting this interval other than zero is not recommended, see the notes below.
- You can set the TDMA channel for data in DMO mode. DMO mode is enabled if the modem's RX and TX frequencies match, or the modem is in DMR Hotspot mode. In this case, only one TDMA channel will be used for communication. Note that on BrandMeister, DMO mode links the two timeslots together, so this setting is irrelevant.
- You can set the DMR ID to route cross-mode C4FM calls to. Also you can set the call type to private or group.
- You can set an ID and call type to reroute calls to and from TG9. See the note below.
- You can set the keepalive interval and RX timeout in seconds. The default keepalive interval for Homebrew is 5 seconds, and RX timeout is 30 seconds.

Call rerouting

You can enable rerouting by setting a reroute ID other than zero. If rerouting is active, all calls are rerouted to and from TG9. This makes using the Homebrew protocol easier for users used to simpler networks like DMRplus.

For example, if you set the reroute ID to 9990/private call, then if you call TG9 on your radio, your call will be routed to 9990/private, and the echo reply will come back to you also on TG9. This way you don't have to deal with RX group lists. Another example: if you set the reroute ID to 3100/group call, then if you call TG9 on your radio, your call will be sent to TG3100, and if someone talks on TG3100, the call will be sent to TG9 on your radio, so it will receive it even if you don't have TG3100 in your RX group list.

Rerouting is dynamic, if you switch to a new reflector/talkgroup, reroute ID will be automatically set to the new reflector/talkgroup.

SSIDs

You can optionally use SSIDs to connect multiple hotspots to the Brandmeister network. For example, if your DMR ID is 2161005, and you want to connect with SSID 43, append the number to your DMR ID in the Homebrew connector settings DMR ID field like this: 216100543. Brandmeister will automatically link you as 2161005 with SSID 43. This way you don't need multiple DMR IDs for multiple hotspots.

Auto connect

As now most of the Brandmeister servers are upgraded to the latest version, using the auto connect feature is not required. The new BM version stores which TG/reflector you were last on, and when you reconnect to the server with your hotspot, it automatically links you to the last used TG/reflector.

If auto connect is enabled, and you switch talkgroup/reflector from your radio, the auto connect ID will be set to the new talkgroup/reflector.

If you are using auto connect to a talkgroup, it is advisable to check "Disconnect TG/ref. on auto connect". If it's enabled, openSPOT will automatically disconnect the currently active TG/reflector after connecting to the server. This is useful if you only want to use a talkgroup, but the

server always links you to a reflector automatically.

Note that the Homebrew protocol does not support auto reflector connecting. openSPOT uses a workaround, it starts a short, empty call to the given reflector/talkgroup ID, so the server will connect the desired reflector/talkgroup. These periodic empty calls will show up on the BrandMeister dashboard. If this bothers you, set auto connect interval to 0, or disable auto connect.

Other notes

When this is the active connector, openSPOT's local echo service is available. Start a call to the local echo DMR ID (by default it's 9999, this can be changed on the *Settings* page), and openSPOT will play back the recorded voice. Calls started on the local echo DMR ID won't get sent to the network.

You can change the current reflector without the web interface by starting a short **private** call with your radio to a reflector's DMR ID. However, this reflector ID won't get saved in the settings, so if the current connection gets lost, you will not be on any reflectors. In that case, you have to manually connect openSPOT to a reflector by starting a short **private** call to the preferred reflector's DMR ID. You can also connect talk groups (if supported by the current master), by dialing their ID and starting a short **group** call.

If you are using a C4FM radio, you can modify the default DMR ID and call type for cross-mode C4FM calls dynamically by sending a DTMF code in the format *4004 for a private call to ID 4004, and #204 for a group call to ID 204.

If you can't connect to a server, please make sure you are using the correct password and UDP port number. If you don't know them, ask the server's administrator.

TS repeat connector

You can use this connector to repeat calls received from one timeslot to the other one. This is only

useful when the modem is in DMR/BS mode. This connector has no settings.

Modem modes supported in this connector: DMR

D-STAR connectors

DCS/XLX connector

D-STAR/DCS/XLX

Save

Modem receive frequency (MHz):
436.000000

Modem transmit frequency (MHz):
436.000000

Server:
DCS025/H (Hungary)

Callsign:
HA2NON

You can use this connector to connect openSPOT to a network which supports the DCS protocol.

Modem modes supported in this connector: D-STAR

To use the DCS connector, follow these steps:

1. Select a server and remote module from the list, preferably the nearest to your current location.
2. Enter your callsign.
3. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server.
- You can set the UDP port of the connector. The default port for DCS is 30051.
- You can set the CCS UDP port of the connector. The default port for CCS is 30062.
- You can manually set the local module (D by default).
- You can manually set the reflector ID.
- You can manually set the remote module.
- You can set the RX timeout in seconds. The default RX timeout is 30 seconds.

When this is the active connector, openSPOT's local echo service is available. Start a call to the local echo callsign (by default it'sE (7 spaces and an E, replace dots with spaces), this can be changed on

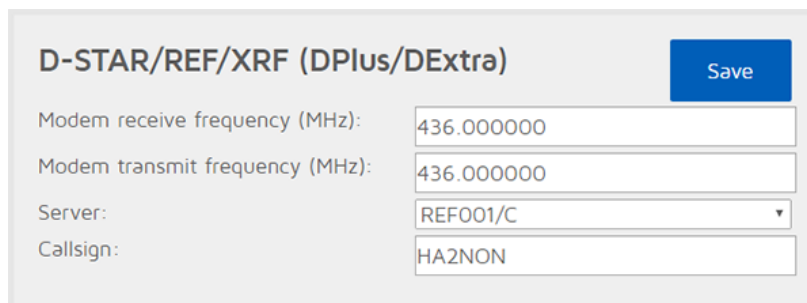
the *Settings* page), and openSPOT will play back the recorded voice. Calls started on the local echo callsign won't get sent to the network.

Try remote module Z on any reflector for the DCS echo service (may not enabled on some servers).

If you start a call to the repeater info callsign/(7 spaces and an I, replace dots with spaces), a voice will say what reflector is currently connected.

You can switch reflectors with your radio by starting a short call to their names as the destination callsign (urcall), like *DCS001 A*, or *XLX001 A*. Note that when you switch servers using this method, the web interface will not show the currently active server in the server dropdown list of the connector settings. You'll find the active server at the server host input box when you turn on the advanced settings.

REF/XRF (DPlus/DExtra) connector



The screenshot shows a web interface for configuring the D-STAR/REF/XRF (DPlus/DExtra) connector. It features a title bar with the connector name and a 'Save' button. Below the title bar are four input fields: 'Modem receive frequency (MHz)' with the value '436.000000', 'Modem transmit frequency (MHz)' with the value '436.000000', 'Server' with a dropdown menu showing 'REF001/C', and 'Callsign' with the value 'HA2NON'.

D-STAR/REF/XRF (DPlus/DExtra)	
Modem receive frequency (MHz):	436.000000
Modem transmit frequency (MHz):	436.000000
Server:	REF001/C
Callsign:	HA2NON

You can use this connector to connect openSPOT to a network, or a D-STAR repeater which supports the REF (DPlus) protocol.

Modem modes supported in this connector: D-STAR

To use the REF/XRF connector, follow these steps:

1. Select a server and remote module from the list, preferably the nearest to your current location.
2. Enter your callsign.
3. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server/repeater.
- You can set the UDP port of the connector. The

default port for REF is 20001.

- You can set the CCS UDP port of the connector.

The default port for CCS is 30062.

- You can manually set the local module (D by default).

- You can manually set the reflector ID.

- You can manually set the remote module.

- You can set the RX timeout in seconds. The default RX timeout is 30 seconds.

Note that if you are connecting to a repeater, you have to set the reflector ID to the callsign of the repeater.

When this is the active connector, openSPOT's local echo service is available. Start a call to the local echo callsign (by default it'sE (7 spaces and an E, replace dots with spaces), this can be changed on the *Settings* page), and openSPOT will play back the recorded voice. Calls started on the local echo callsign won't get sent to the network.

Try remote module E on any reflector for the REF/XRF echo service (may not be enabled on some servers).

If you start a call to the repeater info callsignI (7 spaces and an I, replace dots with spaces), a voice will say what reflector is currently connected.

You can switch reflectors with your radio by starting a short call to their names as the destination callsign (urcall), like *REF001 A*, or *XRFO01 A*, or *XLX001 A*.

Note that when you switch servers using this method, the web interface will not show the currently active server in the server dropdown list of the connector settings. You'll find the active server at the server host input box when you turn on the advanced settings.

System Fusion connectors

FCS connector

System Fusion/FCS

Save

Modem receive frequency (MHz):	436.000000
Modem transmit frequency (MHz):	436.000000
Server:	FCS001
Callsign:	HA2NON
CCS7 ID:	2161005
Room number:	25

You can use this connector to connect openSPOT to a network which supports the FCS protocol.

Modem modes supported in this connector: C4FM, DMR

To use the FCS connector, follow these steps:

1. Select a server and remote module from the list, preferably the nearest to your current location.
2. Enter your callsign.
3. Enter your CCS7/DMR ID.
4. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server.
- You can set the UDP port of the connector. The default port for FCS is 62500.
- You can manually enter the reflector name.
- You can set the keepalive interval and RX timeout in seconds. The default keepalive interval for FCS is 1 second, and RX timeout is 30 seconds.

You can change rooms with your C4FM radio by entering DTMF code in the format *99 for room 99. If you are using a DMR radio, then you can change rooms by starting a private call to the desired room number.

YSFReflector connector

System Fusion/YSFReflector

Save

Modem receive frequency (MHz):
436.000000

Modem transmit frequency (MHz):
436.000000

Server:
Alabama Link / Alabama Link

Callsign:
HA2NON

You can use this connector to connect openSPOT to a network which supports the YSFReflector protocol.

Modem modes supported in this connector: C4FM, DMR

To use the YSFReflector connector, follow these steps:

1. Select a server from the list.
2. Enter your callsign.
3. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can manually enter a host name or IP address for the server.
- You can set the UDP port of the connector. The default port for YSFReflector is 42000.
- You can set the keepalive interval and RX timeout in seconds. The default keepalive interval for YSFReflector is 5 seconds, and RX timeout is 30 seconds.

SharkRF connectors

SharkRF IP Connector Server

SharkRF IP Connector Server

Save

Modem receive frequency (MHz):
436.000000

Modem transmit frequency (MHz):
436.000000

Password:
.....

This is the UDP API server connector you can use to develop your own applications which use openSPOT. You can also use this connector to directly link two openSPOTs together without any servers. The

protocol's documentation can be found on [GitHub](#).

Modem modes supported in this connector: DMR, D-STAR, C4FM, raw (with automatic cross mode conversion between DMR and C4FM DN mode)

openSPOT accepts all connecting client IDs if the password is correct.

To start a SharkRF IP Connector Server, follow these steps:

1. Enter the server's password.
2. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can set the UDP port of the server. The default port is 65100.
- You can set the RX timeout in seconds. The default RX timeout is 30 seconds.

If SharkRF IP Connector Server is the active connector, the following status box shows up on the status page below the *Status* box:

SharkRF IP Connector Server

Connected client ID:	12345
Connected client callsign:	HA2NON

Note that you'll need to forward the UDP port of the server (65100 by default) on your router from the Internet to openSPOT's LAN IP address if you want a client from the Internet to be able to connect to it.

SharkRF IP Connector Client

SharkRF IP Connector Client Save

Modem receive frequency (MHz): 434.000000

Modem transmit frequency (MHz): 434.000000

Server: No saved servers ▾

Server address: 192.168.3.105

Add server Remove

ID: 2161006

Password:

Callsign: HG1MA

This is the UDP API client connector protocol you can use to develop your own applications which use openSPOT. You can also use this connector to connect openSPOT to the SharkRF IP Connector Protocol Server, or a network which supports the SharkRF IP Connector protocol. The protocol's documentation can be found on GitHub. You can also directly link two openSPOTs together without any servers (one of the openSPOT needs to be a client, and the other needs to be a server).

Modem modes supported in this connector: DMR, D-STAR, C4FM, raw (with automatic cross mode conversion between DMR and C4FM DN mode)

To start a SharkRF IP Connector Client, follow these steps:

1. Enter the server's address.
2. Enter an ID, like your DMR/CCS7 ID.
3. Enter the server password.
4. Optionally enter your callsign.
5. Click on the *Save* button, then click on the *Switch to selected* button to make this the active connector.

If advanced mode is turned on, the following settings become available:

- You can set the UDP port of the server. The default port is 65100.
- You can set the keepalive interval and RX timeout in seconds. The default keepalive interval is 5 seconds, and RX timeout is 30 seconds.

DMR AutoCal

DMR AutoCal

Save

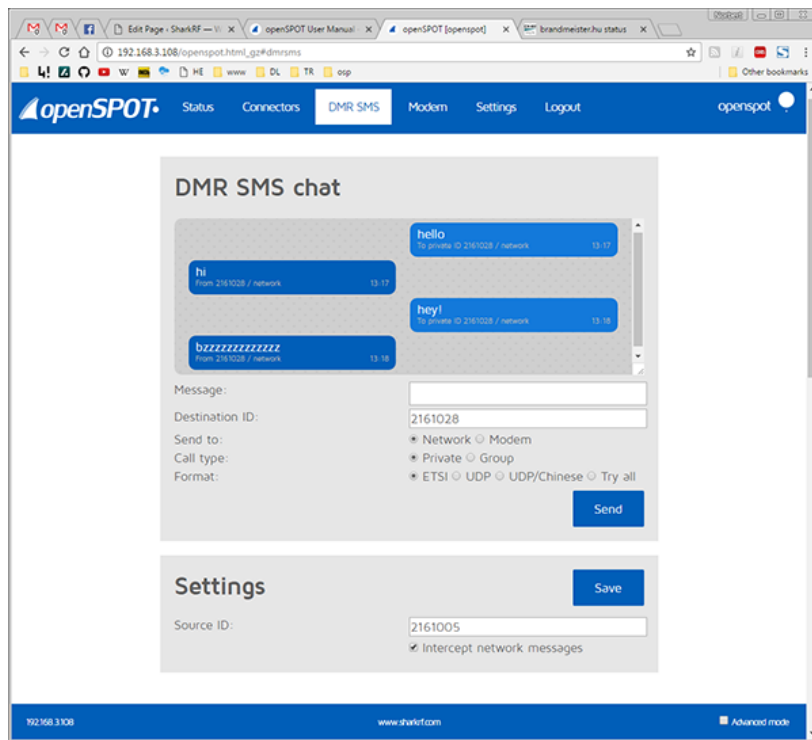
Modem receive frequency (MHz):	436.000000
Modem transmit frequency (MHz):	436.000000
Status:	calibrating...
Progress:	<div></div>
Result:	N/A

If you have high BER and bad uplink audio quality, you can use this special connector to find out which DMR demodulation mode gives the lowest BER and best audio quality with your specific radio. After activating this connector, press and hold PTT until a result is available. The progress bar will go up to 100% after about 1-2 minutes of continuous transmitting with your radio. After a DMR demodulation mode result is available, it is automatically set, and AutoCal process restarts until you release the PTT.

DMR echo test works as normal, but note that RSSI and BER graph data is unavailable when this connector is active.

DMR SMS page

When the modem is in DMR mode, or the currently active connector supports DMR SMS sending, the DMR SMS page shows up on the menu bar.



- Sent messages are displayed on the right, received messages are displayed on the left of the chat box. You can click on each message for additional information.
- The message length is currently limited to 75 characters.
- Note that the modem radio button is disabled if the modem is not in DMR mode.
- Messages sent to openSPOT's DMR echo ID (it's 9999 by default) will be echoed back to the sender.
- Note that openSPOT only stores the last received message, so you have to keep open this page in your browser to receive more than one message at once.
- You can simply use the ETSI format if you want to send messages through the network.
- Use ETSI format if you want to send a message to Hytera radios, and UDP format for Motorola radios. TDMA channel can be selected if the modem is in DMR-BS or DMR-MS modes.
- If you want to receive messages from the network on the DMR SMS page, you have to enable

Intercept network messages. If it's enabled, received SMS messages to the given *Source ID* won't be forwarded to your radio.

- Note that the underlying network must know about the location of your *Source ID*. If you want to receive DMR SMS messages from the network, you have to set the *Source ID* to the DMR ID you set on the connector page, or if you want to use another ID, then you have to start a short call from that DMR ID with your radio, so the network will know that your custom DMR ID can be reached through your openSPOT.

Here's a video about using the DMR SMS page:

openSPOT: Sending and receivin...



[Modem page](#)

Modem settings Save

Mode/submode:

Frequency Save

Modem receive frequency (MHz):

Modem transmit frequency (MHz):

Transmit power:

Modulation Save

Modulation mode:

Bits per second:

Inner deviation (Hz):

Packet settings Save

Packet size (bits):

Sync word length (bits):

Sync word pos. in packet (bits):

Sync word count:

Sync word #1 (hex):

Sync word #2 (hex):

Sync word #3 (hex):

Sync word #4 (hex):

TDMA Save

☐ TDMA enabled

TDMA calibration duration (packets):

TDMA calibration comp. multiplier:

Sync frames num. for chan. validation:

Calibration Save

☒ Auto calibration

Quick cal. delay (sec):

Autocal modem CPU temp. diff. (°C):

Autocal RF modem temp. diff. (°C):

Autocal temp. read interval (sec):

Autocal temp. read delay (sec):

Other settings Save

Call hang time (ms):

RSSI average sample count:

☐ High gain, low linearity

☐ Auto AGC

AGC low threshold (dBm):

AGC high threshold (dBm):

☐ External VCO

Reset modem config to defaults

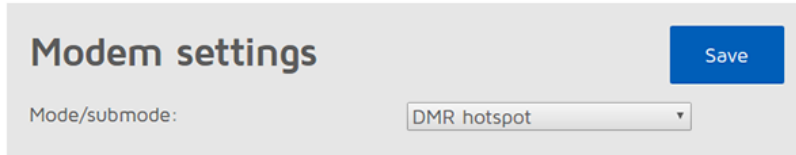
102.968.3.91 www.statefarm.com Advanced mode

On this page, you can change the settings of the modem. All gray blocks have a separate *Save* button, so if you change a setting in a block, make sure you click on it's *Save* button.

If you changed settings in a way the modem won't work for you anymore, you can always reset it's

settings to defaults using the *Reset modem config to defaults* button at the bottom of the modem settings page.

Modem mode



Modem settings Save

Mode/submode: DMR hotspot

Here you can change the modem's active mode.

- Idle mode: In this mode, the modem won't be able to receive or transmit.
- DMR/Hotspot mode. Use this if you want to use a DMR connector. Calls will be sent on both both timeslots simultaneously, so you don't have to worry about timeslot setting in your radio.
- D-STAR mode: Use this if you want to use a D-STAR connector.
- C4FM mode: Use this if you want to use a System Fusion connector.

Advanced mode modem settings

Raw mode

In this mode, the received and sent data will be unprocessed.

DMR/BS mode

Experimental DMR base station mode. In this mode, both timeslots work separately.

DMR transceivers handle BS and simplex operation by looking at the RX and TX frequencies of the current channel. If both RX and TX frequencies are the same, the transceiver will operate in simplex mode, if they are different, in BS mode. If you press PTT and your transceiver is in BS mode, it will first try to wake up the base station by transmitting BS Outbound Activation CSBK frames. The BS then should start to send idle DMR frames on it's

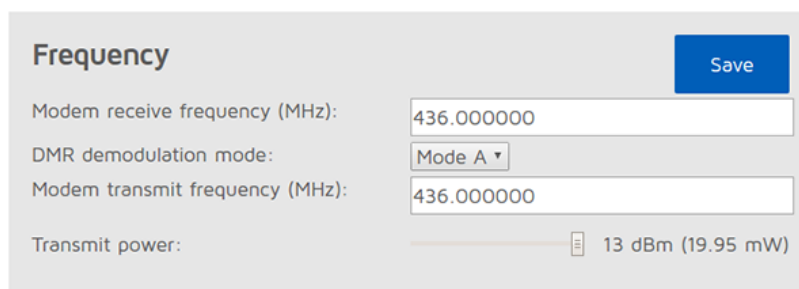
downlink, which the transceiver will detect and then starts the call. If downlink frames are not detected during the wake up phase, the transceiver will respond with a *Repeater wake up failed* message. If the transceiver is in simplex mode, no wake up process is performed.

For the DMR/BS mode to work correctly, you have to set separate RX and TX frequencies both for openSPOT and your transceiver. As openSPOT only has one modem it is not capable of full duplex operation, it can only receive or only transmit at a time. Therefore, the difference between RX and TX frequencies doesn't matter. When openSPOT receives a BS Outbound Activation CSBK frame, it will send a few idle frames on both timeslots, so the transceiver will know the TDMA sync. However, DMR transceivers are not designed for waking up half-duplex base stations, so the wake up procedure is not as reliable with openSPOT as with an industrial full-duplex repeater.

DMR/MS mode

Experimental DMR mobile station mode. In this mode, both timeslots work separately. As BS Outbound Activation CSBK frames are not sent by openSPOT automatically to wake the base station up, you have to do this manually using the UDP API.

Frequency



Frequency Save

Modem receive frequency (MHz): 436.000000

DMR demodulation mode: Mode A ▾

Modem transmit frequency (MHz): 436.000000

Transmit power: 13 dBm (19.95 mW)

Here you can set openSPOT's RX and TX frequencies and its transmit power. Note that changing the frequency here also overwrites the active connector's frequency settings.

- DMR demodulation mode: this is available in DMR modes only. If you have high BER and bad uplink

audio quality, you can try to switch to a different DMR demodulation mode, however, finding out which mode gives you the lowest BER and best audio quality is much easier with using the **DMR AutoCal** special connector. The easiest way to listen to your audio is to use openSPOT's build in DMR local echo (it's on DMR ID 9999 by default). The default demodulation mode is A.

Modulation

Modulation		Save
Modulation mode:	4FSK (Raised cosine) ▼	
Bits per second:	9600	
Inner deviation (Hz):	648	

This block is only available in advanced mode. It's not available in D-STAR modem mode. You can set modem modulation parameters here.

- Modulation mode: modem modulation mode.
- Bits per second: bitrate (not symbol rate).
- Inner deviation (Hz): the inner deviation in Hz. For 2FSK modes, this is the deviation. For 4FSK modes, outer deviation is calculated as *inner deviation* * 3.

Packet settings

Packet settings		Save
Packet size (bits):	288	
Sync word length (bits):	48	
Sync word pos. in packet (bits):	132	
Sync word count:	4	
Sync word #1 (hex):	7F7D5DD57DFD	
Sync word #2 (hex):	D5D7F77FD757	
Sync word #3 (hex):	755FD7DF75F7	
Sync word #4 (hex):	DFF57D75DF5D	

This block is only available in advanced mode. It's not available in D-STAR modem mode. You can set modem packet settings here.

- Packet size (bits): size of a packet in the current mode.

- Sync word length (bits): size of a sync word.
- Sync word pos. in packet (bits): sync word position.
- Sync word count: how many sync words are used.
- Sync word #n (hex): sync words given in hex character pair format. If any of these match, it will be considered as a packet sync.

TDMA

TDMA

Save

☐ TDMA enabled

TDMA calibration duration (packets):

TDMA calibration comp. multiplier:

Sync frames num. for chan. validation:

This block is only available in advanced mode. It's not available in D-STAR modem mode. You can set TDMA parameters here.

- TDMA calibration duration (packets): for precise TDMA operation, timer calibration is needed. The calibration duration can be set here. For example, DMR packets are 30ms long, so 100 packets will give a 3 second calibration duration.
- TDMA calibration comp. multiplier: you can set a (float) multiplier here for the calibration variables. Changing this from 0 is not needed under normal operating circumstances.
- Sync frames num. for chan. validation: this many frames with sync needed to be received by openSPOT so we can acquire TDMA channel lock.

Calibration

Calibration

Save

☒ Auto calibration

Quick cal. delay (sec):

30

Autocal modem CPU temp. diff. (°C):

10

Autocal RF modem temp. diff. (°C):

10

Autocal temp. read interval (sec):

60

Autocal temp. read delay (sec):

30

This block is only available in advanced mode.
You can set calibration parameters here.

- Quick cal. delay (sec): openSPOT waits this many seconds after the last call is finished to start a quick calibration.
- Autocal modem CPU temp. diff. (°C): automatic calibration happens when last automatic calibration and current modem CPU temperature differ this much.
- Autocal RF modem temp. diff. (°C): automatic calibration happens when last automatic calibration and current RF modem temperature differ this much.
- Autocal temp. read interval (sec): read temperatures from modem CPU and RF modem at this interval.
- Autocal temp. read delay (sec): wait at least this many seconds after a call has been finished to read the temperature.

Other modem settings

Other settings

Save

Call hang time (ms):

3000

RSSI average sample count:

10

AGC low threshold (dBm):

-50

AGC high threshold (dBm):

-80

☐ High gain, low linearity

☐ Auto AGC

☐ External VCO

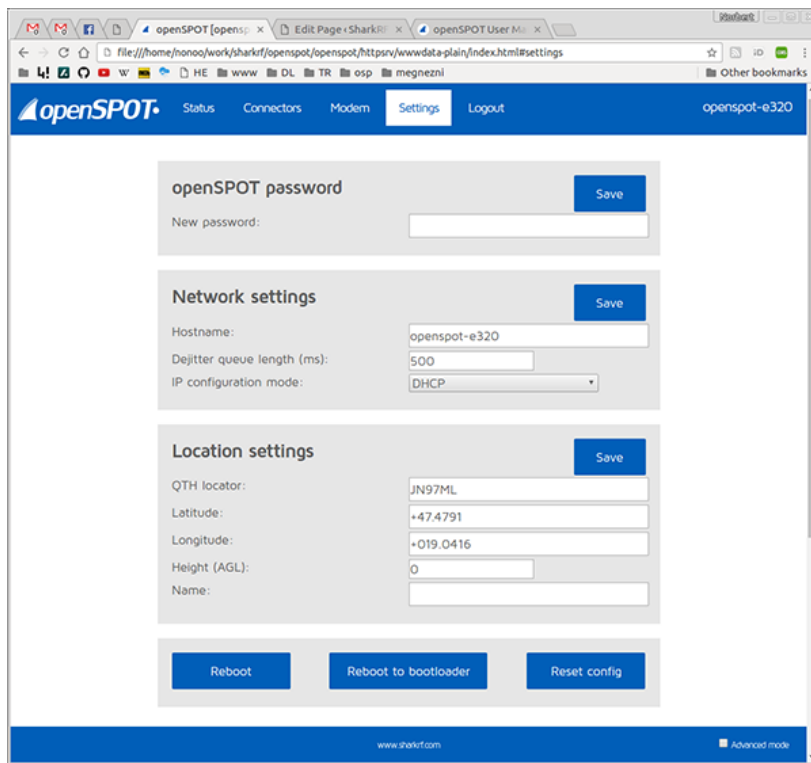
Reset modem config to defaults

This block is only available in advanced mode.
You can set other modem parameters here.

- Call hang time (ms): openSPOT will wait for this many milliseconds after the last valid packet has been received to consider a call has been finished with a timeout.
- RSSI average sample count: read and average RSSI values from the RF modem this many times.
- Sensitivity level: the modem's sensitivity level can be changed in 6 levels. Note that increasing sensitivity decreases the linearity of the filters, which may result in incorrectly demodulated bits (which means increased BER). By default, sensitivity is low (level set to 5 or 6).
- Filter gain: changes sensitivity of the filters. Note that increasing sensitivity decreases the linearity of the filters, which may result in incorrectly demodulated bits (which means increased BER). By default, filter gain is 8.
- Auto AGC: openSPOT's AGC is configured by software for preset values optimal for indoor use and low BER. However, you can turn automatic AGC on, but it is recommended to keep this checkbox off so openSPOT can work on it's own.
- AGC low/high threshold (dBm): when Auto AGC is on, these values will be considered as low and high thresholds for the Auto AGC circuitry in the RF modem.
- External VCO: an external VCO can be applied to openSPOT which can be enabled here. Note: applying an external VCO is considered as an after-market modification and voids the warranty of openSPOT.

Auto AGC and sensitivity level settings are stored independently for each modem mode (DMR, D-STAR, C4FM).

[Settings page](#)



On this page, you can change the general and network settings of openSPOT.

openSPOT password

openSPOT password

Save

New password:

Here you can set the login password. You can set an empty password too.

Network settings

Network settings

Save

Hostname:

openspot-e320

Dejitter queue length (ms):

560

IP configuration mode:

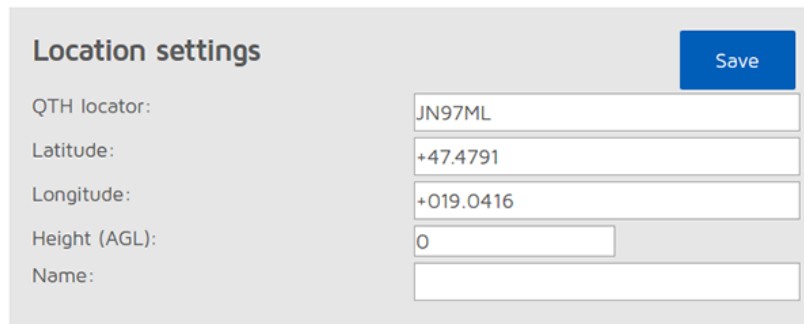
DHCP

- Hostname: this is the hostname of openSPOT. This can be used to open the web interface by entering *http://hostname.local/* or *http://hostname/* in the browser. This hostname will be shown in your router's DHCP Client List.
- Dejitter queue length (ms): this is the length of the dejitter queue in milliseconds. For more info, see the description of the *Dejitter queue graph* on the *Status*

page.

– IP configuration mode: this can be DHCP, DHCP with fallback to auto IP, auto IP and static. You can use auto IP modes if you want to connect openSPOT directly to a computer without a DHCP server. In that case, both the computer and openSPOT will get an auto IP address, and if internet connection sharing is on for the computer's Ethernet interface, openSPOT will be able to reach the internet.

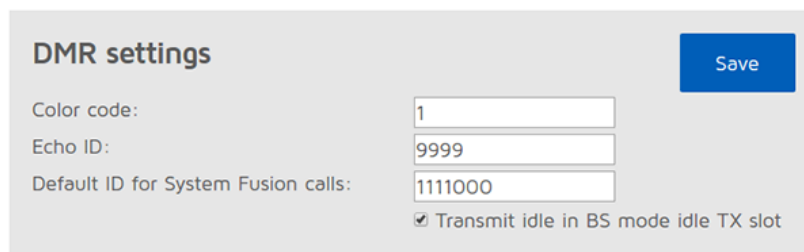
Location settings



The screenshot shows a web form titled "Location settings" with a blue "Save" button in the top right corner. The form contains five input fields: "QTH locator:" with the value "JN97ML", "Latitude:" with the value "+47.4791", "Longitude:" with the value "+019.0416", "Height (AGL):" with the value "0", and "Name:" which is currently empty.

– QTH locator/latitude/longitude/height: these location data is optional, can be used by some connectors like Homebrew to report the location of openSPOT to the connector's network.
– Name: you can give a name for your current QTH.

DMR settings



The screenshot shows a web form titled "DMR settings" with a blue "Save" button in the top right corner. The form contains three input fields: "Color code:" with the value "1", "Echo ID:" with the value "9999", and "Default ID for System Fusion calls:" with the value "1111000". Below these fields is a checkbox labeled "Transmit idle in BS mode idle TX slot" which is checked.

This block is only available in advanced mode.

- Color code: DMR frames with this color code will be processed only.
- Echo ID: if the currently active DMR connector supports it, you start a call to this DMR ID, openSPOT records the call and plays it back when the call is finished.
- Default ID for System Fusion calls: if the source callsign of the System Fusion call can't be parsed

into a number, this ID will be used for the outgoing DMR call.

- Transmit idle in BS mode idle TX slot: in DMR/BS mode, when only one of the TDMA channels are in a call, the modem will transmit idle frames in the other, unused TDMA slots. Note that keeping this on improves call quality.

D-STAR settings

D-STAR settings

Save

Echo callsign:

This block is only available in advanced mode.

- Echo callsign: if the currently active D-STAR connector supports it, you start a call to this callsign, openSPOT records the call and plays it back when the call is finished.

C4FM settings

C4FM settings

Save

☒ Auto mute DTMF commands

This block is only available in advanced mode.

- Auto mute DTMF commands: if a DTMF sequence is being entered, and the first character is a command character (some connectors use * and some use # too), the outgoing call will be automatically ended, so DTMF tones won't annoy other users on the same reflector/talkgroup.

Lock settings

Lock settings

Save

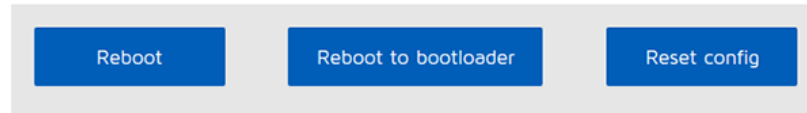
Lock to CCS7/DMR ID (0 to disable):

Lock to callsign (empty to disable):

This block is only available in advanced mode. It allows you to lock openSPOT to the given CCS7/DMR ID and/or callsign. If lock is enabled,

openSPOT's modem will only allow calls originating from the given ID or callsign.

Reboot/reset config



- Reboot button: reboots openSPOT
- Reboot to bootloader button: reboots openSPOT and it will automatically enter into bootloader mode. Note that after 1 minute of inactivity, bootloader will exit and starts the application, so it won't be stuck there.
- Reset config button: you can reset all settings to the defaults.

Using cross modem modes

Using a DMR connector with a C4FM radio

After you've switched to the given DMR connector, make sure that the modem is in *C4FM* mode. Only calls in *DN (Digital Narrow)* mode will be converted, if you transmit in other modes, DMR users will only hear silence.

You have to set your DMR ID as your callsign in your C4FM radio. Otherwise, the source DMR ID of your outgoing calls will be set to the *default System Fusion DMR ID* given in the DMR settings on the *Settings* page. The callsign of the incoming DMR calls will be the source DMR ID as text.

Using a C4FM connector with a DMR radio

After you've switched to the given C4FM connector, make sure that the modem is in *DMR/Hotspot* mode. Only calls in *DN (Digital Narrow)* mode will be converted, if someone transmits in other modes, you will hear silence on your DMR radio.

Your outgoing callsign will be your DMR ID as text. If an incoming call's source callsign can be parsed into a number, it will be its source DMR ID. If parsing fails, the *default System Fusion DMR ID* given in the DMR settings on the *Settings* page will be used as the call source DMR ID.

Using the openSPOT API

You can use both an HTTP and a UDP application programming interface to develop your own applications which use openSPOT.

You can find more information about the API protocols on our GitHub pages:

- [openSPOT HTTP API](#)
- [SharkRF IP Connector Protocol](#)

Acknowledgement

Thanks to our beta testers for their continuous work: István Száraz HA7PTY, Lajos Horváth HA5OGR!

Troubleshooting

If you have problems using openSPOT please ask your questions on the openSPOT [troubleshooting forum](#).

Specifications

- Dimensions: 71 x 67 x 31 mm (without antenna)
- Weight: 150 gr (without antenna)
- Operating temperature range: -40 – +85 C
- Power supply: 5 V DC through the USB port.
- Power consumption: max. 210 mA @ 5 V (1.05 W)

- Antenna impedance: 50 Ω
- RF power output: max. 13 dBm (20 mW)
- Receive/transmit frequency range: 421-458 MHz (JP version: 430-440MHz)
- Receiver sensitivity:
 - Gaussian 2FSK: -114 dBm @ 9.6 kbps
 - Raised Cosine 4FSK: -109 dBm @ 9.6 kbps
- Emission designators: A1A (CW ID), F1E/F1D (D-STAR, System Fusion), FXE, FXD (DMR)

The data sheet which can be used to apply for usage permit in Japan is available for download [here](#).

Further information

Please ask your further questions on the [openSPOT forum](#).

Open source licences

Here are the licences for various open source software used in openSPOT.

alertify.js

The MIT License

Copyright (c) 2009-2014 Stuart Knightley, David Duponchel, Franz Buchinger, António Afonso

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

balloon.css

The MIT License (MIT)

Copyright (c) 2016 Claudio Holanda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

contour.js

Copyright 2016 Forio Contour

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

d3.js

Copyright 2010-2016 Mike Bostock

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jquery.js

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ladda.js

Copyright (C) 2016 Hakim El Hattab, <http://hakim.se>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
THE SOFTWARE.

lodash.js

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

pure.css

Copyright 2013 Yahoo! Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
- Neither the name of the Yahoo! Inc. nor the
names of its contributors may be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

js-sha256.js

Copyright (c) 2015 Chen Yi-Cyuan

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

spin.js

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

sha2.c

/*

* FILE: sha2.c

* AUTHOR: Aaron D. Gifford - <http://www.aarongifford.com/>

*

* Copyright (c) 2000-2001, Aaron D. Gifford

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holder nor the names of contributors

* may be used to endorse or promote products derived from this software

* without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) "AS IS"

AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE
LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*

*/

heatshrink

Copyright (c) 2013-2015, Scott Vokes (vokes.s@gmail.com)

All rights reserved.

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL
WARRANTIES

WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE
FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY
DAMAGES

WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
AN

ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF

OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

jsmn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

HamGridSquare.js

```
// Copyright 2014 Paul Brewer KI6CQ
// License: MIT License http://opensource.org/licenses/MIT or CC-BY-SA
//
// Javascript routines to convert from lat-lon to Maidenhead Grid Squares
// typically used in Ham Radio Satellite operations and VHF Contests
//
// Inspired in part by K6WRU Walter Underwood's python answer
// http://ham.stackexchange.com/a/244
// to this stack overflow question:
// How Can One Convert From Lat/Long to Grid Square
// http://ham.stackexchange.com/questions/221/how-can-one-convert-from-lat-long-to-grid-square
```

SpinKit

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GetGitRevisionDescription.cmake

Original Author: 2009-2010 Ryan Pavlik rpavlik@iastate.edu, abiryan@ryand.net
<http://academic.cleardefinition.com>
Iowa State University HCI Graduate Program/VRAC

Copyright Iowa State University 2009-2010.

Boost Software License – Version 1.0 – August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative

works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

 January 16, 2017 12:06