

WaveShark Communicator

User Manual

Document version 1.1.0
2022-12-12

Covers *WaveShark Communicator* firmware version 1.4.0 and above
Covers *WaveShark Terminal* version 1.0.45 and above



Table of Contents

WaveShark Communicator introduction.....	3
Connecting to your WaveShark Communicator.....	4
External connections, power switch, and LEDs.....	8
Mesh networking.....	9
LoRa™.....	9
Wi-Fi networking.....	11
Other features.....	14
WaveShark Terminal for Windows.....	16
USB Serial interface.....	17
Building WaveShark Communicator networks.....	20
Building WaveShark Communicator network applications.....	21
Updating firmware.....	22
Amateur Radio operation.....	24
Appendix A: WaveShark Communicator USB Serial Interface commands.....	25
Appendix B: WaveShark Communicator device settings.....	28
Appendix C: WaveShark Communicator LoRa™ channels and frequencies.....	30
Appendix D: Enclosure fasteners.....	31

WaveShark Communicator introduction

- Off-grid wireless text messaging and general purpose data modem
- Built-in mesh networking to automatically extend the range of any group of WaveShark devices
- Low power long distance LoRa™ transceiver
- Bluetooth Low Energy (BLE) connectivity for WaveShark Android and iOS messaging apps
- Wi-Fi network connectivity, both for joining existing Wi-Fi networks and for acting as its own Wi-Fi Access Point (AP)
- USB Serial port for use with PCs, Macs, Raspberry Pis, and a variety of other devices
- Extensive control interface via USB Serial port
- Built-in optional AES 128-bit encryption for private communications
- High capacity 4,000 mAh internal rechargeable LiPo battery
- *WaveShark Terminal* software for point and click configuration and operation on Windows
- Field updateable firmware
- Designed, assembled, and individually tested in the USA

Connecting to your WaveShark Communicator

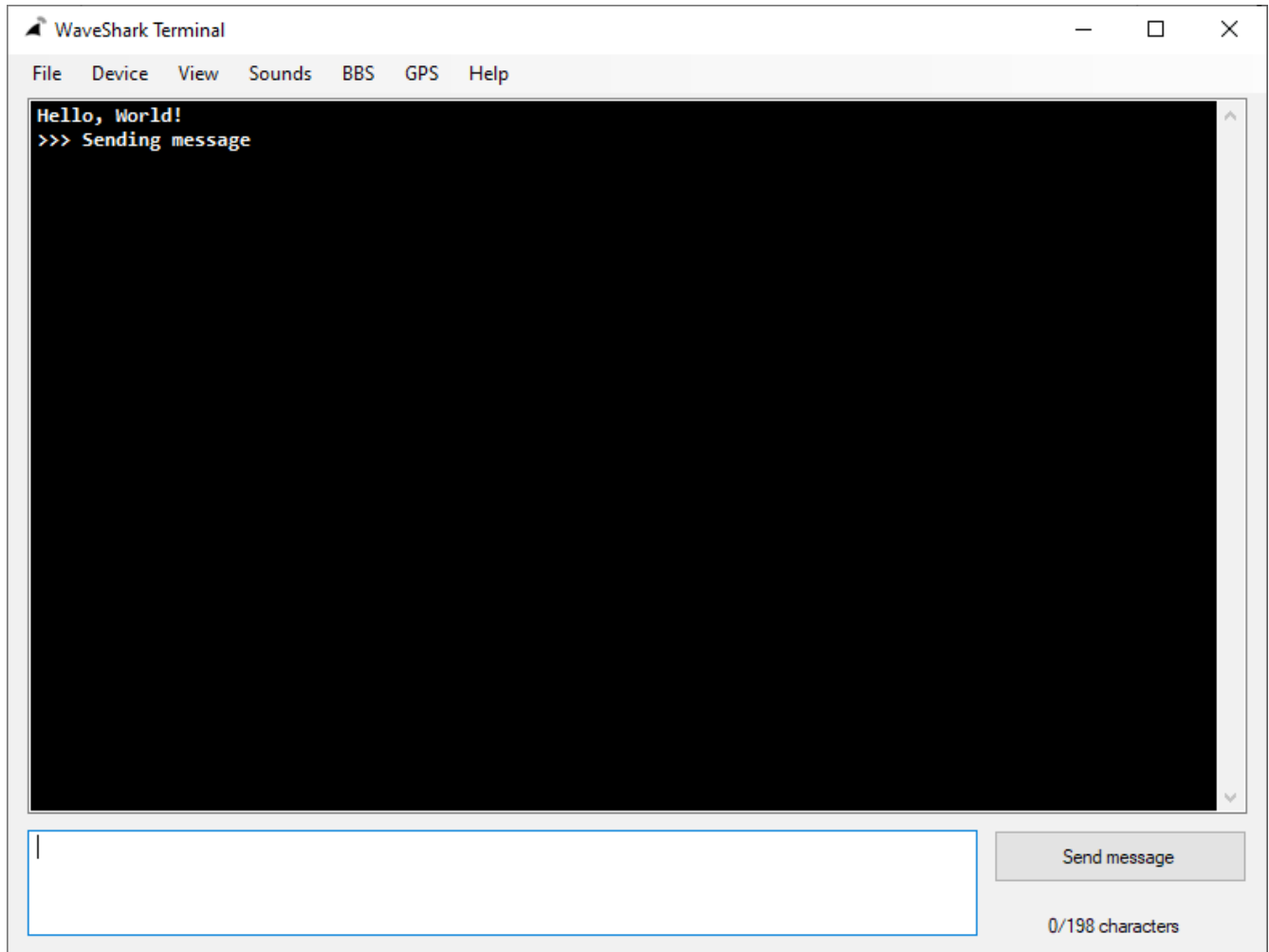
Android and iOS

Download and install the WaveShark app from the Android or iOS app store. This app uses Bluetooth Low Energy (BLE) to communicate with your WaveShark device. It begins by scanning for nearby WaveShark devices and allowing you to select one to pair with. Once the app has been paired with your WaveShark device, it can be used to communicate via text messages and to configure a number of options on your WaveShark device.



Windows PC

Connections from a computer to the WaveShark Communicator are made using a USB cable between the WaveShark Communicator and the computer. The easiest way to connect to your WaveShark Communicator using a Windows PC is with the *WaveShark Terminal* program, which can be freely downloaded from www.waveshark.net. The *WaveShark Terminal* will automatically scan for WaveShark Communicator devices connected to any of your USB ports. Alternatively, connections can be made using programs such as *PuTTY*, but you will have to likely use *Windows Device Manager* to determine which USB Serial port your WaveShark Communicator is associated with.



Mac

The following programs can be used to connect to your WaveShark Communicator device on a Mac:

- Serial (available in the Apple App Store)
- PuTTY
- screen
- minicom
- cu

Linux

The following programs can be used to connect to your WaveShark Communicator device under Linux:

- PuTTY
- screen
- minicom
- cu

Remote firmware update via Wi-Fi

The WaveShark Communicator firmware can also be updated via Wi-Fi. A few steps are required to accomplish this:

- A separate WaveShark Communicator (“*controlling WaveShark Communicator*”) is required to send the commands to the target WaveShark Communicator so that it becomes a Wi-Fi client (station) on an existing Wi-Fi network or so that it becomes its own Wi-Fi Access Point (AP)
- The target WaveShark Communicator must have Remote Control enabled (see *Remote Control* in the *Other features* section below)
- Both the controlling WaveShark Communicator and the target WaveShark Communicator must be configured with the same Remote Control key (see *Remote Control* in the *Other features* section below)
- The target WaveShark Communicator should have a Wi-Fi SSID set so that it knows which existing Wi-Fi network to connect to (Wi-Fi client) or what to call itself if acting as its own AP

```
/RCCMD /WIFISSID <SSID>
```

Alternatively, when your WaveShark Communicator will be acting as an AP, you can skip sending the /WIFISSID command and simply have it use its default SSID.

- The target WaveShark Communicator should have a Wi-Fi password as the default is no password. The password can be set by sending the following command to the target WaveShark Communicator:

```
/RCCMD /WIFIPASS <Password>
```

- Once both WaveShark Communicators are configured as specified above, the controlling WaveShark Communicator needs to send the /WIFI CLIENT or /WIFI AP command to the target WaveShark Communicator:

```
/RCCMD /WIFI CLIENT
```

or

```
/RCCMD /WIFI AP
```

- You can now find the WaveShark Communicator in your list of near-by Wi-Fi Access Points or find it on your local Wi-Fi network by checking with your Wi-Fi Access Point's DHCP clients. Connecting to the AP to upload the new firmware file. Once uploaded, the target WaveShark Communicator will reboot using the new firmware.

External connections, power switch, and LEDs

External connections

- RP-SMA female antenna connector
- MicroUSB connector for powering device, charging internal battery, and communicating with the WaveShark Communicator via its extensive USB Serial interface with a computer, laptop, *WaveShark Terminal* for Windows, *PuTTY*, smart phone or tablet using a USB On-The-Go (OTG) cable, Single Board Computers such as the Raspberry Pi, and more.
See *USB Serial interface* for more information. USB Serial connections should be made at 115,200 baud with “8-N-1” settings (one start bit, 8 data bits, no parity bit, and one stop bit).

Power switch

The WaveShark Communicator features an on/off power switch which physically disconnects all power sources from the device's internal microprocessor. Battery charging still takes place with the power switch in the off position if a power source is connected to the MicroUSB connector, but the LEDs will not operate to indicate whether or not charging is taking place. Battery charging will automatically stop when the battery is fully charged.

LEDs

LED operation is optional and can be enabled or disabled through the WaveShark Communicator's settings. In fact, LED operation must be disabled for the device to enter into its *power-saving mode*.

The WaveShark Communicator has 3 LED indicators. From top to bottom:

- Power LED
 - Solid when powered on and battery is not charging
 - Blinking (two seconds on, quarter second off) when battery is charging
 - Blinking (quarter second on, two seconds off) when battery is low
- Activity LED
 - Off when no receive or transmit activity is taking place
 - Solid when transmitting
 - Blinking when receiving
- Inbox LED
 - Off when no messages waiting to be read
 - Solid when at least one message is waiting to be read

Mesh networking

The WaveShark Communicator features built-in mesh networking that automatically extends the range of a group of WaveShark devices. For example, in situations where the range of an individual WaveShark Communicator is one mile, a number of WaveShark Communicators can combine into a mesh network that extends for many miles.

Message Time-To-Live (TTL)

The Time-To-Live (TTL) of a message within a WaveShark Communicator mesh network determines the maximum number of “hops” that a message will take. Higher TTL values can help push messages out further but can also lead to congestion, making timely delivery of messages difficult. Each WaveShark Communicator ships with a default TTL value that should provide both a long reach while cutting down on congestion. The TTL value can be changed using the WaveShark smart phone app, the *WaveShark Terminal*, or by any other means of connecting to the WaveShark Communicator.

LoRa™

The WaveShark Communicator uses a LoRa™ transceiver for its low power, low speed, long distance communications.

WaveShark Communicator LoRa™ channel

There are 100 different channels (frequencies) available for your WaveShark Communicator to communicate on. The WaveShark Communicator's default channel is 42. See Appendix C for a listing of all available channels and their corresponding frequencies.

LoRa™ transmit power level

23 different levels of LoRa™ transmit power are available, with a power level of 1 using the least power and a power level of 23 using the most power. The WaveShark Communicator's default LoRa™ transmit power level is 23.

LoRa™ Spreading Factor

The LoRa™ Spreading Factor determines the trade-off made between distance and speed. Higher Spreading Factors mean longer distance and lower speed. Conversely, lower Spreading Factors mean shorter distance and higher speed. The possible Spreading Factors are 7, 8, 9, 10, 11, and 12. Each WaveShark Communicator device is set to Spreading Factor 10 by default. We believe that Spreading Factor 10 represents a very good compromise between distance and speed.

The Spreading Factor can be changed by issuing the /SF command. For example, to change to Spreading Factor 12, issue the /SF 12 command.

Please be aware that WaveShark Communicator devices with different Spreading Factors cannot communicate with one another.

Although exact speed differences for different Spreading Factors can be provided, the same cannot be done with distance. It is, however, a fact that higher speed transmissions will travel less far than lower speed transmissions.

We welcome you to experiment with different Spreading Factor settings based on your particular network of WaveShark Communicator devices.

Spreading Factor	Distance	Speed	Speed increase over Spreading Factor 12
7	Minimum	427 bytes per second	18.6X
8		244 bytes per second	10.6X
9		137 bytes per second	6.0X
10 (Default)		76 bytes per second	3.3X
11		42 bytes per second	1.8X
12	Maximum	23 bytes per second	-

Wi-Fi networking

Although Wi-Fi is disabled by default on the WaveShark Communicator, it can easily be enabled so that the WaveShark Communicator acts as a client (station) in an existing Wi-Fi network or so that it acts as its own Wi-Fi Access Point (AP).

NOTE: Wi-Fi and Bluetooth are mutually exclusive on the WaveShark Communicator. Enabling one will automatically disable the other. This is due to a technical limitation with respect to the amount of memory available to the WaveShark Communicator's internal microcontroller.

Wi-Fi client mode

The WaveShark Communicator can operate as a Wi-Fi client (station) in an existing Wi-Fi network by connecting to an existing Wi-Fi Access Point (AP). The WaveShark Communicator will automatically obtain its own IP address and other network configuration information from an existing Wi-Fi network using DHCP.

Wi-Fi client mode can be configured from the WaveShark Communicator's USB Serial interface or Remote Control commands using the following commands:

```
/WIFISSID <SSID of Wi-Fi AP to connect to>  
/WIFIPASS <Password for connecting to Wi-Fi AP>  
(alternatively use /WIFIPASSCLEAR for connecting to Wi-Fi AP's which do not require a password)  
/WIFI CLIENT
```

You can use the /WIFI command without any arguments to get the current Wi-Fi information, including the WaveShark Communicator's IP address.

Wi-Fi Access Point (AP) mode

The WaveShark Communicator can operate as a Wi-Fi Access Point and provide an Access Point to other Wi-Fi devices to connect to, including other computers, smartphones, Wi-Fi enabled microcontrollers, Single Board Computers, and more.

Wi-Fi AP mode can be configured from the WaveShark Communicator's USB Serial interface or Remote Control commands using the following commands:

```
/WIFISSID <SSID of Wi-Fi AP>  
/WIFIPASS <Password for connecting to Wi-Fi AP>  
(alternatively use /WIFIPASSCLEAR for operating a Wi-Fi AP which do not require a password)  
/WIFI AP
```

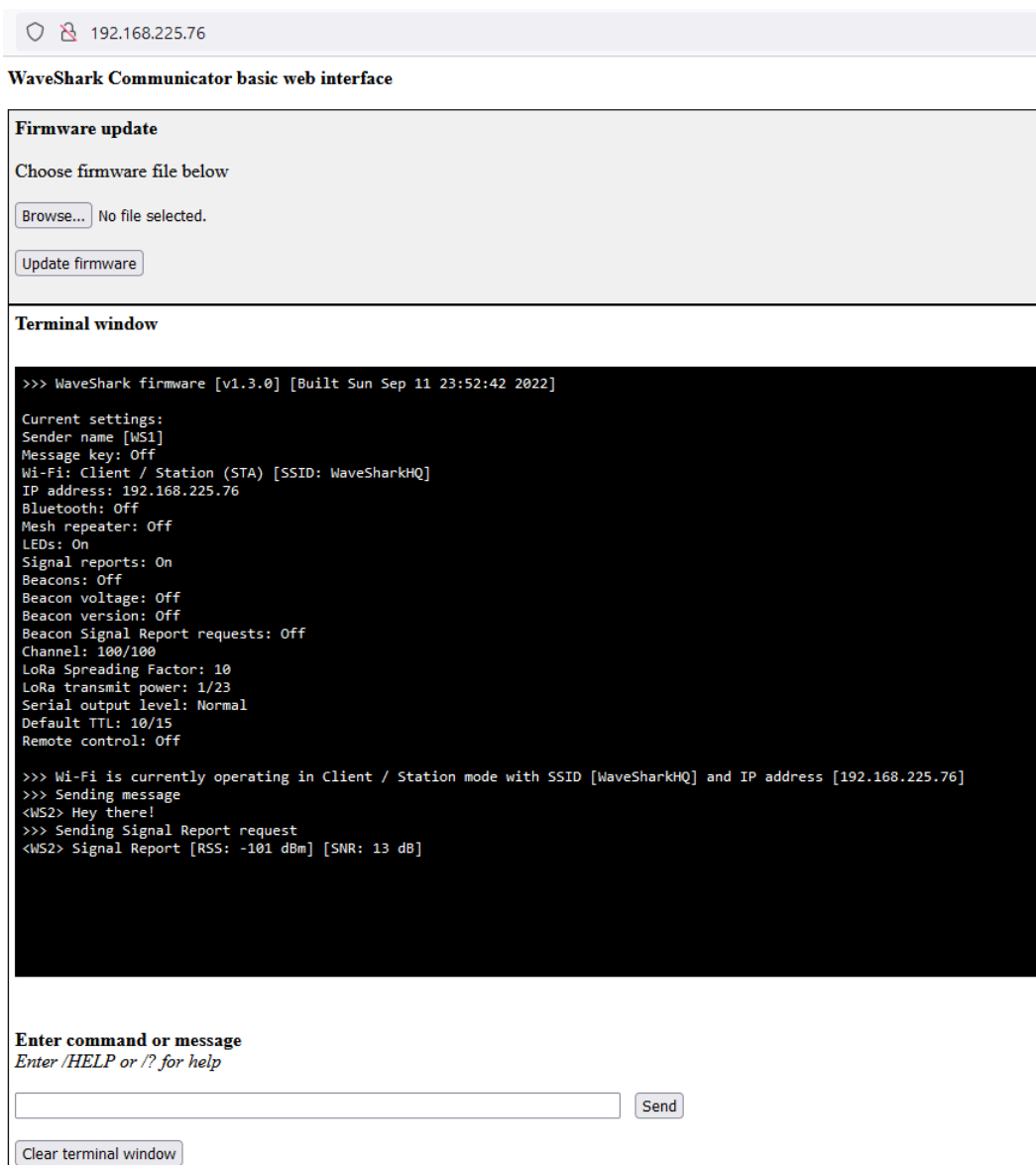
You can use the /WIFI command without any arguments to get the current Wi-Fi information, including the WaveShark Communicator's IP address. The WaveShark Communicator will always be found at 192.168.4.1 when operating as an AP.

Wi-Fi terminal

The WaveShark Communicator hosts a very convenient terminal-like experience, very similar to that found in the free *WaveShark Terminal* software for Windows, that you can use from your computer's Web browser or from your smartphone's Web browser.

Simply enter the IP address of your WaveShark Communicator and you will be presented with the following web page. This web page allows you full control of your WaveShark Communicator, allowing any command which can be sent via the USB Serial interface to be sent here. You can also send and receive messages from this web page. Additionally, this web page can be used to update the firmware on the device.

NOTE: This web page must be accessed using HTTP, not HTTPS (i.e. without SSL).



The screenshot shows a web browser window with the address bar displaying "192.168.225.76". The page title is "WaveShark Communicator basic web interface".

Firmware update

Choose firmware file below

No file selected.

Terminal window

```
>>> WaveShark firmware [v1.3.0] [Built Sun Sep 11 23:52:42 2022]
Current settings:
Sender name [WS1]
Message key: Off
Wi-Fi: Client / Station (STA) [SSID: WaveSharkHQ]
IP address: 192.168.225.76
Bluetooth: Off
Mesh repeater: Off
LEDs: On
Signal reports: On
Beacons: Off
Beacon voltage: Off
Beacon version: Off
Beacon Signal Report requests: Off
Channel: 100/100
LoRa Spreading Factor: 10
LoRa transmit power: 1/23
Serial output level: Normal
Default TTL: 10/15
Remote control: Off

>>> Wi-Fi is currently operating in Client / Station mode with SSID [WaveSharkHQ] and IP address [192.168.225.76]
>>> Sending message
<WS2> Hey there!
>>> Sending Signal Report request
<WS2> Signal Report [RSS: -101 dBm] [SNR: 13 dB]
```

Enter command or message
Enter /HELP or /? for help

Wi-Fi HTTP API

The WaveShark Communicator also hosts a simple HTTP API when operating as a Wi-Fi client or Wi-Fi Access Point (AP).

The following simple API end-points can be accessed via HTTP in order to perform various forms of automation with a WaveShark Communicator:

HTTP GET /responses

This end-point allows you to get any pending command output, or any pending messages, from the WaveShark Communicator. Typical applications will poll this end-point every 250, 500, or 1,000 milliseconds.

HTTP GET /waveshark

This end-point allows easy identification of a WaveShark Communicator operating on a Wi-Fi network. Accessing this end-point will return the string “WaveShark” for aid in identification.

HTTP GET /name

This end-point allows easy access to a WaveShark Communicator's sender name / device name.

HTTP GET /password

This end-point allows you to determine if the WaveShark Communicator is using a Wi-Fi password or not. It does **NOT** provide the actual Wi-Fi password, if any, being used.

HTTP POST /command

This end-point allows you to send commands or messages to the WaveShark Communicator. You can use it to perform any command available via the USB Serial interface and you can also use it to send messages to other WaveShark Communicators. The command or message should be sent under the “commandText” key.

Wi-Fi HTTP API cURL examples

Sending a message:

```
curl -X POST -s http://<Device IP address>/command -d "commandText=Hello out there!"
```

Rebooting the WaveShark Communicator:

```
curl -X POST -s http://<Device IP address>/command -d "commandText=/reboot"
```

Updating the device's sender name / device name:

```
curl -X POST -s http://<Device IP address>/command -d "commandText=/name My new name"
```

Getting the device's current settings as a JSON string:

```
curl -X POST -s http://<Device IP address>/command -d "commandText=/json"
```

Updating the device's sender name / device name from a JSON string:

```
curl -X POST -s http://<Device IP address>/command -d "commandText=/json {'senderName': 'My new name'}"
```

Other features

Power-saving mode

The WaveShark Communicator features a power-saving mode that allows the device to continue sending and receiving messages while using only approximately 10% to 25% of its normal power requirements, resulting in greatly extended battery life.

Several conditions must be met in order for the WaveShark Communicator to enter into its power-saving mode:

- LED operation must be disabled
- Bluetooth must be disabled
- Wi-Fi must be disabled
- No activity on the MicroUSB port for at least 15 minutes

Device anonymity

Unlike smart phones, basic cellular phones, and most Internet-connected devices, WaveShark Communicator devices do not transmit any information that can be used to link an individual WaveShark Communicator device or its transmissions back to the identity of the original purchaser. WaveShark Communicator messages do contain a somewhat random 16-bit device address in order to optimize mesh operations, but this number is very far from any sort of globally unique identifier and WaveShark maintains no records whatsoever that can be used to tie a WaveShark Communicator device address back to the original purchaser.

Beacons

A beacon mode can be enabled on the WaveShark Communicator to send out periodic messages every so many seconds.(every 60 to 3,600 seconds). Each beacon contains a sequential number (Beacon 1, Beacon 2, Beacon 3, etc). The WaveShark Communicator can optionally be set to send its current firmware version in each beacon. The WaveShark Communicator can also optionally be set to send its current USB and battery voltage levels in each beacon. Additionally, beacons can optionally contain a Signal Report request. Beacons are disabled by default.

Signal Reports

WaveShark Communicator devices can request Signal Reports from other WaveShark Communicator devices. Other WaveShark Communicator devices will send back a Signal Report if this feature is enabled, which it is by default.

Encryption

Each WaveShark Communicator device can perform AES 128 bit encryption and decryption in order to perform private communications. Encryption is disabled by default.

Mesh networking encryption

Encrypted messages still benefit from traveling through the mesh network, even when intermediary devices do not have the correct key to decrypt their contents.

Device name / Sender name

Each WaveShark Communicator device has a default device name which takes the form *WaveShark ABCDEF*, where *ABCDEF* is one of 16,777,216 different possible combinations of the numbers 0-9 and the letters A-F. This device name also doubles as the sender name included in each message that the WaveShark Communicator device sends. The device name can be changed at any time and can also be reverted to its default value at any time.

Inbox

Your WaveShark Communicator automatically stores a copy of the last 10 messages it has received. These messages will either be automatically retrieved when connected using the WaveShark app or can be manually retrieved by using the `/INBOX` command in the WaveShark Communicator USB Serial interface.

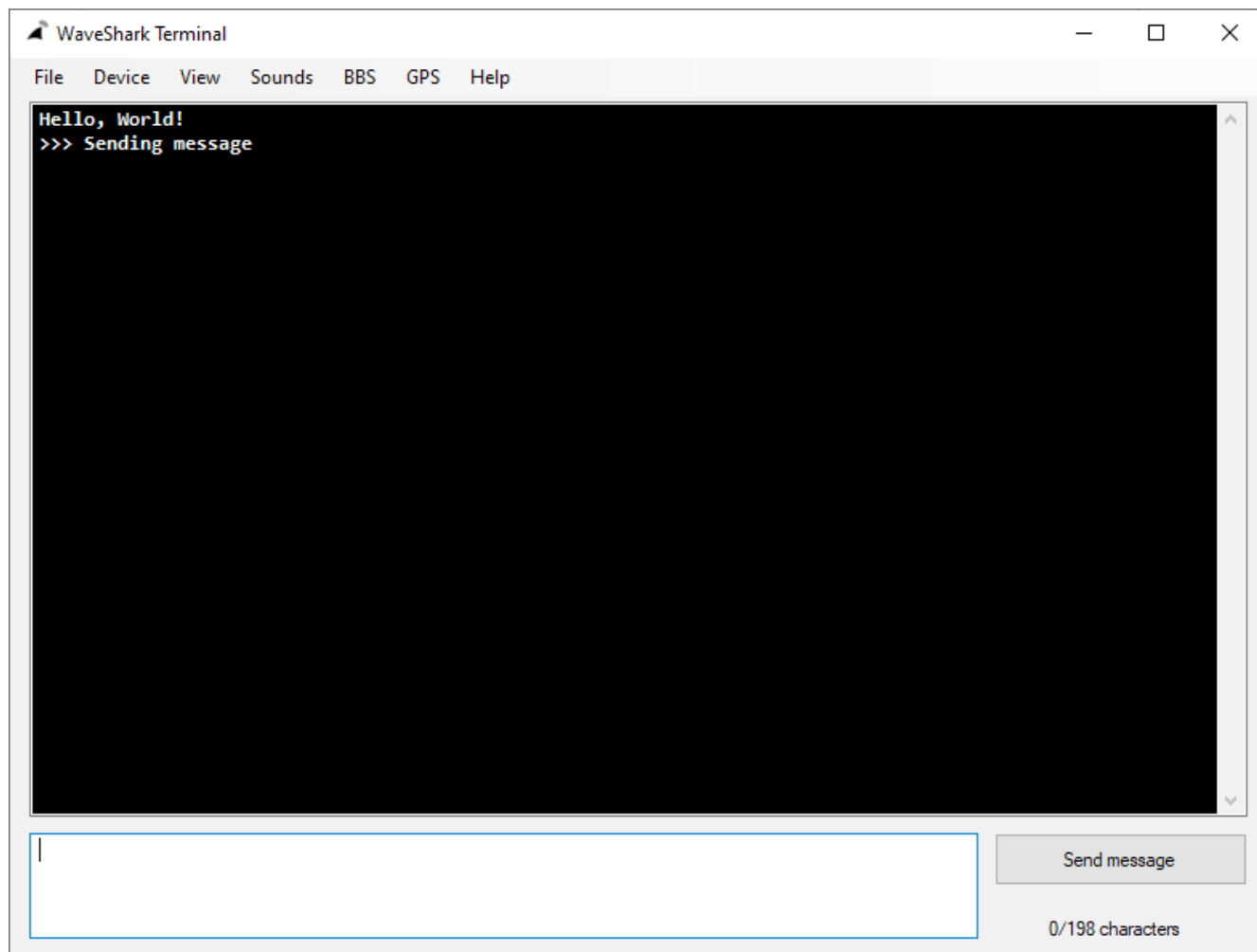
Remote Control

Although remote control of WaveShark Communicator devices is disabled by default, it can be enabled by issuing the `/REMOTECONTROL ON` command. Each WaveShark Communicator comes programmed with a random remote control key (password) to prevent accidental access in the event that remote control is enabled without setting a key. The remote control key can be set by issuing the `/RCKEY` command along with a new key. For example, `/RCKEY ABC123`.

Once remote control has been enabled on a WaveShark Communicator device and a remote control key has been set, the WaveShark Communicator can be remotely controlled (remotely configured) by other WaveShark Communicator devices using the same remote control key. Any command starting with a `"/` can be remotely invoked. For example, a device can be remotely rebooted by issuing the `/REBOOT` command and a device can be made to change its channel by issuing the `/CHANNEL` command along with a new channel number.

To send remote control commands to a WaveShark Communicator device, you must first set the name of the device that you wish to target. This can be accomplished using the `/RCTARGET` command along with the name of the device you wish to remotely control. For example, `/RCTARGET Tower1`. Once the remote control target has been set, you can send remote control commands using the `/RCCMD` command. For example, `/RCCMD /REBOOT` will send the reboot command to the device currently targeted. Successful commands that don't reboot the device, change its channel, or make it otherwise unable to immediately communicate back will be acknowledged with a message saying that the remote control command was successfully processed.

WaveShark Terminal for Windows



WaveShark Terminal is a program for using your WaveShark Communicator on a Windows PC. It can be freely downloaded from www.waveshark.net. *WaveShark Terminal* allows you to easily:

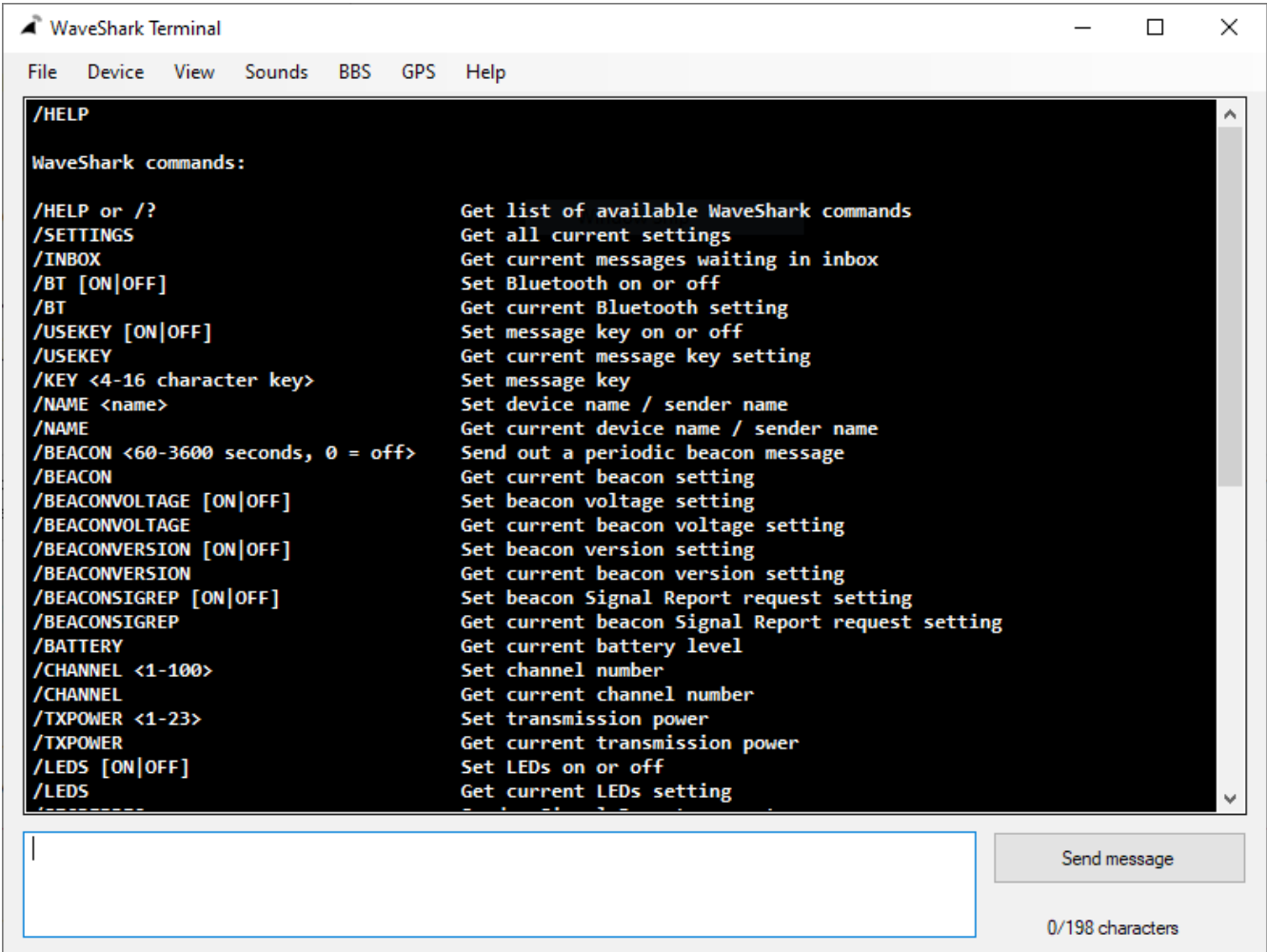
- Communicate with other WaveShark Communicators
- Update your WaveShark Communicator's settings
- Update your WaveShark Communicator's firmware
- Generate network coverage data when paired with a suitable GPS receiver
- Run a simple BBS

USB Serial interface

The WaveShark Communicator features an extensive control interface available via its USB port. This is referred to as its *USB Serial interface*. This interface is suitable for using your WaveShark Communicator with programs such as *WaveShark Terminal*, Putty, Terminate, Minicom, Screen, and more. It is also a very suitable interface for performing automatic control of your WaveShark Communicator. Examples of this sort of automatic control include writing custom software that uses your WaveShark Communicator as a modem to send and receive telemetry data, to provide a BBS for other WaveShark Communicator users to check into, and much more.

Built-in help

The *USB Serial interface* has a built-in help command that allows you to get a listing of every command that the WaveShark Communicator understands. This help can be accessed by sending either `/HELP` or `/?`.



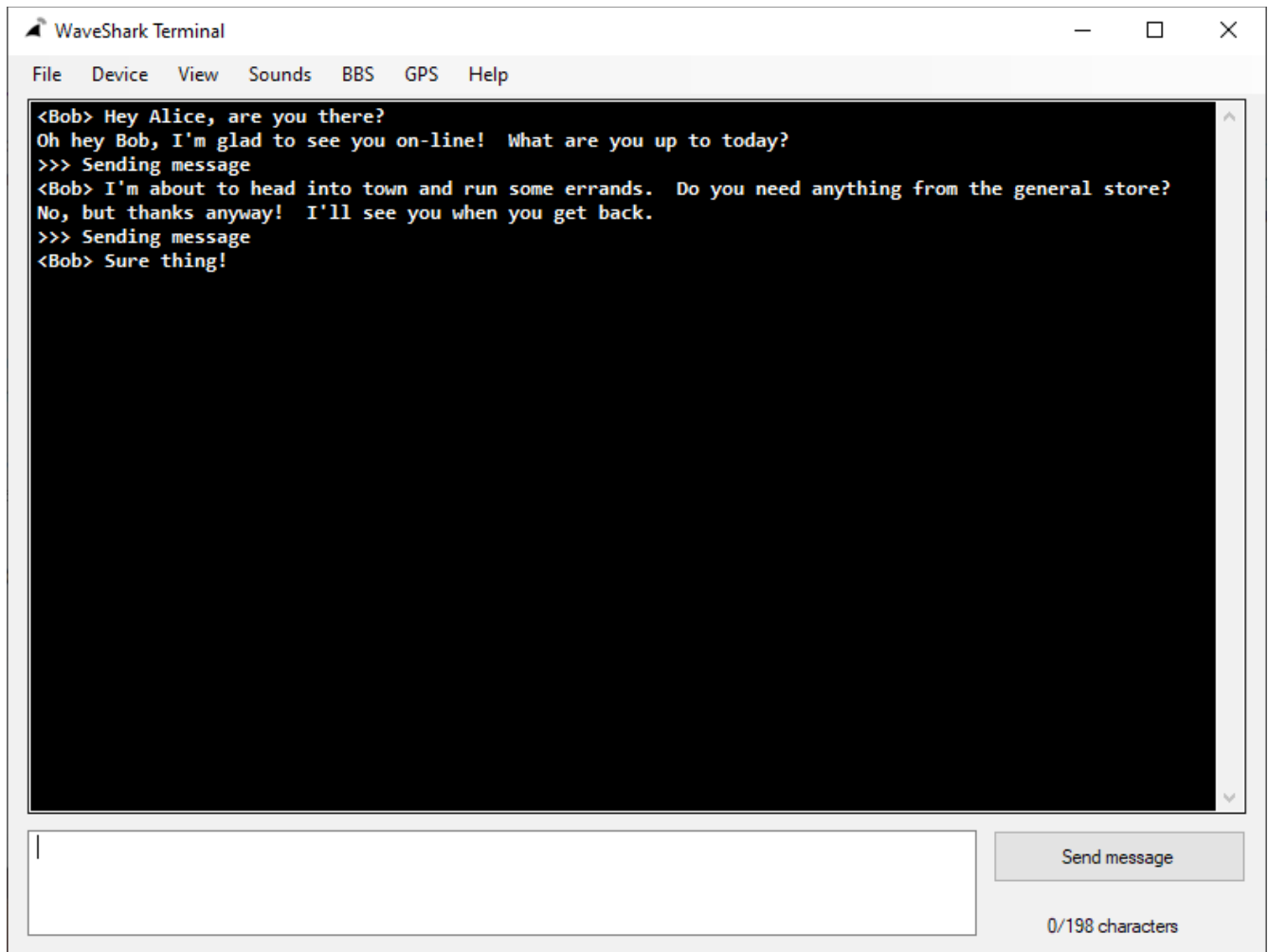
The screenshot shows the WaveShark Terminal application window. The title bar reads "WaveShark Terminal" and the menu bar includes "File", "Device", "View", "Sounds", "BBS", "GPS", and "Help". The main terminal area displays the output of the `/HELP` command, listing various WaveShark commands and their functions. Below the terminal area is a text input field and a "Send message" button. The character count "0/198 characters" is shown at the bottom right.

```
WaveShark commands:

/HELP or /?           Get list of available WaveShark commands
/SETTINGS             Get all current settings
/INBOX                Get current messages waiting in inbox
/BT [ON|OFF]          Set Bluetooth on or off
/BT                   Get current Bluetooth setting
/USEKEY [ON|OFF]      Set message key on or off
/USEKEY               Get current message key setting
/KEY <4-16 character key> Set message key
/NAME <name>          Set device name / sender name
/NAME                 Get current device name / sender name
/BEACON <60-3600 seconds, 0 = off> Send out a periodic beacon message
/BEACON               Get current beacon setting
/BEACONVOLTAGE [ON|OFF] Set beacon voltage setting
/BEACONVOLTAGE        Get current beacon voltage setting
/BEACONVERSION [ON|OFF] Set beacon version setting
/BEACONVERSION        Get current beacon version setting
/BEACONSIGREP [ON|OFF] Set beacon Signal Report request setting
/BEACONSIGREP         Get current beacon Signal Report request setting
/BATTERY              Get current battery level
/CHANNEL <1-100>      Set channel number
/CHANNEL              Get current channel number
/TXPOWER <1-23>       Set transmission power
/TXPOWER              Get current transmission power
/LEDS [ON|OFF]        Set LEDs on or off
/LEDS                 Get current LEDs setting
```

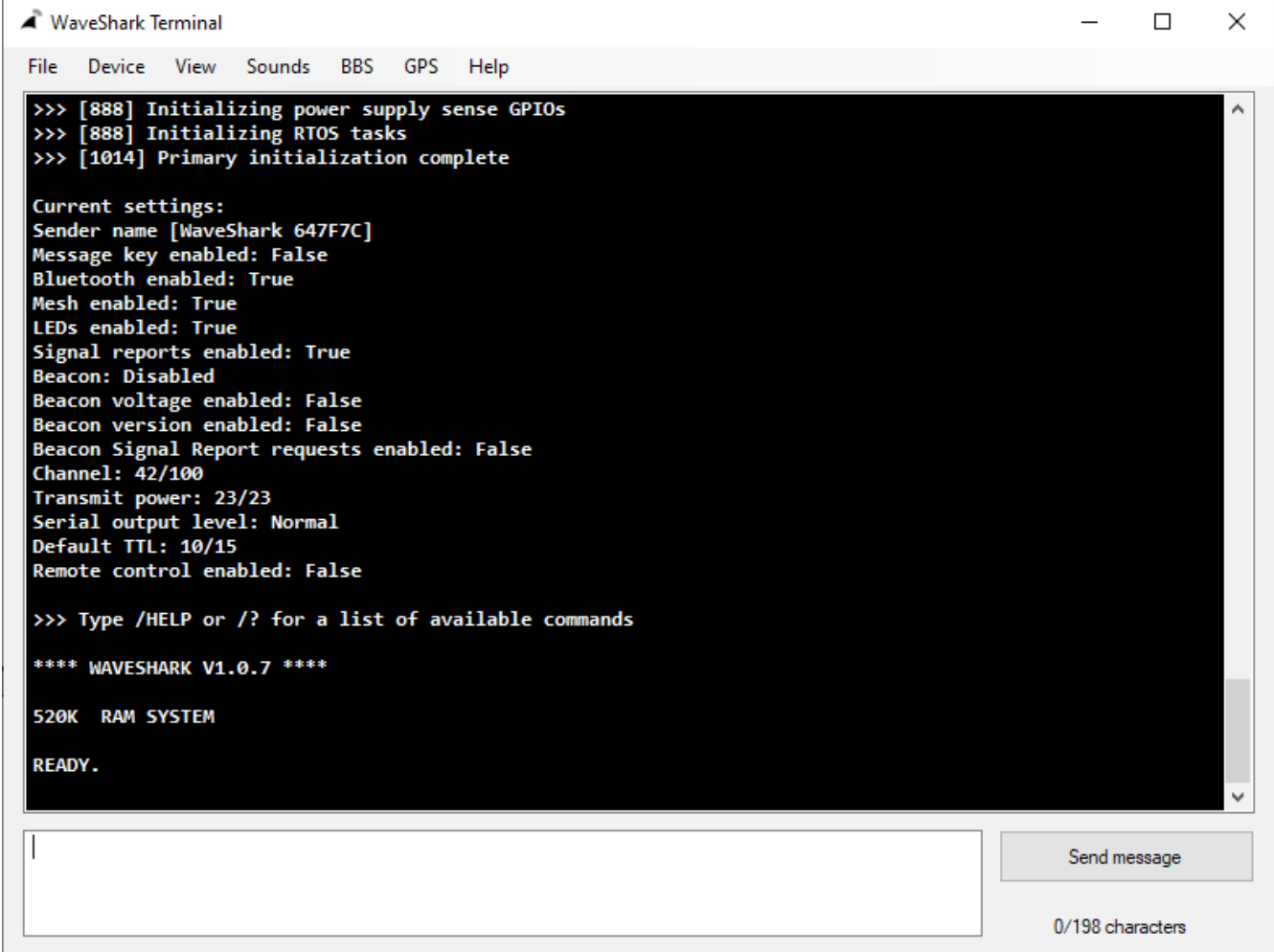
Sending and receiving messages

Any line of text entered into the *USB Serial interface* which does not start with a / (forward slash) character will automatically be interpreted as a message to send to other WaveShark Communicator devices. Messages sent from other WaveShark Communicator devices will automatically be made available via the *USB Serial interface*.



READY prompt

The WaveShark Communicator will always print “READY.” as its last line of output when first booting up. This is provided to ensure there is a consistent way for programs automatically controlling a WaveShark Communicator to know when the device is ready to respond to commands.



The screenshot shows a terminal window titled "WaveShark Terminal" with a menu bar containing "File", "Device", "View", "Sounds", "BBS", "GPS", and "Help". The terminal output is as follows:

```
>>> [888] Initializing power supply sense GPIOs
>>> [888] Initializing RTOS tasks
>>> [1014] Primary initialization complete

Current settings:
Sender name [WaveShark 647F7C]
Message key enabled: False
Bluetooth enabled: True
Mesh enabled: True
LEDs enabled: True
Signal reports enabled: True
Beacon: Disabled
Beacon voltage enabled: False
Beacon version enabled: False
Beacon Signal Report requests enabled: False
Channel: 42/100
Transmit power: 23/23
Serial output level: Normal
Default TTL: 10/15
Remote control enabled: False

>>> Type /HELP or /? for a list of available commands

**** WAVESHARK V1.0.7 ****

520K RAM SYSTEM

READY.
```

At the bottom of the terminal window, there is an input field with a cursor, a "Send message" button, and a character count "0/198 characters".

Available commands

See Appendix A for a listing of every available *USB Serial interface* command and an explanation of each command's usage.

Building WaveShark Communicator networks

Your best bet at extending communications to their maximum distance is to designate some WaveShark Communicators as “repeaters” and place them in high up places to maximize line-of-sight advantage.

We recommend that dedicated repeater devices be placed in a weatherproof enclosure along with a solar panel. A small 5 VDC 1 A (5W) solar panel should power any WaveShark device indefinitely. For dedicated repeater devices, we recommend disabling LEDs (/LEDS OFF), disabling Bluetooth (/BT OFF), and disabling Wi-Fi (/WIFI OFF) so that they will enter power-saving mode.

Another consideration is which WaveShark Communicator devices repeat messages and which do not. All WaveShark Communicator devices function as repeaters by default. You may find that you get better performance by disabling this feature on devices which you have not designated as a repeater. This can be done by issuing the /MESH OFF command.

Building WaveShark Communicator network applications

The WaveShark Communicator was built from the ground up to behave like a general purpose wireless mesh data modem. Programmatic control of the device is easily achieved using its extensive USB Serial interface.

Examples of programming languages that can be used to turn WaveShark Communicator nodes into “network applications” include Python, C, C++, C#, and Java.

Examples of network applications that we envision being created by either the WaveShark company or by WaveShark customers include:

A simple Bulletin Board System (BBS), message forum, or community “wall”

A program reads from the WaveShark Communicator's USB Serial port, waiting for messages that address the name of the BBS, forum, or wall and contain a command. For example:

<Bob> @TowerBBS READ NEW MESSAGES

<Bob> @TowerBBS ADD MESSAGE Hey y'all, I'll be getting into town tonight at 7:00 PM

The BBS, forum, or wall would then write to the USB Serial port, responding to commands from users:

<TowerBBS> @Bob I've added your message

An Internet gateway for linking various local WaveShark Communicator networks

A program reads from the WaveShark Communicator's USB Serial port, waiting for messages that address the name of the gateway and forwards these messages over an Internet TCP/IP connection to a computer in some other geographical area which is also connected to a WaveShark Communicator network. For example:

<Bob> @Tampa Situation is all good here in Gainesville. How are things in Tampa?

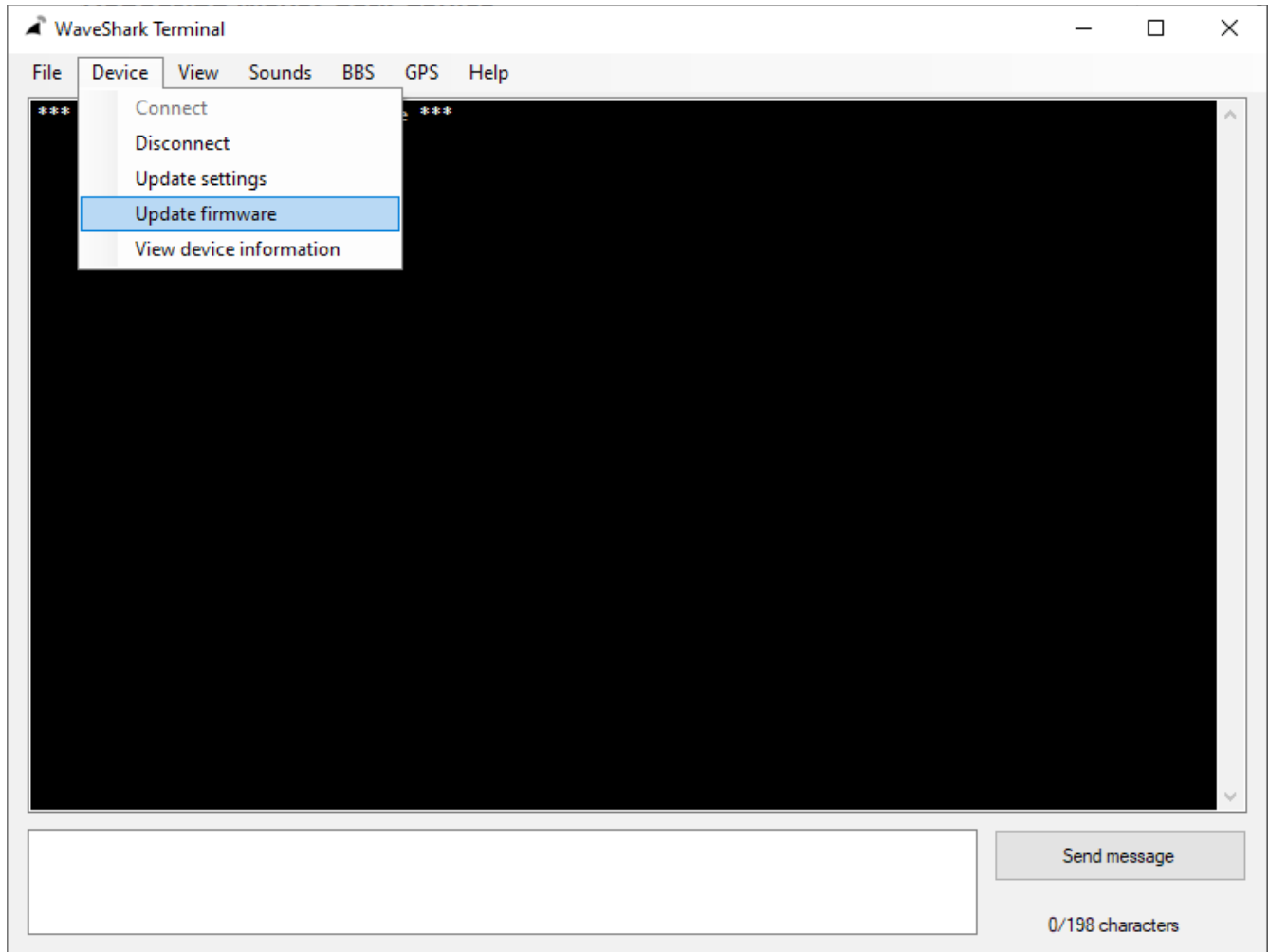
Sending telemetry data

A microcontroller (Arduino, etc), Single Board Computer (Raspberry Pi, etc), or any other type of computer is connected to various sensors and a WaveShark Communicator. The microcontroller / computer sends sensor data every 15 minutes to a receiving computer connected to a WaveShark Communicator.

Updating firmware

Windows PC

The easiest way to update your WaveShark Communicator's firmware on a Windows PC is using the free *WaveShark Terminal* software, which is available for download at www.waveshark.net. New firmware releases will be available for download at www.waveshark.net.



Mac

The WaveShark Communicator firmware can be updated on a Mac using the widely available *esptool.py* Python program. Please contact WaveShark at info@waveshark.net for assistance in updating your WaveShark Communicator's firmware on a Mac.

Linux

The WaveShark Communicator firmware can be updated on Linux using the widely available *esptool.py* Python program. Please contact WaveShark at info@waveshark.net for assistance in updating your WaveShark Communicator's firmware on Linux.

Amateur Radio operation

WARNING: U.S. Federal law generally prohibits radio broadcasts without a license issued by the FCC. Anyone found operating a radio station without FCC authorization can be subject to a variety of enforcement actions, including seizure of equipment, fines, and other civil and criminal penalties. Operating the WaveShark Communicator on U.S. Amateur Radio frequencies requires a valid Amateur Radio license with the appropriate privileges on the bands being transmitted on.

The WaveShark Communicator can transmit on the following U.S. Amateur radio bands / frequencies:

U.S. Amateur Radio band	Start frequency	End frequency
2m	144.000 MHz	148.000 MHz
70cm	420.000 MHz	450.000 MHz
33cm (complete overlap with 902-928 MHz U.S. ISM band)	902.000 MHz	928.000 MHz

Careful attention must be paid to the U.S. Amateur Radio band plans when operating on these frequencies so that data transmissions are only made on appropriate frequencies.

To enable operation on U.S. Amateur Radio frequencies, you must first issue the following command via the WaveShark Communicator's *USB Serial Interface*:

/IAMAHAM

This command will remind you that only licensed U.S. Amateur Radio operators are authorized to use this feature.

Once you have enabled operation on the U.S. Amateur Radio bands, you can use the **/HAMFREQ** command to tune to any frequency in the above table.

Usage of this command is as follows:

/FREQ <Frequency in MHz x 10,000>

For example, to tune to 146.520 MHz, you would issue the following command:

/FREQ 1465200

(146.5200 MHz x 10,000 = 1465200)

Appendix A: WaveShark Communicator *USB Serial Interface* commands

The following is a table of every command available via the WaveShark Communicator *USB Serial Interface*, along with an explanation of each command's intended usage.

Command	Arguments	Description
/HELP or /?		Get a listing of every available command
/SETTINGS		Get a listing of all current settings on your WaveShark Communicator device
/INBOX		Get messages waiting in device's inbox
/BT	ON, OFF	Enable or disable Bluetooth
/BT		Get current Bluetooth setting
/USEKEY	ON, OFF	Enable or disable use of message key / encryption key
/WIFI	OFF, CLIENT, AP	Disable Wi-Fi or enable Wi-Fi as a client (station) or Access Point (AP), respectively
/WIFI		Get current Wi-Fi setting and status
/WIFISSID	1-32 characters	Set Wi-Fi SSID for use as either a Wi-Fi client (station) or Wi-Fi Access Point (AP)
/WIFISSID		Get current Wi-Fi SSID
/WIFIPASS	8-63 characters	Set Wi-Fi password to use as either a Wi-Fi client (station) or Wi-Fi Access Point (AP)
/WIFIPASSCLEAR		Clear the Wi-Fi password
/WIFISSIDRESET		Reset the Wi-Fi SSID to the factory default value
/WIFIMAC		Get Wi-Fi MAC address
/USEKEY		Get current message key / encryption key setting
/KEY	4 to 16 characters	Set message key / encryption key
/NAME	3 to 20 characters	Set device name / sender name
/NAME		Get current device name / sender name
/BEACON	0, 60 to 3600	Disable Beacons (0) or set a Beacon to be sent every 60 to 3,600 seconds
/BEACON		Get current Beacon setting

/BEACONVOLTAGE	ON, OFF	Enable or disable the inclusion of the WaveShark Communicator's current USB and battery voltage levels in each Beacon sent
/BEACONVOLTAGE		Get current Beacon voltage setting
?BEACONVERSION	ON, OFF	Enable or disable the inclusion of the WaveShark Communicator's current firmware version in each Beacon sent
/BEACONVERSION		Get current Beacon version setting
/BEACONSIGREP	ON, OFF	Enable or disable the inclusion of a Signal Report request in each Beacon sent
/BEACONSIGREP		Get current Beacon Signal Report request setting
/BATTERY		Get current battery and USB voltage levels
/CHANNEL	1 to 100	Set WaveShark Communicator LoRa™ channel
/CHANNEL		Get current WaveShark Communicator LoRa™ channel
/TXPOWER	1 to 23	Set LoRa™ transmit power level
/TXPOWER		Get current LoRa™ transmit power level
/LEDS	ON, OFF	Enable or disable LEDs
/LEDS		Get current LEDs setting
/SIGREPREQ		Send a Signal Report request
/SIGREP	ON, OFF	Enable or disable responding to Signal Report requests
/SIGREP		Get current Signal Report request response setting
/SEROUT	NORMAL, FIELDTEST, DEBUG	Set <i>USB Serial interface</i> output level. NORMAL is the default output level and provides a minimum of extra information beyond showing messages received. FIELDTEST extends the NORMAL output level by adding SNR, RSS, and TTL readings to each message received. The DEBUG level provides everything in the NORMAL and FIELDTEST levels but also adds a massive amount of additional output that may be useful for diagnosing the performance of your device or network.
/SEROUT		Get current <i>USB Serial interface</i> output level

/MESH	ON, OFF	Enable or disable participation in the WaveShark Communicator mesh
/MESH		Get current mesh participation setting
/TTL	0 to 15	Disable movement of transmissions through the WaveShark Communicator mesh (0) or set the maximum number of hops (Time To Live) that your transmission should take through the mesh (1 to 15)
/TTL		Get current TTL setting
/TRACEROUTE	3 to 20 characters	Trace the route through the WaveShark Communicator mesh to another WaveShark Communicator device
/TRACEROUTERESP	ON, OFF	Enable or disable responses to TRACEROUTE requests
/TRACEROUTERESP		Get current TRACEROUTE responses setting
/REMOTECONTROL	ON, OFF	Enable or disable the ability for other WaveShark Communicator devices to control this WaveShark Communicator device remotely
/REMOTECONTROL		Get current Remote Control setting
/RCKEY	4 to 16 characters	Set Remote Control key
/RCTARGET	3 to 20 characters	Set target device name for Remote Control commands
/RCTARGET		Get current target device name for Remote Control commands
/RCCMD	1 to 200 characters	Send Remote Control command to current device name for Remote Control commands using current Remote Control key
/LOCALECHO	ON, OFF	Enable or disable local echo (this setting is not saved to the device and will revert back to ON after the device reboots)
/JSON	Settings JSON	Update settings from JSON
/JSON		Get current device configuration as JSON
/VERSION		Get current firmware version
/FACTORYRESET		Reset all device settings to factory defaults
/REBOOT		Reboot the device

Appendix B: WaveShark Communicator device settings

The following is a table of every configurable setting on a WaveShark Communicator and each setting's default value:

Setting	Range of acceptable values	Default value
Bluetooth on or off	ON, OFF	ON
Wi-Fi off, client (station), or Access Point (AP)	OFF, CLIENT, AP	OFF
Wi-Fi SSID	Any characters, 1 to 32 characters in length	WaveShark XXXXXX, where XXXXXX is the second half of the WaveShark Communicator's serial number
Wi-Fi password	Any characters, 8 to 63 characters in length, or an empty string (no password)	Empty string (no password)
Messaging key on or off	ON, OFF	OFF
Messaging key	Any characters, 4 to 16 characters in length	Random value
Device name	Any characters, 3 to 20 characters in length	WaveShark XXXXXX, where XXXXXX is the second half of the WaveShark Communicator's serial number
Beacon interval	0 = off, 60 to 3,600 seconds	0
Beacons include voltage levels	ON, OFF	OFF
Beacons include firmware version	ON, OFF	OFF
Beacons include Signal Report request	ON, OFF	OFF
WaveShark Communicator LoRa™ channel	1 to 100	42
LoRa™ Spreading Factor	7 to 12	10
LoRa™ transmit power	1 to 23	23
LEDs on or off	ON, OFF	ON
Respond to Signal Requests	ON, OFF	ON
USB Serial interface output level	NORMAL, FIELDTEST, DEBUG	NORMAL

Mesh participation	ON, OFF	ON
Mesh TTL	1 to 15	10
Remote Control on or off	ON, OFF	OFF
Remote Control key	Any characters, 4 to 16 characters in length	Random value
TRACEROUTE responses	ON, OFF	ON

Appendix C: WaveShark Communicator LoRa™ channels and frequencies

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	902.000	26	908.250	51	914.500	76	920.750
2	902.250	27	908.500	52	914.750	77	921.000
3	902.500	28	908.750	53	915.000	78	921.250
4	902.750	29	909.000	54	915.250	79	921.500
5	903.000	30	909.250	55	915.500	80	921.750
6	903.250	31	909.500	56	915.750	81	922.000
7	903.500	32	909.750	57	916.000	82	922.250
8	903.750	33	910.000	58	916.250	83	922.500
9	904.000	34	910.250	59	916.500	84	922.750
10	904.250	35	910.500	60	916.750	85	923.000
11	904.500	36	910.750	61	917.000	86	923.250
12	904.750	37	911.000	62	917.250	87	923.500
13	905.000	38	911.250	63	917.500	88	923.750
14	905.250	39	911.500	64	917.750	89	924.000
15	905.500	40	911.750	65	918.000	90	924.250
16	905.750	41	912.000	66	918.250	91	924.500
17	906.000	42	912.250	67	918.500	92	924.750
18	906.250	43	912.500	68	918.750	93	925.000
19	906.500	44	912.750	69	919.000	94	925.250
20	906.750	45	913.000	70	919.250	95	925.500
21	907.000	46	913.250	71	919.500	96	925.760
22	907.250	47	913.500	72	919.750	97	926.000
23	907.500	48	913.750	73	920.000	98	926.250
24	907.750	49	914.000	74	920.250	99	926.500
25	908.000	50	914.250	75	920.500	100	926.750

The default WaveShark LoRa™ channel is 42

Frequency (MHz) = $902 + (2 \times (\text{Channel} - 1) \times 0.125)$ MHz

LoRa™ bandwidth is 125 kHz

Appendix D: Enclosure fasteners

The WaveShark Communicator enclosure can be disassembled using a T8 Torx driver. Four longer 4-40 threaded screws fasten the front outside lid of the WaveShark Communicator to the enclosure and 3 shorter 4-40 threaded screws fasten the PCB to the inside of the enclosure. In the event that the device needs its battery replaced, it can be found underneath the PCB.

T8 Torx drivers can be easily and inexpensively purchased from companies such as *McMaster-Carr*.

NOTE: Only a genuine WaveShark Communicator battery should be used during battery replacement. Please contact WaveShark for more information. In the event that you must replace the battery without contacting us, it is absolutely imperative that you replace the battery with a LiPo battery of the same voltage as the factory battery and with an integrated Battery Management System (BMS) with under-charge and over-charge protection. Failure to do so can result in battery fire or explosion.