

Programming the Pebble HF

This guide covers how to update the firmware on your [Pebble HF](#).

There is more than one way to do it. We start with **the easy way** — a web-based updater that needs no software, no drivers, and no compiling. Other methods (using different programmers and avrdude on the command line) are documented further down for the more

technical crowd. The Arduino IDE may be used also, but is not covered here.

If you just want the simplest path, read **The Easy Way** and stop there.

Table of Contents

- [The Easy Way \(Web-Based Updater\)](#) — recommended; nothing to install.
 - [The Command-Line Way \(avrdude\)](#) — for the technically inclined.
 - [Factory Programming](#) — reference for how boards are programmed before shipping.
-

The Easy Way (Web-Based Updater)

This is a browser-based tool that flashes your Pebble HF in a few clicks:

- **No software to install**
- **No drivers to install**
- **No compiling or building anything**

You open a web page, click **Connect & Flash**, and it programs and verifies the firmware for you.

Web updater: <https://mspiceland.github.io/pebblehf-web-firmware-updater/>

What you need

1. **A Pebble HF with a programming header** (a 6-pin header). If yours doesn't have one yet, see [Programming header](#) below — you still have good options.
2. **WB2CBA's Atmega ISP Programmer** or an **STK500v1-compatible programming adapter**. This is the key piece. We

recommend one that shows up on your computer as a plain USB COM port, because that's what lets the web tool talk to it with **no drivers**:

- **"Atmega ISP Programmer" by WB2CBA** (*recommended — coming soon*). This is the adapter designed alongside the Pebble HF by Barb (WB2CBA), the hardware designer. Plug it into almost any computer and it just works as a USB serial port — no extra drivers, which is exactly what the web updater needs. It is **not available yet**, but we are working to make it available as soon as possible.
- **STK500v1-compatible programming adapter.**
These older Atmel-style serial programmers also speak STK500v1 and may work. Not tested yet.

Other cheap programmers like the **USBTinyISP** and **USBasp** can program the Pebble HF too, but they do **not** work with this web tool. Those are covered in the other methods below.

3. A supported computer and browser.

- Works on **Windows, macOS, Linux, and Chromebook.**
- The tool uses the browser's **Web Serial** feature, so you need a **Chromium-based browser: Chrome, Edge, Brave, or Opera.** (Safari and Firefox are not supported.)

Steps

1. **Connect the adapter to your computer** with its USB cable.
2. **Connect the other end to the programming header on your Pebble HF.** Make sure the pinout orientation is correct — double-check that the pin labels line up on both sides before powering on.
3. **Open the web updater:**
<https://mspiceland.github.io/pebblehf-web-firmware-updater/>
4. **Choose the firmware.** The page has a dropdown listing known-good firmware versions; the latest is selected and downloaded automatically. (If you have your own `.hex` file, choose **"Upload my own .hex file"** and drag it in.)
5. **Click "Connect & Flash"** and pick your adapter from the list your browser shows.

6. **Wait for the Writing and Verifying bars to finish.** When it reports success, the new firmware is on your Pebble HF. You can disconnect everything.

That's the whole process.

Programming header

The Pebble HF programs through a small **6-pin header**. A few notes:

- **New kits will include the header.** The easiest thing is to solder it on while you build the radio.
- **Already built yours without a header? You have options:**
 - **Tack on a 6-pin header.** Hold a 6-pin header against the side of the board and carefully solder each pin. It sticks up a little more, but it works fine.
 - **Use a "pogo-pin" adapter.** These spring-loaded pin headers press onto the board only while you program — nothing to solder. They're inexpensive (around \$12) and very convenient. Examples: [Amazon](#) • [eBay](#).

Troubleshooting

- **The browser shows "Web Serial is not available."** You're using a non-Chromium browser. Switch to Chrome, Edge, Brave, or Opera.
- **Your adapter doesn't appear in the list.** Make sure it's plugged in and that it's an STK500v1 serial adapter that enumerates as a COM/serial port.
- **The firmware download fails.** You can still program: download the `.hex` from the [releases page](#), then choose **"Upload my own .hex file"** in the web tool.
- **Verify fails or it can't sync.** Re-check the programming header orientation and that all 6 pins are making good contact, then try again.

The Command-Line Way (avrdude)

*Prefer the simplest route? Use [The Easy Way](#) above instead — this section is for the technically inclined. The Arduino IDE can also be used, but isn't covered here. Until the WB2CBA adapter is available, several users on Discord (link is at the bottom of <https://learnmorsecode.org>) who have **USBtinyISP** and **USBasp** adapters have offered to help others get set up.*

This section walks you through loading a new firmware version onto your Pebble HF using a small programmer and a single command. It looks technical, but it really comes down to **plug in a programmer, type one line, press Enter**. No coding, no compiling, nothing to figure out.

Your Pebble HF already came fully programmed and working. This is only for *updating* it to a newer firmware version — so there is nothing to set up on the radio itself and no way to "break" the chip's basic settings. You are only replacing the program that runs on it.

You'll use a free tool called **avrdude**. It's the same tool that every Arduino-style program uses behind the scenes — we're just skipping all the extra software around it and talking to it directly. That's what keeps this simple.

The 60-Second Roadmap

Here's the whole process before we get into details:

1. **Get a programmer** — a small USB gadget that plugs into your radio's programming header.
2. **Download the firmware** — one `.hex` file from our Releases page.
3. **Install avrdude** — the free tool that does the actual work.
4. **One-time setup for your computer** — Windows needs a quick driver step; Mac and Linux are basically plug-and-play.
5. **Connect the programmer to the Pebble HF** — line up one cable, mind the orientation.
6. **Run one command** — copy the line for your programmer, press Enter, done in about two minutes.

That's it. Everything below is just these six steps spelled out.

Step 1 — Get a Programmer

You need an **ISP programmer**. This guide supports three common types — any one of them works:

Programmer	Where people get it	Notes
USBasp	eBay, AliExpress, Amazon (search "USBasp")	Cheapest and most common. Inexpensive clones work fine.
USBtinyISP (<i>incl. the SparkFun Pocket Programmer</i>)	SparkFun, Adafruit, eBay	Works great; on Windows use the driver step below (not SparkFun's own driver).
stk500v1 (<i>e.g. the "Atmel ISP Programmer by WB2CBA"</i>)	Various / our own programmer	A serial-style programmer that shows up as a COM/serial port.

A few things to look for when ordering:

- You want a **6-pin programming cable** (a 2×3 connector). Many cheap programmers ship a 10-pin cable instead — if so, you'll also want a **10-pin-to-6-pin adapter**, or a cable that has both connectors on it.
- USBasp boards have a small **voltage jumper (5V / 3.3V)**. Make sure it's set to **5V**. Most arrive that way, but check.

Tip: If you're buying from overseas marketplaces, expect shipping delays.

Step 2 — Download the Firmware

1. Go to the Releases page:
<https://github.com/mspiceland/usdx-pebblehf/releases>
2. Pick the version you want (the newest stable release is at the top).
3. Under that release's **Assets**, download the **.hex** file. Save it somewhere easy to find — your **Downloads** folder is perfect.

About the filename: Whatever the file is named when you download it, that's the name you'll type later. It doesn't matter if the name is long or doesn't end in `.hex` — the command below handles that automatically. In the examples we call it `firmware.hex`; just substitute the real name of the file you downloaded.

Step 3 — Install avrdude

Pick your operating system.

Windows

1. Go to <https://github.com/avrdudes/avrdude/releases>
2. Download the file named `avrdude-<version>-windows-x64.zip` (the newest one; pick `x64` for a normal PC).
3. **Extract** the zip (right-click → *Extract All*). You'll get a folder containing `avrdude.exe` and `avrdude.conf`.
4. **Copy your downloaded `firmware.hex` into that same folder.** Keeping avrdude and the firmware together means you won't have to fuss with file paths later.

That's the whole "install" — there's nothing to run yet.

Mac

Open the **Terminal** app (in *Applications* → *Utilities*) and install avrdude with Homebrew:

```
brew install avrdude
```

If you don't have Homebrew yet, paste this first, then run the line above:

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Linux

Open a terminal and install avrdude with your package manager:

```
# Debian / Ubuntu / Raspberry Pi OS
sudo apt update && sudo apt install avrdude

# Fedora
sudo dnf install avrdude

# Arch
sudo pacman -S avrdude
```

Step 4 — One-Time Computer Setup

This is the only step that differs much between operating systems. Do it once and you never have to think about it again.

Windows

If you're using a USBasp or USBtinyISP (including the SparkFun Pocket Programmer), Windows won't know what the device is until you give it the right driver. The reliable tool for this is **Zadig**:

1. Plug in your programmer. Windows may show it as an unknown device with a yellow "!" — that's expected.
2. Download Zadig from <https://zadig.akeo.ie/> and run it (no installation needed).
3. In Zadig, from the dropdown, select your programmer (it'll be listed as **USBasp** or **USBtinyISP**, or as an unknown device). If you don't see it, choose **Options → List All Devices**.
4. Set the driver to **libusbK**.
5. Click **Install Driver** (or *Replace Driver*).
6. Unplug the programmer and plug it back in.

That's it — Windows now recognizes it.

The SparkFun Pocket Programmer's own driver can refuse to install on modern Windows (an unsigned-driver warning). The Zadig **libusbK** step above sidesteps that completely, so use it for the Pocket Programmer too.

If you're using an stk500v1 programmer (the WB2CBA "Atmel ISP Programmer"), you do **not** use Zadig. It shows up as a **COM port** instead. Modern Windows usually installs the needed USB-serial driver automatically. To find your COM port number:

1. Plug in the programmer.
2. Open **Device Manager** → expand **Ports (COM & LPT)**.
3. Note the **COM** number (e.g. COM5). You'll use it in the command later.

Mac

Nothing to install. USBasp and USBtinyISP programmers work as soon as they're plugged in.

- **For an stk500v1 programmer**, you'll need its port name. Plug it in, then run this in Terminal and look for an entry like `/dev/cu.usbserial-XXXX`:

```
ls /dev/cu.*
```

Linux

USBasp and USBtinyISP work out of the box, but Linux restricts USB access by default, so you'll either run the command with `sudo`, or add a one-time rule so you don't need `sudo`. The simplest path is just to put `sudo` in front of the command in Step 6.

- **For an stk500v1 programmer**, find its port name. Plug it in, then run:

```
ls /dev/ttyUSB* /dev/ttyACM*
```

It'll usually be `/dev/ttyUSB0`. If you get a "permission denied" error later, either use `sudo` or add yourself to the `dialout` group with `sudo usermod -aG dialout $USER` (then log out and back in).

Step 5 — Connect the Programmer to the Pebble HF

Use the [Pebble HF programming pinout image](#) as your reference — it shows exactly where the 6-pin programming header is and which way pin 1 faces.

Three things to get right:

1. **Orientation (the important one)**. The cable only works one way around. Match **pin 1** of the cable to **pin 1** on the Pebble HF header. On a ribbon cable, pin 1 is usually the wire

with the **red (or marked) stripe**. On the board, pin 1 is marked on the pinout image. If programming fails later, flipping the connector 180° is the first thing to try.

2. **Use the 6-pin connector.** If your programmer only has a 10-pin cable, use a 10-to-6 adapter (see Step 1).
3. **Power.** The programmer supplies power to the radio through this cable, so **do not power the Pebble HF from its own battery or power supply while programming.** (On a USBasp, this is also why the 5V/3.3V jumper should be on **5V**.)

Plug the programmer into your computer's USB port, and the cable into the Pebble HF.

Step 6 — Run the Command

Almost done. Open a terminal, go to where your firmware file is, and run the one line that matches your programmer.

Open a terminal in the right place

Windows: Open the avrdude folder (the one where you put `firmware.hex` in Step 3). Click in the **address bar** at the top of the File Explorer window, type `powershell`, and press **Enter**. A PowerShell window opens already pointing at that folder.

Mac / Linux: Open Terminal and move into your Downloads folder:

```
cd ~/Downloads
```

Type your command

Find the line for your programmer below. In every command, replace `firmware.hex` with the exact name of the file you downloaded.

USBasp

```
# Windows
.\avrdude.exe -c usbasp -p m328p -U flash:w:firmware.hex:i

# Mac / Linux (add sudo on Linux if you hit a permission error)
avrdude -c usbasp -p m328p -U flash:w:firmware.hex:i
```

USBtinyISP (including the SparkFun Pocket Programmer)

```
# Windows
.\avrdude.exe -c usbtiny -p m328p -U flash:w:firmware.hex:i

# Mac / Linux (add sudo on Linux if you hit a permission error)
avrdude -c usbtiny -p m328p -U flash:w:firmware.hex:i
```

stk500v1 (the WB2CBA "Atmel ISP Programmer") — this one also needs your **port** from Step 4:

```
# Windows (replace COM5 with your COM number)
.\avrdude.exe -c stk500v1 -p m328p -P COM5 -b 19200 -U
flash:w:firmware.hex:i

# Mac (replace the port with your /dev/cu.usbserial-XXXX name)
avrdude -c stk500v1 -p m328p -P /dev/cu.usbserial-XXXX -b 19200 -U
flash:w:firmware.hex:i

# Linux (replace the port with your /dev/ttyUSB0 name)
avrdude -c stk500v1 -p m328p -P /dev/ttyUSB0 -b 19200 -U
flash:w:firmware.hex:i
```

What's in the command, in plain English: -c is your programmer type, -p m328p is the chip inside the Pebble HF, and -U flash:w:firmware.hex:i means "write this file to the radio." The :i on the end is what lets you use whatever filename you downloaded.

Press **Enter**. You'll see a progress bar or two, and after a minute or so it finishes with something like:

```
avrdude: 25000 bytes of flash written
avrdude: 25000 bytes of flash verified

avrdude done. Thank you.
```

When you see **flash verified** and **avrdude done. Thank you.**, the update worked. Unplug the programmer, power up your Pebble HF, and check the version on the startup/menu screen.

If Something Goes Wrong

Most problems are one of these, and all are easy fixes:

- **avrdude: warning: cannot set sck period (USBasp).** This is harmless — older USBasp clones just can't report their speed. If the rest of the output shows **flash verified**, you're done; ignore the warning.
- **initialization failed / target doesn't answer / rc=-1.** The programmer can't talk to the chip. In order, check:
 1. Cable **orientation** — flip the 6-pin connector 180° (this is the #1 cause).
 2. USBasp **voltage jumper** set to **5V**.
 3. Cable fully seated on both ends.
 4. If it still won't connect on a USBasp, slow it down by adding **-B 20** right after **-c usbasp**, e.g. **... -c usbasp -B 20 -p m328p ...**.
- **could not find USB device (USBasp / USBtinyISP on Windows).** The driver step was missed — go back to Step 4 and run **Zadig** to install the **libusbK** driver.
- **can't open device / programmer is not responding (stk500v1).** Usually the wrong port or a port that's in use. Re-check the **COM/ /dev** name from Step 4, and close any other program that might be using it (like the Arduino IDE or a serial monitor). Make sure **-b 19200** is in the command.
- **Permission denied (Linux).** Put **sudo** in front of the command, or add yourself to the **dialout** group (see Step 4).
- **'avrdude' is not recognized / command not found.** On Windows, make sure you're in the folder that contains **avrdude.exe** and used **.\avrdude.exe**. On Mac/Linux, re-check that Step 3 installed successfully.

If you're still stuck, copy the full text avrdude printed and send it along — that output almost always pinpoints the problem.

Factory Programming

Every Pebble HF is fully programmed and tested before it ships. That includes both writing the firmware **and** setting the microcontroller's fuse bytes. This section is here for reference only — as an owner, you should **never need to touch the fuses**.

When you update your radio later (via [The Easy Way](#) or [The Command-Line Way](#)), you are only replacing the firmware. The fuses were set correctly at the factory and stay that way, so the update tools don't change them.

How it's done

Factory programming is performed with the `utils/flash_pebble.py` script in this repository. It auto-detects the USB serial programmer, then flashes the firmware, programs the fuses, and reads the fuses back twice to verify them:

```
python3 utils/flash_pebble.py
```

Run `python3 utils/flash_pebble.py --help` for options (such as specifying the `.hex` file or serial port).

Fuse settings (reference)

The script programs these fuse values for the ATmega328P (shown here only for reference — these are set at the factory and should not need to be changed again):

Fuse	Value
Low (<code>lfuse</code>)	<code>0xFF</code>
High (<code>hfuse</code>)	<code>0xD6</code>
Extended (<code>efuse</code>)	<code>0xFD</code>

If you ever did need to set them manually with a stk500v1 programmer, the equivalent command is:

```
avrdude -c stk500v1 -p m328p -P /dev/cu.usbserial-XXXX -b 19200 \  
-U lfuse:w:0xFF:m \  
-U hfuse:w:0xD6:m \  
-U efuse:w:0xFD:m
```