
Flash via AN945: EFM8 Factory Bootloader HID.

Apr 26, 2022

Contents:

1	Usage	3
1.1	Installation	3
1.2	efm8	3
1.3	efm8_read	4
1.4	u2fzero	5
2	Indices and tables	7
	Python Module Index	9
	Index	11

Flash via AN945: EFM8 Factory Bootloader HID.

CHAPTER 1

Usage

Communication is over USB-HID. This is implemented via the `hidapi` python wrapper for the `hidapi` native library.

On linux you can use udev to grant access:

```
echo 'SUBSYSTEM=="usb", ATTRS{idVendor}=="10c4", MODE="0666"' | sudo tee /etc/udev/  
rules.d/70-silabs.rules  
udevadm trigger
```

```
efm8 firmware.hex
```

Also includes an example that resets a <https://u2fzero.com/> into the bootloader and flashes in one command.

```
u2fzero firmware.hex
```

And a way to (slowly) read the firmware back

```
efm8_read firmware.hex
```

1.1 Installation

```
sudo apt install libusb-1.0-0-dev libudev-dev  
python3 -m pip install efm8
```

1.2 efm8

Flash via AN945: EFM8 Factory Bootloader HID.

```
usage: efm8 [-h] [-p PRODUCT] [-s SERIAL] firmware
```

firmware

Intel Hex format file to flash

-h, --help

show this help message and exit

-p <product>, --product <product>

USB Product ID of device to program

-s <serial>, --serial <serial>

Serial number of device to program

The protocol is documented in [AN945](#).

Flash via AN945: EFM8 Factory Bootloader HID.

exception efm8.BadChecksum

Checksum mismatch.

exception efm8.BadResponse

Command not confirmed.

exception efm8.Unsupported

Input file not understood.

efm8.crc (data)

CITT-16, XModem.

efm8.create_frame (cmd, data)

Bootloader frames start with '\$', 1 byte length, 1 byte command, x bytes data.

efm8.flash (manufacturer, product, serial, frames)

Send bootloader frames over HID, and check confirmations.

efm8.read_flash (manufacturer, product, serial, length)

Exploit CRC to read back firmware.

efm8.read_intel_hex (filename)

Read simple Intel format Hex files into byte array.

efm8.to_frames (data, checksum=True, run=True)

Convert firmware byte array into sequence of bootloader frames.

efm8.toaddr (addr)

Split a 16bit address into two bytes (doesn't check it is a 16bit address ;-).

efm8.tostr (buf)

Ensure we have str across python versions.

efm8.twos_complement (input_value, num_bits=8)

Calculate unsigned int which binary matches the two's complement of the input.

efm8.write_hex (buf, filename)

Write an Intel Format Hex file.

1.3 efm8_read

Flash via AN945: EFM8 Factory Bootloader HID.


```
usage: efm8_read [-h] [-p PRODUCT] [-s SERIAL] [-l LENGTH] firmware
```

firmware

Intel Hex format file to flash

-h, --help

show this help message and exit

-p <product>, --product <product>

USB Product ID of device to program

-s <serial>, --serial <serial>

Serial number of device to program

-l <length>, --length <length>

Length to read

1.4 u2fzero

Extra utils for U2F-Zero devices.

```
usage: u2fzero [-h] [-p PRODUCT] [-s SERIAL] firmware
```

firmware

Intel Hex format file to flash

-h, --help

show this help message and exit

-p <product>, --product <product>

USB Product ID of device to program

-s <serial>, --serial <serial>

Serial number of device to program

Extra utils for U2F-Zero devices.

`efm8.u2fzero.main()`

Command line.

`efm8.u2fzero.reset` (*manufacturer, product, serial*)

Send zeroU2F jump to bootloader, trigger the host to see the device change.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

e

`efm8`, 4

`efm8.u2fzero`, 5

Symbols

-h, -help
 efm8 command line option, 4
 efm8_read command line option, 5
 u2fzero command line option, 5
 -l <length>, -length <length>
 efm8_read command line option, 5
 -p <product>, -product <product>
 efm8 command line option, 4
 efm8_read command line option, 5
 u2fzero command line option, 5
 -s <serial>, -serial <serial>
 efm8 command line option, 4
 efm8_read command line option, 5
 u2fzero command line option, 5

B

BadChecksum, 4
 BadResponse, 4

C

crc() (in module *efm8*), 4
 create_frame() (in module *efm8*), 4

E

efm8 (module), 4
 efm8 command line option
 -h, -help, 4
 -p <product>, -product <product>, 4
 -s <serial>, -serial <serial>, 4
 firmware, 3
 efm8.u2fzero (module), 5
 efm8_read command line option
 -h, -help, 5
 -l <length>, -length <length>, 5
 -p <product>, -product <product>, 5
 -s <serial>, -serial <serial>, 5
 firmware, 5

F

firmware
 efm8 command line option, 3
 efm8_read command line option, 5
 u2fzero command line option, 5
 flash() (in module *efm8*), 4

M

main() (in module *efm8.u2fzero*), 5

R

read_flash() (in module *efm8*), 4
 read_intel_hex() (in module *efm8*), 4
 reset() (in module *efm8.u2fzero*), 5

T

to_frames() (in module *efm8*), 4
 toaddr() (in module *efm8*), 4
 tostr() (in module *efm8*), 4
 twos_complement() (in module *efm8*), 4

U

u2fzero command line option
 -h, -help, 5
 -p <product>, -product <product>, 5
 -s <serial>, -serial <serial>, 5
 firmware, 5
 Unsupported, 4

W

write_hex() (in module *efm8*), 4