
EFM8 Bootloader

Release 0.0.2

Nov 05, 2021

Contents:

1	Installation	3
2	efm8	5
3	efm8_read	7
4	u2fzero	9
5	Indices and tables	11
	Python Module Index	13
	Index	15

Talks to the factory Bootloader on EFM8 to flash firmware. The protocol is documented in [AN945](#).

CHAPTER 1

Installation

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
```

Then install some native prerequisites:

```
sudo apt install libusb-1.0-0-dev libudev-dev python-dev
```

Then *pip* install:

```
pip install efm8
```


CHAPTER 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

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.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.

CHAPTER 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

CHAPTER 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 5

Indices and tables

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

e

`efm8`, 5

`efm8.u2fzero`, 9

Symbols

-h, -help
 efm8 command line option, 5
 efm8_read command line option, 7
 u2fzero command line option, 9
 -l <length>, -length <length>
 efm8_read command line option, 7
 -p <product>, -product <product>
 efm8 command line option, 5
 efm8_read command line option, 7
 u2fzero command line option, 9
 -s <serial>, -serial <serial>
 efm8 command line option, 5
 efm8_read command line option, 7
 u2fzero command line option, 9

B

BadChecksum, 5
 BadResponse, 5

C

crc() (in module *efm8*), 5
 create_frame() (in module *efm8*), 5

E

efm8 (module), 5
 efm8 command line option
 -h, -help, 5
 -p <product>, -product <product>, 5
 -s <serial>, -serial <serial>, 5
 firmware, 5
 efm8.u2fzero (module), 9
 efm8_read command line option
 -h, -help, 7
 -l <length>, -length <length>, 7
 -p <product>, -product <product>, 7
 -s <serial>, -serial <serial>, 7
 firmware, 7

F

firmware
 efm8 command line option, 5
 efm8_read command line option, 7
 u2fzero command line option, 9
 flash() (in module *efm8*), 5

M

main() (in module *efm8.u2fzero*), 9

R

read_flash() (in module *efm8*), 5
 read_intel_hex() (in module *efm8*), 6
 reset() (in module *efm8.u2fzero*), 9

T

to_frames() (in module *efm8*), 6
 toaddr() (in module *efm8*), 6
 twos_complement() (in module *efm8*), 6

U

u2fzero command line option
 -h, -help, 9
 -p <product>, -product <product>, 9
 -s <serial>, -serial <serial>, 9
 firmware, 9
 Unsupported, 5

W

write_hex() (in module *efm8*), 6