FAST EnergyCam Protocol Firmware Update

Table of Contents

Index of Tables

Introduction

FAST EnergyCam device firmware can be updated via Modbus communication. The update process is fail safe, even a power loss during the update procedure will not lead to a damaged EnergyCam. During the update process the image being transferred is first stored in a certain area in EnergyCam device internal memory and once the image has been completely transferred and successfully checked for consistency the installation takes place.

The transfer of the update image is split into chunks which each fit into a single Modbus frame. All meta information needed for the update procedure is part of the update image itself. Only a single binary file has to be transferred.

The needed Modbus registers are (see *FAST_EnergyCam-Protocol-MODBUS-Slave.pdf* for further information):

- holding register: UpdateChunk
- input register: UpdateCRCOK

The update procedure is split into three phases:

1. Transfer initial chunk which contains only the header information
2. Transfer the complete update image (which contains the header again).
3. Check consistency of transferred image (installation starts automatically when valid)

Protocol

Since each chunk is transported with standard Modbus framing which includes CRC and response back to host a secure data transmission is established. In case EnergyCam does not respond to a transmitted chunk the host should re-transmit the same chunk once again. Since a ChunkStartAddress comes along with the ChunkData even a double transmission does not hurt.

Since the update image is stored in an internal flash memory a certain flash timing has to be maintained. The Modbus responses are synchronous to these flash timings which makes it easy to transfer the update image as fast as possible.

The update procedure phases in detail:

1. **Transfer initial chunk**
   To start the update procedure transfer the update image header with ChunkStartAddress = 0 and

exactly 40 bytes (20 words). The header consists of the very first 40 bytes of the update image. This will lead to a flash erase which lasts several seconds. Upon completeness a Modbus response is returned. When the header contains unexpected data a Modbus exception is returned.

2. **Transfer complete update image**
   The update image is split into several chunks and has to be transferred chunk-wise. Start again with the data of the initial chunk, e.g. the header has to be transferred again, but ChunkStartAddress should start now with 40 (dec). Max length of ChunkData comprises 240 bytes (120 words). Byte count of ChunkData could be shorter than 240 bytes but has to be a multiple of 4. Except the very last chunk which does not has to be a multiple of 4. When the chunk is written to flash a Modbus response is returned. When no response (after timeout) or a Modbus exception is returned this chunk has to be repeated with same ChunkStartAddress.
3. **Check consistency**
   When all chunks are successfully transferred consistency has to be checked. When input register UpdateCRCOK returns 1 the transfer was successful and the installation takes place automatically. This step leads to a reboot of EnergyCam. When boot has completed input registers AppBuildnumber should return the updated build number. When input register UpdateCRCOK returns 0 the update image transfer was wrongly performed.

These different timings have to be expected:

Flash operation	typ. timing [ms]	Comment
Image erase	5000	Takes place when header is written with ChunkStartAddress = 0
Write chunk	100	Takes place for all other chunks

Table 1: Flash operation timings

Chunk layout

Note that as normal in Modbus protocols all words (16 bit) transported are coded as big endian.

Word number (16bit)	meaning	Comment
0	ChunkStartAddress1	Update chunk. Chunk has to be written in a single frame consisting of up to 122 words. Byte address (32 bit) of first binary data following relative to first byte in update image. Very first byte of update image has address 0. Amount of ChunkData in bytes have to be a multiple of 4, except for very last chunk! Maximum is 120 words = 240 bytes (240 mod 4=0)
1	ChunkStartAddress0	
2	ChunkData (word0)	
...	ChunkData (word...)	
121	ChunkData (word119)	

Table 2: Chunk layout

Upgrade policy

Please note that only upgrades are possible, meaning that the build number of the update image has to be higher that the one currently installed. Updating with very same build number is only possible when the firmware type differs (see input register AppFirmwareType which currently distinguishes wM-Bus stacks with protocol version S2 and T2).

Note that an update image with smaller build number (which would be a down grade) can be successfully transferred, input registers UpdateCRCOK will return true but installation will fail, meaning that after the automatically performed reboot the build number is not altered.

Reference implementation

Please have a look at the publicly available reference implementation on

https://github.com/ffcrg/ecpi

History

Date	Author	Version	Changes
12th Jun 12	SPR	0.1	Initial
26th Jun 12	SPR	0.2	Reviewed by FBL
17th Jun 14	SPR	0.3	Cosmetic
28th Jun 14	SPR	1.0	Unification of naming
25th Mar 14	SPR	1.1	Corrected filenames of referenced .pdf

Table 3: History