# SMA NET template

Joachim Bürmann, *IFTOOLS GmbH*

November 1, 2016

**Th SMA-Net is a proprietary protocol developed by SMA, one of the largest manufacturers of photovoltaic inverters in the world. It is compatible inter alia with TCP/IP, PPP (Point-to-Point Protocol, RFC 1662) and HDLC (High Level Data Link Control, ISO 13239).**

## Frame structure

**Table 1:** *SMA NET frame format*

| Frame | | | Content | | Frame | |
|---|---|---|---|---|---|---|
| Start | Addr | Control | Protocol | Data | Checksum | Stop |
| 1 Byte | 1 Byte | 1 Byte | 2 Bytes | 7-262 Bytes | 2 Bytes | 1 Byte |
| 7E | FF | 03 | | | | 7E |

- **Start** Indicates the start of a new telegram and is always hex 7E
- **Address** The real device address is encapsulated in the SMA Data content and since the frame is sent to every device set to the broadcast address FF.
- **Control** The field is always set to 03 (unnumbered information)
- **Content** An encapsulate frame. The first two bytes contain the protocol identifier used by the frame. 0x4041 specifies a SMA Data telegram.
- **Checksum** The CRC16 checksum in little-endian format.
- **Stop** Indicates the stop of the current telegram and is always hex 7E

### Escape character

Since the start and stop character 0x7E must not be used in the data content, an escape character method is applied which substitutes the 7E with the sequence 0x7D 0x5E (in detail 0x7D followed by the replacing byte XOR-ed with 0x20). Beside the 0x7E and 0x7D itself the following characters or bytes are also specified as to be escaped: 0x11 (XON), 0x12 (DC2) and 0x13 (XOFF).

### Checksum

The checksum is a CRC16 with polynomial hex 81021: $x^{16} + x^{12} + x^5 + 1$
It is calculated over the address, control and content (protocol and data) and has to be applied to the original sequence (before any escape substitution).

```
./pycrc.py --width=16 \\
    --reflect-in=true --reflect-out=true \\
    --xor-in=0xFFFF --xor-out=0xFFFF \\
    --poly=0x81021 --generate=c
    --algorithm=table-driven > ~/Desktop/smanet.c
```

### The SMA Data telegram format

**Table 2:** *SMA Data telegram format*

| Header | | | | | Data |
|---|---|---|---|---|---|
| Src | Dest | Ctrl | Pkt Cnt | Cmd | Data |
| 2 Bytes | 2 Bytes | 1 Byte | 1 Byte | 1 Byte | 0-255 Bytes |

- **Src (Source)** Telegram sender address.
- **Dest (Destination)** Telegram receiver or receiver group address.

- **Ctrl (Control)** The bits of this byte indicates various control signals.

| | |
|---|---|
| Bit 7 | Determines if the destination is a group (1) or single (0) address |
| Bit 6 | Marks the telegram as a Request (0) or Response (1) telegram |
| Bit 5 | Reserved |
| Bit 4 | Set when the device blocks (1) the string of function |
| Bit 0..3 | Reserved |

- **Pkt Cnt (Packet count)** Used when the requested data cannot fit in one telegram. In this case the responding device has to set this field with the remaining packets, which then must request in further inquiries.
- **Cmd** The command the receiver has to perform. The available commands are listed below.
- **Data** The data of the telegram. This field can be 0...255 bytes long and depends on the belonging command.

### Commands

#### CMD_SEARCH_DEVICE

Performs a search for the device with the given serial number. The number is coded as a 4-Byte long in little endian order. The group bit has to be set to 1 since the command is broadcast to all devices, but only the according device has to response.
The data in the response is 12 bytes long and contains the serial number (4 bytes) and the device type (8 bytes and filled with ASCII null when smaller).

#### CMD_GET_DATA

This command requests data from a device. The requested data is described in a 2-byte transfer field (or mask) followed by a channel index (1 byte).
The response fields are taken from[1]. The transfer mask and index are identically with the request. According to [1] exist the following data fields:

| | |
|---|---|
| Transfer Mask | Byte 3,4 (identical with the request) |
| Time stamp | Byte 5...8 (in Unix epochal time) |
| Time base | Byte 9...12 (in seconds) |

#### CMD_SET_DATA

Sends data to any SMA Data network participant. This involves certain parameters, counter values, digital outputs and operating modes. The content is defined by the transfer mask and the receiver response with the same transfer mask and the number of the received data records (as a word).

**Table 3:** *The structure of the transfer mask*

| Bit(s) | Description | Group |
|---|---|---|
| 0 | Analog | Signal type |
| 1 | Digital | |
| 2 | Counter | |
| 3 | Status | |
| 4-7 | Reserve (=0) | |
| 8 | Input | Signal group |
| 9 | Output | |
| 10 | Parameter | |
| 11 | Instant. values | |
| 12 | Archivs data | |
| 13 | Test channel | |
| 15,16 | Reserve (=0) | |

## References

[1] Christopher Tapper. Low Cost 3G Enabled Data Logger for Photovoltaic Systems, Thesis for Bachelor of Engineering, Australian National University, 2014