|  |
| --- |
| The company |
| MODBUS Translator |
| Message Specification |
|  |
| **Marc Biundo** |
|  |

|  |
| --- |
|  |

|  | **SunSaver MPPT Message Layout** | | | |
| --- | --- | --- | --- | --- |
| **ModBus**  **Logical**  **Addr** | **Item name** | **Item description** | **Units** | **Scaling / Range /**  **Example Data** |
| - | Message ID | $PMPPTDAT | - | - |
| - | FW Version | Firmware Version |  | 1.0.0 |
| 9 | Adc\_vb\_f | Battery voltage, filtered (τ ≈ 1s) | V | n·100·2-15 |
| 10 | Adc\_va\_f | Array voltage, filtered (τ ≈ 1s) | V | n·100·2-15 |
| 11 | Adc\_vl\_f | Load voltage, filtered (τ ≈ 1s) | V | n·100·2-15 |
| 12 | Adc\_ic\_f | Charging current, filtered (τ ≈ 1s) | A | n·79.16·2-15 |
| 13 | Adc\_il\_f | Load current, filtered (τ ≈ 1s) | A | n·79.16·2-15 |
| 14 | T\_hs | Heatsink temperature | ºC | -127 to+127 |
| 15 | T\_batt | Battery temperature (Either ambient or RTS if connected) | ºC | -127 to+127 |
| 16 | T\_amb | Ambient temperature | ºC | -127 to+127 |
| 17 | T\_rts | Remote battery temperature sensor (0x80 if not connected) | ºC | -127 to+127 |
| 18 | charge\_state | Charge state - | - |  |
| 19 | array\_fault | Array fault bit field - | - |  |
| 20 | Vb\_f | Battery voltage, slow filter (τ ≈ 25s) | V | n·100·2-15 |
| 21 | Vb\_ref | Battery regulator reference voltage | V | n·96.667·2-15 |
| 22 | Ahc\_r\_HI | Ah charge resettable, HI word | Ah | n·0.1 |
| 23 | Ahc\_r\_LO | Ah charge resettable, LO word - | - |  |
| 24 | Ahc\_t\_HI | Ah charge total, HI word | Ah | n·0.1 |
| 25 | Ahc\_t\_LO | Ah charge total, LO word - | - |  |
| 26 | kWhc | KWh charge (resettable?) | kWh | n·0.1 |
| 27 | load\_state | Load state - | - |  |
| 28 | load\_fault | Load fault bit field - | - |  |
| 29 | V\_lvd | load current compensated LVD voltage | V | n·100·2-15 |
| 30 | Ahl\_r\_HI | Ah load resettable, HI word | Ah | n·0.1 |
| 31 | Ahl\_r\_LO | Ah load resettable, LO word - | - |  |
| 32 | Ahl\_t\_HI | Ah load total, HI word | Ah | n·0.1 |
| 33 | Ahl\_t\_LO | Ah load total, LO word - | - |  |
| 34 | hourmeter\_HI | hour meter, HI word | h | 0 to (224-1) |
| 35 | hourmeter\_LO | hour meter, LO word - | - |  |
| 36 | alarm\_HI | alarm bit field – HI word - | - |  |
| 37 | alarm\_LO | alarm bit field – LO word - | - |  |
| 38 | dip\_switch | dip switch settings at power on switch[1..8] in bits[0..7]- | - |  |
| 39 | led\_state | SOC LED state - | - |  |
| 40 | Power\_out | Charge output power | W | n·989.5·2-16 |
| 41 | Sweep\_Vmp | Array Vmp found during sweep | V | n·100·2-15 |
| 42 | Sweep\_Pmax | Arrap Pmax(output) found during sweep | W | n·989.5·2-16 |
| 43 | Sweep\_Voc | Array Voc found during sweep | V | n·100·2-15 |
| 44 | Vb\_min\_daily | Vb minimum voltage – daily | V | n·100·2-15 |
| 45 | Vb\_max\_daily | Vb maximum voltage – daily | V | n·100·2-15 |
| 46 | Ahc\_daily | Ah charge - daily | Ah | n·0.1 |
| 47 | Ahl\_daily | Ah load - daily | Ah | n·0.1 |
| 48 | array\_fault\_daily | Array fault bit field - daily - | - |  |
| 49 | load\_fault\_daily | Load fault bit field - daily - | - |  |
| 50 | alarm\_HI\_daily | alarm bit field – daily, HI word - | - |  |
| 51 | alarm\_LO\_daily | alarm bit field – daily, LO word - | - |  |
| 52 | vb\_min | minimum battery voltage | V | n·100·2-15 |
| 53 | vb\_max | maximum battery voltage | V | n·100·2-15 |
| - | Checksum | \*hh | - | 00 - FF |
| - | EOM | <CR><LF> (End of message termination) | - |  |

Modeled after NEMA 0183 specification. A simple ASCII serial communication protocol that define how data are transmitted in a “sentence” from the ‘talker’ to the ‘listener’.

**Protocol rules:**

* Each message's starting character is a dollar sign.
* The next eight characters identify the talker (5 characters) and the type of message (3 characters).
* All data fields that follow are comma-delimited.
* Where data is unavailable, the corresponding field remains blank (it contains no character before the next delimiter.
* The first character that immediately follows the last data field character is an asterisk, but it is only included if a checksum is supplied.
* The asterisk is immediately followed by a checksum represented as a two-digit hexadecimal number. The checksum is the bitwise exclusive OR of ASCII codes of all characters between the *$* and *\**. According to the official specification, the checksum is optional for most data sentences, but is compulsory for RMA, RMB, and RMC (among others).
* <CR><LF> ends the message.

Each item adheres to Modbus rules; that is, they are 16-bit raw values that need to be scaled as indicated in the above table.

|  |  |  |  |
| --- | --- | --- | --- |
| **SunSaver MPPT Logged Data Layout** | | | |
| **Item name** | **Item description** | **Units** | **Scaling / Range /**  **Example Data** |
| Message ID | $PMPPTLOG | - | - |
| Hourmeter | 24 bit value ‘hour meter’ timestamp | h | 0 to (224-1) |
| Alarm\_daily | 24 bit Alarm bit field – daily | - | - |
| Vb\_min\_daily | Vb minimum voltage – daily | V | n·100·2-15 |
| Vb\_max\_daily | Vb maximum voltage - daily | V | n·100·2-15 |
| Ahc\_daily | Ah charge - daily | Ah | n·0.1 |
| Ahl\_daily | Ah load - daily | Ah | n·0.1 |
| Array\_fault\_daily | Array fault bit field – daily | - | - |
| Load\_fault\_daily | Load fault bit field – daily | - | - |
| Va\_max\_daily | Va maximum voltage – daily | V | n·100·2-15 |
| Time\_ab\_daily | Time in absorption - daily | min | - |
| Time\_eq\_daily | Time in equalize - daily | min | - |
| Time\_fl\_daily | Time in float – daily | min | - |
| - | Reserved | - | - |
| - | Reserved | - | - |
| - | Reserved | - | - |
| Checksum | \*hh | - | 00 - FF |
| EOM | <CR><LF> (End of message termination) | - |  |

**Logged Data**

The SS-MPPT stores approximately 32 days of data. This data is stored in a circular buffer where the oldest data is over-written by the newest data. The log data must be requested and sorted into correct order before the data will be useful.

* The logged data is mapped from 0x8000-0x81FF (1kB)
* The data consists of up to 32 blocks of data. (One block is generally one day w/ some exceptions)
* Each block is 32bytes (16 modbus variables)
* It is written in a circular buffer format. All blocks must be read and then put in linear order via the hourmeter field. Ignore blocks w/ hourmeters of 0x000000 or 0xFFFFFF. The largest hourmeter block is the most recent.

Data is stored in big endian format.

struct { Uint24 hourmeter;

Uint24 alarm\_daily;

Uint16 Vb\_min\_daily;

Uint16 Vb\_max\_daily;

Uint16 Ahc\_daily;

Uint16 Ahl\_daily;

Uint16 array\_fault\_daily;

Uint16 load\_fault\_daily;

Uint16 Va\_max\_daily;

Uint16 time\_ab\_daily; v0.7 code and later

Uint16 time\_eq\_daily; v0.7 code and later

Uint16 time\_fl\_daily; v0.7 code and later

Uint16 reserved[3];

}

**Data Retrieval Commands**

In order to make the retrieval of data controllable there are three methods for the translator to send messages.

1. **‘t**’ - Periodic method. Message string will be sent at a timed interval (every 5 seconds for instance).
2. **‘r**’ - Command based method. Send messages upon command from the receiver. Periodic method disabled.
3. **‘l’** - Command based Logged Data method. Shuts down status messages and sends one log entry per command. The status messages are disable until either of the above commands are sent.

The logged data is retrieved using the **‘l’** (logged) command. Each command cause one entry of the MPPT log circular buffer to be read and sent out. Each entry has a ‘hourmeter’ time stamp. After all the entries have been retrieved (32) they need to be sorted by the ‘hourmeter’ field. This is also the way you can tell if all the entries have been received. The status messages are disabled so as not to mix up the receiver on what data is being received.

After getting the logged data, the status messages can be turn back on by either sending the **‘t’** command causing the messages to be sent on a time interval or the **‘r’** command that will get one status message only.