SunSpec Alliance Interoperability Specification

Common Elements

SunSpec Alliance Common Elements Workgroup

John Blair, John Nunneley, Karl Lambert, Pat Adamosky, Ronnie Petterson, Lynn Linse

Version 1.4

ABSTRACT

This document describes the common elements of the SunSpec Alliance interoperability specification

Change History

1.3: Added change history

1.3: Allow device aggregation via multiple common blocks

1.3 Manufacturer ID should be registered with SunSpec

1.3 Added M/O/C column to xls map.

1.3 Require Manufacturer, Model, and Serial Number to be supplied

1.4 Add Copyright

Introduction

The SunSpec Alliance Interoperability Specifications describe the data models and MODBUS register mappings for devices used in Renewable Energy systems.

The data models can be easily moved by XML and other technology, so these specifications are not tied to the 30-year old MODBUS protocol; MODBUS is the initial implementation specified because MODBUS is well understood and supported within renewable energy systems.

Each type of device is defined in a separate document that details the fields and values specific to that type. Current device models defined are

- Inverter models
- Meter models
- Environmental models (including Meteorological, Irradiance, Temperature, Inclinometer, GPS/Location, and Reference Point)
- String Combiners
- Panel / Module DC-DC compensators

Additional device models will be taken up and defined as directed by the SunSpec Alliance membership.

This document focuses on the elements of the specification that are common to all device types.

- Common data model
- Standard data formats
- MODBUS register mappings
- Device Specification Format
- Extensibility
- Device Aggregation
- Optionality

Common Device Model

All SunSpec devices support a consecutive collection of well defined blocks, each prefaced with a well-known Device Identity and length field. This allows a remote client to browse the contents skipping blocks with unrecognized Device Identify (DID) values. The Common Model specifically allows for extensibility in the following ways:

1. Definition of new Device Specific Blocks. New devices are defined and assigned a new device type id.
2. Revisions of Device Specific Blocks. Revisions to devices are defined and assigned a new device type id.
3. Vendor specific extensions. Models may include fields that have vendor specific values. An example of this is the status and event code fields. Vendors may also define a Vendor Extension Block that includes fields and values specific to the vendor. These are assigned a device type id. This block may be then concatenated to the associated Device Specific Block.
4. Composite device types. Device Specific Blocks may be concatenated together to form a composite device. An example of this is a collection of environmental devices.
5. Device aggregations. A map may contain multiple Common Blocks. Each Common Block marks the start of a new device. This is useful when a map represents an aggregation of devices. (See the Module/Panel specification for an example of this usage).

All SunSpec devices must include at least three blocks:

- The first block is the Common Block, which supplies vendor and model information for the device.
- The second (and subsequent) blocks will be device specific – for example a meter block, or a meter block followed by a GPS Location block.
- The final End Block formally marks the end of the Device Specific Blocks.

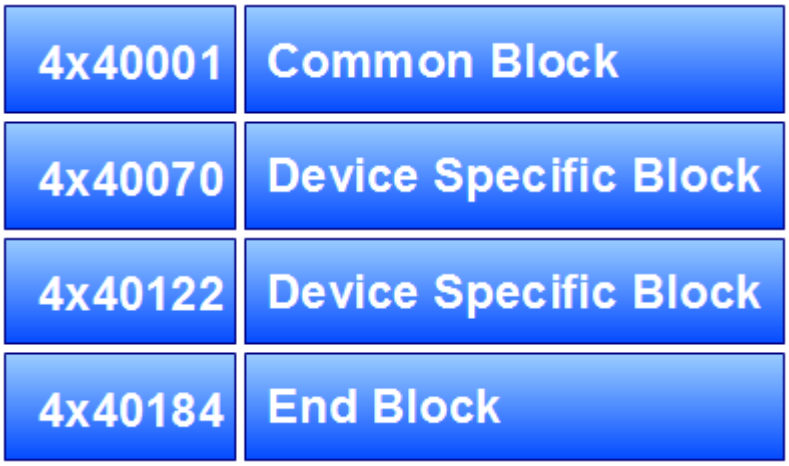

Figure 1 - Example SunSpec Device with two device-specific blocks

Data Element Optionality

Data element support may be designated as Mandatory, Optional, or Conditional. Mandatory means the attribute is required to be supported. Optional means the element is not required. Implementations shall return the "Not Supported" value if the element is not used in this case. Conditional means that the element is required if some other option is implemented. E.g. specification of a scale factor.

Common Block

The following data elements shall be provided to uniquely identify a SunSpec device.

- **C_SunSpec_ID** – A well-known value that confirms to the client that this device has implemented a collection of SunSpec Alliance Device blocks.
- **C_SunSpec_DID –** A well-known value that uniquely identifies the specific type of SunSpec device model this block represents. The DIDs are assigned by SunSpec.
- **C_SunSpec_Length –** A value that is the length of the device model block in Modbus 16-bit registers. This value does NOT include the C_SunSpec_DID or C_SunSpec_Length. I.e. it is the number of registers that follow the length.
- **C_Manufacturer** – A unique value that identifies the Manufacturer of this device. Manufacturers are encouraged to register their ID with SunSpec to guarantee uniqueness. SunSpec maintains a database of certified products by Manufacturer.
- **C_Model** – A manufacturer specific value that identifies the model of this device. SunSpec maintains a database of certified products by Model.
- **C_Options** - A manufacturer specific value that identifies any model options for this device.

- **C_Version** - A manufacturer specific value that identifies the firmware version of this device.
- **C_SerialNumber** - A manufacturer specific value that uniquely identifies this device within the manufacturer name space.
- **C_Device Address** - Protocol specific value to address this device instance. For MODBUS devices, this is the MODBUS device ID.

Note: SunSpec requires that the result of concatenating the three strings **C_Manufacturer, C_Model,** and **C_SerialNumber** returns a globally unique string for all of a manufacture's products. If the optional **C_Model** value is not implemented, then concatenating the two strings **C_Manufacturer** and **C_SerialNumber** must be a globally unique string. This string may be used (for example) by logging and uploading functions.

Device Specific Block

The SunSpec Common Model is then followed by one or more device specific blocks of data values. The Device Specific Block begins like the Common Model Block.

- **C_SunSpec_DID –** A well-known value that uniquely identifies the specific type of SunSpec device model this is.
- **C_SunSpec_Length –** A value that is the length of the device model block in registers.
- Other data fields would follow as required. Since the register length was included in C_SunSpec_Length., this allows clients to skip blocks of an unknown type and format.

End of SunSpec Device Marker

The overall SunSpec Device Block is punctuated by a termination C_SunSpec_DID ((0xFFFF)) and a C_SunSpec_Length value of zero.

Standard Data Formats

The MODUBS specification is not explicit on how to encode numbers other than 16 bit integers. Differences do exist between one manufacturer's implementation and another's. Not all implementations of a specific device will support all of the values defined for that device. Unsupported values are indicated by supplying the "NOT_IMPLEMENTED" type specific value in response. This specification restricts values to the following standard data formats.

16 bit Integer Values

Values are stored in big-endian order per the MODBUS specification and consist of a single register. All integer values are documented as signed or unsigned. All signed values are represented using two's-compliment.

Modbus Register	1	
Byte	0	1

Bits	15	14	13	12	10	11	9	8	7	6	5	4	3	2	1	0

int16 Range: -32767 ... 32767 Not Implemented: 0x8000

uint16 Range: 0 ... 65534 Not Implemented: 0xFFFF

acc16 Range: 0 ... 65535 Not Accumulating: 0x0000

NOTE: it is up to the master to detect rollover of accumulated values.

32 bit Integer Values

32 bit integers are stored using two registers in big-endian order

Modbus Register	1		2	
Byte	0	1	2	3
Bits	31 ... 24	23 ... 16	15 ... 8	7 ... 0

int32 Range: -2147483647 ... 2147483647 Not Implemented: 0x80000000

uint32 Range: 0 ... 4294967294 Not Implemented: 0xFFFFFFFF

acc32 Range: 0 ... 4294967295 Not Accumulating: 0x00000000

NOTE: it is up to the master to detect rollover of accumulated values.

64 bit Integer Values

64 bit integers are stored using for registers in big-endian order.

Modbus Register	1		2	
Byte	0	1	2	3
Bits	63 ... 56	55 ... 48	47 ... 40	39 ... 32

Modbus Register	3		4	
Byte	4	5	6	7
Bits	31 ... 24	23 ... 16	15 ... 8	7 ... 0

int64 Range: -9223372036854775807 ... 9223372036854775807 Not Implemented: 0x8000000000000000

uint64 Range: 0 … 18446744073709551614 Not Implemented: 0xFFFFFFFFFFFFFFFF

acc64 Range: 0 ... 18446744073709551615 Not Accumulating: 0x0000000000000000

NOTE: it is up to the master to detect rollover of accumulated values.

String Values

Store variable length string values in a fixed size register range using a NULL (0 value) to terminate or pad the string. For example, up to 16 characters can be stored in 8 contiguous registers as follows

Modbus Register	1		2		3		4		5		6		7		8	
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Character	E	X	A	M	P	L	E	spc	S	T	R	I	N	G	!	NULL

NOT_IMPLEMENTED value: all registers filled with NULL or 0x0000

Floating Point Values

Floating point values are 32 bits and encoded according to the IEEE 754 floating point standard.

Modbus Register	1															
Byte	0								1							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEEE 754	sign	Exponent							Fraction							

Modbus Register	2															
Byte	2								3							
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEEE 754	Fraction least															

float32 Range: see IEEE 754 Not Implemented: 0x7FC00000 (NaN)

Scale Factors

As an alternative to floating point format, values are represented by integer values with a signed scale factor applied. The scale factor explicitly shifts the decimal point to the left (negative value) or the right (positive value). Scale factors may be fixed and specified in the documentation of a value, or may have a variable scale factor associated with it. E.g. A value "Value" may have an associated value "Value_SF" that is a 16 bit two's compliment integer.

sf signed range: -10 … 10 Not Implemented: 0x8000

Defined Units

No units are defined as part of the Common Model. Units are defined as needed by specific device models. Where units are shared across device models, care will be taken to ensure a common definition of those units.

Assigned Values

Values that are well-known and part of the Common model are defined here.

C_SunSpec_ID : 0x53756e53 ("SunS")

C_SunSpec_DID : Assigned Device IDs

The following device model identifiers have been assigned

0XX Common Models

001 - Common Block

002 – Aggregator Block

1XX Inverter Models

101 - Single Phase Inverter Model (Integer+SF)

102 - Split Phase Inverter Model (Integer+SF)

103 - 3 Phase Inverter Model (Integer+SF)

111 - Single Phase Inverter Model (Floats)

112 - Split Phase Inverter Model (Floats)

113 - 3 Phase Inverter Model (Floats)

2XX Meter Models

201 - Single Phase Meter Model

202 - Split Phase Meter Model

203 - Wye-Connect Meter Model

204 - Delta-Connect Meter Model

3XX Environmental Models

301 - Base Meteorological Model (DEPRECATED)

302 - Irradiance Model

303 – Back of Module Temperature Model

304 - Inclinometer Model

305 - Location Model

306 - Reference Point Model

307 – Base Meteorological Model (Corrected)

308 – Mini Meteorological Model

4XXX String Combiner Models

401 – Basic String Combiner

402 – Advanced String Combiner

5XXX Module (Panel) Models

501 – Panel Model (Float)

502 – Panel Model (Integer)

64XXX Vendor Extension Blocks

(None Defined at this time)

65XXX SunSpec Reserved

65535 - End of SunSpec Map (0xFFFF)

C_Status: Assigned Status Codes

Devices that support a status code value in the Device Specific Block shall support the following status values.

C_STATUS_NORMAL	0x00000000	:	Operating Normally
C_STATUS_ERROR	0xFFFFFFFE	:	Generic Failure
C_STATUS_UNK	0xFFFFFFFF	:	Status Unknown

Device specific status codes shall be defined in corresponding device model.

MODBUS Register Mappings

All SunSpec device register maps will have a well-known base address with an alternate base address that may be used if the base address is unavailable for a particular manufacturer's device model. The first register in the map shall be the well-known 32 bit identifier C_SunSpec_ID. This allows for discovery of SunSpec compatible devices. If the base register does not return this value, the alternate base register shall be checked. If this test fails, the device is not SunSpec compatible. The next value indicates the SunSpec device model type. The first device model block shall always be the SunSpec Common Model. The third value is the length of

the device model block, followed by the device model values. Additional device model types are concatenated with their corresponding type, length, and values.

SunSpec Base and Alternate Base Register Addresses

Base Register: 40001 Alternate Base Register: 50001 (decimal value)

40001 and 50001 are actual register offsetsthat start at 1 – not a function code and not to be confused with the Modicon convention, which would represent these as 4x40001 and 4x50001.

To read register 40001, use the hexadecimal offset of 0x9C40 (40,000) on the wire.

If you read beyond the end of the block, an exception may or may not be thrown according to the implementation. If no exception is thrown, then data that comes after the End Block is invalid and should not be used. It is recommended that masters read the common block to determine the contents of the map.

Fixed device block sizes must be the size as defined. It is nonconformant to truncate a fixed size device block.

REMINDER: This specification only addresses the format of the data. The data can be moved via Modbus/TCP or RTU – or any other media which can move Modbus data. Implementations are responsible for returning measured values as designed. For example, power and voltage readings may not be in sync depending on the product design.

Common MODBUS Model Mapping

MODUBS maps are defined using this common table format.

Start	End	#	R/W	Name	Type	Units	ScaleFactor	Contents	Description
0001	0002	2	R	C_SunSpec_ID	uint32	N/A	N/A	0x53756e53 (SunS)	Well-known value. Uniquely identifies this as a SunSpec Modbus Map
0003	0003	1	R	C_SunSpec_DID	uint16	N/A	N/A	0x0001	Well-known value. Uniquely identifies this as a SunSpec Common Model block
0004	0004	1	R	C_SunSpec_Length	uint16	registers	N/A	65	Length of common model block
0005	0020	16	R	C_Manufacturer	String(32)	N/A	N/A	N/A	Well-known value
0021	0036	16	R	C_Model	String(32)	N/A	N/A	N/A	Manuf specific value
0037	0044	8	R	C_Options	String(16)	N/A	N/A	N/A	Manuf specific value
0045	0052	8	R	C_Version	String(16)	N/A)	N/A	N/A	Manuf specific value
0053	0068	16	R	C_SerialNumber	String(32)	N/A	N/A	N/A	Manuf specific value
0069	0069	1	R/W	C_DeviceAddress	uint16	N/A	N/A	N/A	Modbus Id
0070	**0070**	**1**	**R**	**C_SunSpec_DID**	**uint16**	**N/A**	**N/A**	**Device Id**	**Start of next Device**
0071	0071	1	R	C_SunSpec_Length	unit16	N/A	N/A	Device Length	Device Model Block Size
0072	…	Length Registers of Device Model Specific Data							
…	…	…							
Next	Block	Another DID/Length Block/Data or EndOfSunSpecBlock DID value						0xFFFF	End of Map
Last	Last	EndofSunSpecBlock length						0x0000	End of Map