Document #: A12031-1.6

Status: Approved

Version 1.6

Approval date: 06-01-2013

Common Models

SunSpec Alliance Specifications

Contributors:

John Blair, John Nunneley, Karl Lambert, Pat Adamosky, Ronnie Petterson, Lynn Linse, Bill Randle, Bob Fox, Aaron Parker, Tom Tansy

Abstract

This document describes the common elements of the SunSpec information modeling system and is an approved SunSpec Alliance interoperability specification.

License Agreement and Copyright Notice

Prepared by the SunSpec Alliance

4030 Moorpark Avenue, Suite 109

San Jose, CA 95117

Website: www.sunspec.org

Email: info@sunspec.org

Revision History

Revision	Date	Reason
1.3	06-01-2013	• Added change history • Allow device aggregation via multiple common blocks • Manufacturer ID should be registered with SunSpec • Added M/O/C column to xls map • Require Manufacturer, Model, and Serial Number to be supplied
1.4		Add Copyright
1.5		• Updated Logo • Corrected Assigned ID range for String Combiner and Module • Added Assigned ID ranges for Inverter model, version 1.2, Module model, version 1.1, Inverter Controls and Network Interface • Added concept of a scale factor as a basic SunSpec type • Added concept of SunSpec Device type to Common Block • Changed over to new names for points • Added definition for acc64, ipaddr, ipv6addr types • Scale factor clarification wrt unimplemented • Align the document with the new short names
1.6	09-24-2014	Change to new SunSpec document format

About the SunSpec Alliance

The SunSpec Alliance is a trade alliance of developers, manufacturers, operators and service providers, together pursuing open information standards for the distributed energy industry. SunSpec standards address most operational aspects of PV, storage and other distributed energy power plants on the smart grid—including residential, commercial, and utility-scale systems—thus reducing cost, promoting innovation, and accelerating industry growth.

Over 70 organizations are members of the SunSpec Alliance, including global leaders from Asia, Europe, and North America. Membership is open to corporations, non-profits, and individuals. For more information about the SunSpec Alliance, or to download SunSpec specifications at no charge, please visit www.sunspec.org.

About the SunSpec Specification Process

SunSpec Alliance specifications are initiated by SunSpec members desiring to establish an industry standard for mutual benefit. Any SunSpec member can propose a technical work item. Given sufficient interest and time to participate, and barring any significant objections, a workgroup is formed and its charter is approved by the board of directors. The workgroup meets regularly to advance the agenda of the team.

The output of the workgroup is generally in the form of an Interoperability Specification. These documents are considered to be normative, meaning that there is a matter of conformance required to support interoperability. The revision and associated process of managing these documents is tightly controlled. Other documents are informative, or make some recommendation with regard to best practices, but are not a matter of conformance. Informative documents can be revised more freely and frequently to improve the quality and quantity of information provided.

SunSpec Interoperability Specifications follow this lifecycle pattern of DRAFT, TEST, APPROVED and SUPERSEDED.

For more information or to download a SunSpec Alliance specification, go to http://www.sunspec.org/specifications.

Table of Contents

Table of Figures

Nomenclature

Abbreviation	Meaning
DA	Protocol specific value to address device instance
ID	Device Identity
L	Length of device model block
Md	Manufacturer-specific value identifying device model
Mn	Value identifying device manufacturer
Opt	Manufacturer specific value identifying any model options for the device
Pad	Sunspec pad16 register
RTU	Remote Telemetry Unit
SN	Manufacturer-specific value identifying device within the manufacturer name space
TCP	Transmission Control Protocol
Vr	Manufacturer-specific value identifying firmware version of the device
XML	Extensible Markup Language

1. INTRODUCTION

The SunSpec Alliance Interoperability Specifications describe the information models and Modbus register mappings for devices used in Renewable Energy systems. A physical device, such as an inverter, is represented by a collection of one or more of these logical models. This document describes the Common Models and model design conventions. All SunSpec devices must contain the device information common model that describes this particular device as the first logical block of the device.

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.

2. SUNSPEC DEVICE TYPES

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

1. Inverter
2. Meter
3. Weather Station
4. String Combiner
5. PV Module
6. Tracker
7. Charge Controller
8. Storage

99. Other

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

3. SUNSPEC COMMON MODELS AND CONVENTIONS

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 model
- Network configuration models
- Optionality

4. 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 Identity (ID) 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 model id.
2. Revisions of Device Specific Blocks. Revisions to device models are defined and assigned a new device model id. Simple models may be appended to without affecting backwards compatibility and do not require a new model id. Readers shall rely on the length of the model to determine this.
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 Weather Station represented by a collection of environmental models.
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.

4x40001	Common Block
4x40070	Device Specific Block
4x40122	Device Specific Block
4x40184	End Block

Figure 1: Example SunSPec Device with no two-device specific blocks

Data Element Optionality

Data element support may be designated as Mandatory, or Optional. 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.

4.1 Common Block

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

- **SunSpec_ID** – A well-known value that confirms to the client that this device has implemented a collection of SunSpec Alliance Device blocks.
- **ID –** A well-known value that uniquely identifies the specific type of SunSpec device model this block represents. The IDs are assigned by SunSpec. For the Common Block, this value is "1".
- **L –** A value that is the length of the device model block in Modbus 16-bit registers. This value does NOT include the ID or L. I.e. it is the number of registers that follow the length. For the Common Block, this value is 66 but may be 65. All SunSpec device blocks should be of even length. Common Blocks of length 66 shall contain the SunSpec pad16 value as the final register.
- **Mn** – 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. The Mn should be concise and constrained to simple alphanumeric text void of spacing.
- **Md** – A manufacturer specific value that identifies the model of this device. SunSpec maintains a database of certified products by Model. The Md should be concise and constrained to simple alphanumeric text void of spacing.
- **Opt** - A manufacturer specific value that identifies any model options for this device.
- **Vr** - A manufacturer specific value that identifies the firmware version of this device.
- **SN** - A manufacturer specific value that uniquely identifies this device within the manufacturer name space.
- **DA** - Protocol specific value to address this device instance. For Modbus devices, this is the Modbus device ID.
- **Pad** – SunSpec pad16 register. Must be included for Common Blocks of length 66. Not included if length is 65.

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

4.2 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.

- **ID –** A well-known value that uniquely identifies the specific type of SunSpec device model this is.
- **L –** A value that is the length of the device model block in registers. The length should be even and may contain a rounding pad register. The length does not include the ID or L register. It is the number of registers that follow.
- Model data fields would follow as required. If the ID is not, the L value is used to skip over the model data and move to the next block.

4.3 End of SunSpec Device Marker

The overall SunSpec Device Block is punctuated by a termination ID ((0xFFFF)) followed by a length value of zero.

5. STANDARD DATA FORMATS

The Modbus 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.

5.1 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

Figure 2: 16 bit integer values

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

uint16 Range: 0 … 65534 Not Implemented: 0xFFFF

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

pad16 Range: 0x8000 Always returns 0x8000

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

5.2 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

Figure 3: 32 bit integer values

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

uint32 Range: 0 … 4294967294 Not Implemented: 0xFFFFFFFF

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

ipaddr 32 bit IPv4 address Not Configured: 0x00000000

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

5.3 64 bit Integer Values

64 bit integers are stored using four 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

Figure 4: 64 bit integer values

int64 Range: -9223372036854775807 … 9223372036854775807

Not Implemented: 0x8000000000000000

acc64 Range: 0 ... 9223372036854775807 Not Accumulated: 0

NOTE: Only positive values in the int64 range are allowed. Values outside of this range shall be considered invalid.

NOTE: This value shall rollover after the highest positive value in the int64 range (0x7fffffffffffffff). It is up to the reader to detect rollover of accumulated values.

5.4 128 Bit Integer Values

128 bit integers are stored using eight registers in big-endian order.

ipv6addr 128 bit IPv6 address Not Configured: 0

5.5 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

Figure 5: Example string values data format

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															

Figure 6: Floating point values

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

5.6 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" of type "sunssf" that is a 16 bit two's compliment integer.

sunssf signed range: -10 … 10 Not Implemented: 0x8000 or 0x0000

NOTE: Any value outside of the sunssf range shall be considered to be "Not Implemented".

6. 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.

7. ASSIGNED VALUES

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

Note: Current / latest list of assigned values can be found at http://sunspec.org/AssignedIDs

SunSpec ID: 0x53756e53 ("SunS")

7.1 SunSpec ID: Assigned Model IDs

The following device model identifiers have been assigned

0XX Common Models

001 - Common Block

002 – Aggregator Block

01X Configuration Models

010 – Interface Configuration Model

011 – Ethernet Link Configuration

012 – IPv4 Protocol Configuration

013 – IPv6 Protocol Configuration

014 – Proxy Server Configuration

015 – Interface Packet Counters

016 – Simplified Network Configuration

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)

120 – Inverter Controls Nameplate Ratings

121 – Inverter Controls Basic Settings

122 – Inverter Controls Measurements and Status

123 – Inverter Controls Immediate Controls

124 – Inverter Controls Basic Storage Control

125 – Inverter Controls Pricing Signal

126 – Inverter Controls Static Volt-VAR Arrays

127 – Inverter Controls Frequency Watt Control

128 – Inverter Controls Dynamic Reactive Current

129 – Inverter Controls LVRT Arrays

130 – Inverter Controls HVRT Arrays

131 – Inverter Controls Watt-Power Factor Arrays

132 – Inverter Controls Voltage-Watt Arrays

133 – Inverter Controls Basic Scheduling

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

4XX String Combiner Models

401 – Basic String Combiner (superseded)

402 – Advanced String Combiner (superseded)

403 – Basic String Combiner 1.1

404 – Advanced String Combiner 1.1

5XX PV Module (Panel) Models

501 – Panel Model (Float)

502 – Panel Model (Integer)

64XXX Vendor Extension Blocks

See http://SunSpec.org/AssignedIDs for the latest

65XXX SunSpec Reserved

65535 - End of SunSpec Map (0xFFFF)

7.2 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.

8. 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 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 (decimal values)

Preferred Base Register: 40001

Alternate Base Register: 50001

Alternate Base Register: 00001

Base registers are actual register offsets that 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 nonconforming 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 that 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.

9. COMMON MODBUS MODEL MAPPING

Modbus maps are defined using this common table format.

Start	End	#	R/W	Name	Type	Units	Scale Factor	Contents	Description	M/O
1	2	2	R	SunSpec_ID	uint32	N/A	N/A	0x53756e53 (SunS)	Well-known value. Uniquely identifies this as a SunSpec Modbus Map	M
3	3	1	R	ID	uint16	N/A	N/A	0x0001	Well-known value. Uniquely identifies this as a SunSpec Common Model block	M
4	4	1	R	L	uint16	registers	N/A	65,66	Length of common model block	M
5	20	16	R	Mn	String(32)	N/A	N/A	N/A	Well-known value	M
21	36	16	R	Md	String(32)	N/A	N/A	N/A	Manuf specific value	O
37	44	8	R	Opt	String(16)	N/A	N/A	N/A	Manuf specific value	O
45	52	8	R	Vr	String(16)	N/A)	N/A	N/A	Manuf specific value	O
53	68	16	R	SN	String(32)	N/A	N/A	N/A	Manuf specific value	M
69	69	1	R/W	DA	uint16	N/A	N/A	N/A	Device Address	O
70	**70**	**1**	**R**	**Pad**	**pad16**	**N/A**	**N/A**	**N/A**	**Pad register to make the model even**	**O**
71	**71**	**1**	**R**	**ID**	**uint16**	**N/A**	**N/A**	**Device Id**	**Start of next Device**	**M**
72	72	1	R	L	unit16	N/A	N/A	Device Length	Device Model Block Size	M
73	…	Length Registers of Device Model Specific Data								
…	…	…								

Start	End	#	R/W	Name	Type	Units	Scale Factor	Contents	Description	M/O
Next	Block	Another DID/Length Block/Data or EndOfSunSpecBlock DID value						0xFFFF	End of Map	M
Last	Regi ster	EndofSunSpecBlock length						0x0000	End of Map	M

Figure 7: Common Modbus Model Mapping