

# Modbus Implementation in APC Smart-UPS

By Daniel Cohen & Manuel J. Palomino

## PROJECT AT A GLANCE

### Project Type

Control, monitor and configure one or multiple Smart-UPS devices through a custom application making use of standard Modbus communications to collect and deliver data from the UPS management system.

### Products Supported

APC Smart-UPS in the SMT family (excluding rack mount 1U) running firmware greater than UPS 09.0. SMT models with UPS 08.0 or higher may be updated. See Knowledge Base article [FA16473](#) on [APC.com](#) for information on firmware updates.

### Additional Equipment Required

No additional hardware is required for RS-232 Modbus communications. If RS-485 is required an adaptor may be used. Review the Knowledge Base article [FA156062](#) in [APC.com](#) for recommendations.

### Recommended References

Modbus Application Protocol Specification V1.1b3 – Modbus Organization, 2012

Modbus over serial line specification and implementation guide V1.02 – Modbus Organization, 2006



## Modbus implementation in SMART-UPS

Modbus is a serial communications protocol, managed and made publicly available by the Modbus Organization ([www.modbus.org](#)). While it was developed for, and is heavily used in, industrial control environments, it is a well defined protocol suitable for machine to machine communications. A wide variety of readily available software and hardware devices exist today that support communications via Modbus.

Schneider Electric's APC Smart-UPS implement the Modbus protocol to provide access to controls and configuration parameters through user-defined applications. The data available via Modbus is the same data accessible through APC's software products.

This document outlines Modbus communications over the range of transport layers, as well as where the implementation deviates from the MODBUS specifications. The standard definition of the Modbus protocol can be obtained, royalty-free, from [www.modbus.org](#).

**Table of Contents**

<b>1</b>	<b>Smart-UPS Data Concepts</b> .....	<b>4</b>
1.1	Definitions .....	4
1.1.1	Usage.....	4
1.2	Data Types implemented within Modbus Map .....	4
1.2.1	Bitfield Data Type .....	4
1.2.2	Enumeration Data Type.....	4
1.2.3	Binary-Point-Integer Data Type .....	4
1.2.4	String Data Type.....	5
1.3	Mapping of Data Types into Modbus Registers .....	5
1.3.1	Numbers .....	5
1.3.2	Bitfields .....	6
1.3.3	Strings.....	6
<b>2</b>	<b>Transport Layers of Communication</b> .....	<b>6</b>
2.1	SERIAL CONNECTION .....	6
2.2	USB CONNECTION.....	6
<b>3</b>	<b>Modbus Implementation</b> .....	<b>6</b>
3.1	Register Blocks.....	7
3.1.1	Register Block Reads .....	7
3.1.2	Register Block Writes .....	7
3.2	Undefined Register Addresses.....	7
3.2.1	Undefined Register .....	7
3.2.2	Not Applicable Register .....	8
3.3	Writing a Modbus Register .....	8
<b>4</b>	<b>Modbus RTU</b> .....	<b>8</b>
4.1	MODBUS DEVICE ADDRESS .....	8
4.2	UNSUPPORTED FEATURES.....	8
4.2.1	ASCII Transmission Mode .....	8
4.2.2	Modbus Message RTU Framing.....	8
4.3	Modbus RTU- RS232 Serial .....	9
4.4	Modbus RTU- USB .....	9
4.4.1	Overview .....	9
4.4.2	USB Host Drivers.....	9
4.4.3	USB HID Usages .....	9
4.4.4	Framing.....	10
<b>5</b>	<b>Modbus Register Organization</b> .....	<b>11</b>
5.1	Status Data (0x0000 – 0x0007F).....	11
5.2	Dynamic Data (0x0080 – 0x01FF).....	11
5.3	Inventory Information (0x0200 – 0x03FF) .....	11
5.4	Static Data (0x0400 – 0x5FF).....	11
5.5	Commands (0x0600 – 0x6FF).....	11
5.6	Protocol Verification (0x0800 – 0x8FF) .....	11
<b>6</b>	<b>Enabling Modbus on Smart-UPS</b> .....	<b>11</b>
6.1	Model compatibility .....	11
6.2	Firmware Update .....	11
6.3	Enabling Modbus Communications .....	11
	<b>Appendix A – Register List</b> .....	<b>12</b>
	<b>Appendix B – Register Bit Descriptions</b> .....	<b>20</b>
	UPSStatus_BF .....	20

SimpleSignalingStatus_BF .....	20
GeneralError_BF .....	21
PowerSystemError_BF .....	21
BatterySystemError_BF .....	21
ReplaceBatteryTestStatus_BF .....	22
RunTimeCalibrationStatus_BF .....	22
LifeTimeStatus_BF .....	23
UserInterfaceStatus_BF .....	23
Input.InputStatus_BF .....	24
VoltageACSetting_BF .....	24
AcceptableFrequencySetting_BF .....	24
UPSCommand_BF .....	25
SimpleSignalingCommand_BF .....	25
Register: UPSStatus_BF .....	25
RunTimeCalibrationCommand_BF .....	25
UserInterfaceCommand_BF .....	25
OutletStatus_BF .....	26
OutletCommand_BF .....	27
I_SOGRelayConfigSetting_BF .....	28
<b>Appendix C – Enumeration Descriptions .....</b>	<b>29</b>

## 1 Smart-UPS Data Concepts

APC products have an internal communication protocol that has its own defined data set.

### 1.1 Definitions

#### 1.1.1 Usage

A usage is a data point that is defined by the APC internal protocol. Each Modbus register referenced in the Modbus memory map is either an entire usage, or part of a defined as a usage.

### 1.2 Data Types implemented within Modbus Map

The Modbus register map in APC products uses the same data types implemented by the product internally for its normal operation. Each usage is one of the following data types.

All data is always conveyed in MSB first ([big-endian](#)) method unless otherwise specified.

#### 1.2.1 Bitfield Data Type

A *bitfield* data type is one that is designed to store binary states of related or unrelated data items. It is used to compress the data representation to multiples of bytes. Bitfields may have bits that are defined as being mutually exclusive so that only one bit can be set at a time. Bitfields are defined in units of 8 bit bytes up to 8 bytes so that states can be written in one bitfield, valid size for bitfields are 8, 16, 32 and 64 bits.

An example bitfield could be the presence of modules on a device. For a device that has a maximum of 6 modules, the bitmask might be 0x3F. If the value reported by the Modbus slave is 0x10 then the 4<sup>th</sup> module is present, if the value reported is 0x11 then the 0<sup>th</sup> and 4<sup>th</sup> module are present. A value of 0x80 would be invalid since it is outside of the bit mask.

#### 1.2.2 Enumeration Data Type

This data type is an integer that represents a value out of a set of possibilities for a particular property of a device.

An example of an enumeration could be the power quality setting as good, fair, and poor respectively. A value of 1 would mean that the power quality is good. Here the minimum value reported would be 1 and the maximum value is 3.

Enumerations are always signed, and have a maximum size of 8 bytes meaning that the set can only contain up to 4294967295 entries.

#### 1.2.3 Binary-Point-Integer Data Type

All numeric values are represented as binary point integers (BPI). It is used for integer and floating point data. It is intended for readings and settings. It can be signed or unsigned depending on the usage definition. The usage definition will contain the Binary Point Location (BPL) (scaling factor), signed / unsigned info, size of the data, and units.

The maximum size of a BPI is 8 bytes.

##### 1.2.3.1 Converting to and from Floating Point Number

The following sections show how to convert a number from floating point to a BPI and back. For the example we will use the value of -56.8, and convert it to a BPI of size 2 with a BPL of 5.

##### 1.2.3.2 Converting from Floating Point Number to Binary-Point-Integer

- Multiply the number by  $2^{\text{bpl}}$  by the absolute value
  - $N * 2^{\text{bpl}}$   $56.8 * 2^5 = 1816.96$
- Add a  $\frac{1}{2}$  to round the number. Remember this is a lossy some precision will be lost.
  - $1817.6 + 0.5 = 1818.1$
- Discard the decimal remainder you are now left with an integer value
  - 1818
- Do the two's complement ( Only if it is a negative number we are converting)

- 1818 becomes -> 63718 or in hex F8E6  
<- this is the BPI with BPL 5 (float = -56.8)

### 1.2.3.3 Converting from Binary-Point-Integer to Floating Point Number

We will take the number F8E6 and covert it from BPI size 2, BPL 5 back to a floating point number

- Do the twos complement (only if the data type is defined as signed and the most significant bit is 1
  - F8E6 becomes -> 71A which is 1818 in decimal 10
- Divide by  $2^{bpl}$ 
  - $1818/2^5 = 56.815$
- Because the original number was negative. We know this because the most significant bit in the Binary point integer was 1, we negate the number to get the original value.
  - -56.815

This number now needs to be rounded to the correct number of decimal places. In the Master data dictionary definition of the data type the number of decimal places is specified. For a data type of this nature a value 1 would be appropriate

### 1.2.4 String Data Type

A string data type is an array of single byte characters. The range of valid characters is defined as US-ASCII only (0x20 - 0x7E). Strings do not have null termination characters at the end. Whenever a string is written, the entire string has to be written (no partial string writes are allowed). Padding for unused characters within a string should use spaces (0x20).

## 1.3 Mapping of Data Types into Modbus Registers

APC's UPS implementation of Modbus memory map will only use register (16-bit integer). Since usage data is not restricted to 16-bit registers, the following sections describe how the usage data is mapped into Modbus registers. Mapping of numeric data (BPI, Enumerations) is as follows:

### 1.3.1 Numbers

#### 1.3.1.1 Octet / Bytes (8 bits)

The upper 8 bits will be set to 0, the lower 8 bits will receive the data, and the resultant number will be stored as a 16-bit number

#### 1.3.1.2 16 bit Words

The number will be stored as a 16-bit number. 16 bit words are transmitted according to Modbus spec MSB first (Big-Endian).

#### 1.3.1.3 Odd Number of Octets (24 bits, 40 bits, etc)

This data size is not allowed, so there will be no mapping required.

#### 1.3.1.4 Long, and Long Long (32 and 64 bit numbers)

This data will be stored in 16-bit chunks crossing multiple Modbus registers. The MSW (highest 16-bits will be stored in the lowest Modbus register number) followed in sequential Modbus registers.

#### 1.3.1.5 Unusual sized numbers (48 bits, and >64 bits)

This data size is not allowed, so there will be no mapping required.

#### 1.3.1.6 Signed Numbers

Signed numbers use twos compliment encoding.

#### 1.3.1.7 Scaling of Numbers

Modbus registers are inherently integers. Scaling is done with base 2 multipliers in the same format as usage BPI's (See section

**Error! Reference source not found. Error! Reference source not found.),**

### 1.3.2 Bitfields

The Modbus register map will define the size of the bitfield that is being communicated. The spanning of registers will be done the same as for numbers (MSW will be lower register number). Bit #0 is the LSB of a register. Bitfields are treated as numbers and map to registers in the same manner as described in section 1.3.1 - Numbers.

### 1.3.3 Strings

The Modbus register map indicate the string size.. Mapping of the first byte of a string will be the MSB of the first register, second byte will be the LSB of the first register. The 3rd byte will be the MSB of the second register, etc. Strings will not be guaranteed to be terminated, unused characters will be filled with trailing 0x00 (NULL), 0x20 (Space), or 0xFF. Writing of strings is limited to 0x20 – 0x7E.

#### 1.3.3.1 Reading

When reading strings, leading zero values and trailing spaces should be stripped. For example:

“\0\0\0\0My UPS “  
should be displayed as:  
“My UPS”

In hexadecimal:

0x00004D7920555053202020202020202020

Should be displayed as:

0x4D7920555053

#### 1.3.3.2 Writing

When writing strings, the written value should adhere to these rules:

1. The string should be left-justified and padded with spaces to meet the register block size requirement (see 4.1.2)
2. The string should not contain a null terminator

Example: To write the string “My UPS” to a register block of 8 registers, the hexadecimal value written should be

0x4D79205550532020202020202020.

## 2 Transport Layers of Communication

The Modbus protocol will be available on a various physical layers. The layers are largely independent; however they use a common core of the Modbus design. Different products that use the Modbus protocol will only need to implement the physical layers that are relevant to that product.

### 2.1 SERIAL CONNECTION

The Serial interface uses RS-232 and is configured to 9600 BPS, No Parity, 8 Data bits, 1 Stop Bit. This physical

layer will use Modbus RTU (See Section Modbus RTU)

### 2.2 USB CONNECTION

USB v2.0 Full Speed is implemented via Human Interface Device (HID) usages. Set and Get Feature are implemented as well as asynchronous via Interrupt. Support for USB v3.0 is not available at this time. This physical layer implementation will use the Modbus RTU (See Section Modbus RTU) packets but will have some minor changes to support USB.

## 3 Modbus Implementation

This implementation is based on the Modbus Application specification, which can be found at:

[http://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf)

The following items in this section are areas of non-compliance or extensions to the above specification.

### 3.1 Register Blocks

In this implementation, registers can be combined into logical blocks for single data items that are larger than 16 bits.

#### 3.1.1 Register Block Reads

Reading blocks of registers is accepted as long as the combination of address and quantity of registers to read does not split a register block.

Example: Assume the following register blocks:

Data Item Index	Starting Register	Size
0	0	4
1	4	2

The following read operations and the results:

Address	Quantity of Registers	Response
1 or 2 or 3	X	Exception, Illegal Data Address
0	1 or 2 or 3	Exception, Illegal Data Address
0	5	Exception, Illegal Data Address
0	4	Data for Register Block 0
0	6	Data for Register Block 0 and 1
4	1	Exception, Illegal Data Address
4	2	Data for Register Block 1

#### 3.1.2 Register Block Writes

Similar rules apply to write operation, with the additional requirement that only a single register block may be written to in one Modbus command.

Example: Assuming the same register blocks as in the above example, the following write operations and the results:

Address	Quantity of Registers	Response
1 or 2 or 3	X	Exception, Illegal Data Address
0	1 or 2 or 3	Exception, Illegal Data Address
0	5	Exception, Illegal Data Address
0	4	Write Accepted to Register Block 0
0	6	Exception, Illegal Data Address
4	1	Exception, Illegal Data Address
4	2	Write Accepted to Register Block 1

### 3.2 Undefined Register Addresses

There are 2 types of undefined registers in a Modbus register map,

address blocks (function codes 0x03 Read Holding Registers and 0x04 Read Input Registers) that partially or entirely address undefined addresses will similarly return 0xFFFF for all undefined addresses.

#### 3.2.1 Undefined Register

An undefined register is one that has no definition (eg. Register 0x1FF has no definition). Reading and undefined register will be accepted and will return data value of 0xFFFF. Modbus reads of contiguous

Modbus writes that use undefined addresses are rejected with the Illegal Data Address exception code.

### 3.2.2 Not Applicable Register

A register that is not applicable is one that is defined but not applicable to the particular product in communication (eg. SwitchedOutlet 3 is not implemented). A Modbus read of this type of register will return the value in the definition of the register for un-implemented, this is typically a 0x0000 or 0xFFFF. When writing to a not applicable register a Modbus illegal address exception will be returned.

### 3.3 Writing a Modbus Register

When writing a usage that does not span Modbus registers (one that is 2 bytes in length), Modbus will return the value of the register after the write has been applied. The application in the master should

interrogate this response to determine if the write has worked properly. The most common reason that a write will not work is that the value written was illegal.

When writing a usage that spans Modbus registers (one that is more than 2 bytes in length and must be written with Modbus function code 0x16), the application should do a read using function code 0x4 to verify if the write has functioned properly.

When writing a command to verify that the command has worked properly a read of the associated status register should be done to determine the status of the command.

For example when writing a ReplaceBatteryCommand, the ReplaceBatteryStatus should be read to determine the result of the command.

## 4 Modbus RTU

Modbus RTU specification is based on the Modbus over Serial Line Specification and Implementation guide v1.02. This document is available on the [www.modbus.org](http://www.modbus.org) web site, at: [http://www.modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf)

The following sections describe the implementation details which differ from the specification.

### 4.1 MODBUS DEVICE ADDRESS

The slave address of the APC device will default to 1 unless changed via some other mechanism. Likely mechanisms for changing the address will be the local display. The address of a device if configurable will be from 1-223. Addresses 224-247 are prohibited as they are reserved for internal use.

### 4.2 UNSUPPORTED FEATURES

APC devices will not support the following features that are called out in the specification.

#### 4.2.1 ASCII Transmission Mode

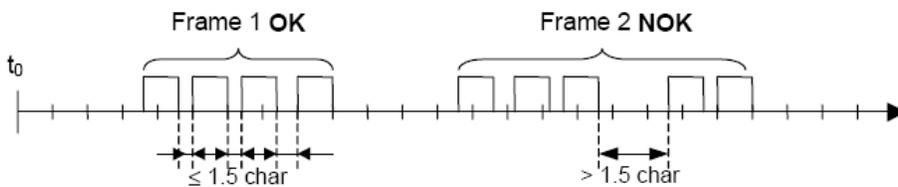
This is an optional feature and is not implemented.

#### 4.2.2 Modbus Message RTU Framing

Reference 2.5.1.1 of specification. Modbus framing of packet requirements are unusual since there is no start of character. For 9600 (No Parity) the following idle time before a packet is required:

1 Character = 1.04ms, 3.5 Character = 3.64ms and 4.5 Character = 4.6875ms

The specification calls out an inter-character timing of maximum of 1.5 characters = 1.56ms



However, for implementation purposes, APC devices are capable of detecting a minimum inter-character timeout of 15ms and require a minimum inter-frame interval of 35ms.

- Inter-character intervals of <15ms will not be detected.
- Abnormal end of packet (transmission error) uses the 35ms inter-frame timeout.

Master response timeout is application defined per the Modbus over serial line specification and implementation guide V1.02, section 2.4.1. This application requires a minimum 250ms Response Timeout.

### 4.3 Modbus RTU- RS232 Serial

Reference section 2.5.1 of Modbus over serial line specification and implementation guide V1.02. Modbus requires support of Even parity. To be compliant with APC devices using other protocols the default parity will be set to NO parity which differs from the specification. Even parity may not be supported.

Default baud rate is 9600, with 8 data bits, no parity bit, one stop bit, no flow control. This is a slight departure from the Implementation Class support defined in the Modbus over serial line specification and implementation guide V1.02:

Feature	APC Device	Modbus-Basic		Modbus-Default value
Addressing	Slave configurable from 1 to 227 Default = 1	Slave : configurable address from 1 to 247	Master : to be able to address a slave from address 1 to 247	-
Broadcast	No	Yes		-
Baud Rate	9600 - fixed	9600 ( 19200 is also recommended)		19200 (if implemented, else 9600)
Parity	NONE	EVEN		EVEN
Mode	RTU only	RTU		RTU
Electrical Interface	RS232 only	RS485 2W-cabling or RS232		RS485 2W-cabling
Connector	RJ 50 (10P10C)	RJ 45 ( recommended )		-

### 4.4 Modbus RTU- USB

#### 4.4.1 Overview

APC Smart-UPS devices enumerate as HID Power Devices. See the following documents for background information on USB HID Class and USB HID Class Power Devices:

[www.usb.org/developers/devclass\\_docs/HID1\\_11.pdf](http://www.usb.org/developers/devclass_docs/HID1_11.pdf)

[www.usb.org/developers/devclass\\_docs/pd/cv10.pdf](http://www.usb.org/developers/devclass_docs/pd/cv10.pdf)

Modbus RTU-USB communication is done via Interrupt Transfers, on Endpoint 1 of the APC device, through vendor-specific usages.

#### 4.4.2 USB Host Drivers

On Windows and Linux operating systems, a system driver is attached to the UPS device upon enumeration (when the device is connected to the host computer for the first time). In order to communicate via interrupt transfers, the default operating system driver must be replaced with an alternative driver.

#### 4.4.3 USB HID Usages



## [ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

The following USB HID usages are used in Modbus-RTU-USB:

Usage	Usage ID (hexadecimal)
ModbusRTURx	FF86:FC
ModbusRTUTx	FF86:FD

### 4.4.3.1 ModbusRTURx

This usage receives Modbus requests from the client (host PC) via interrupt OUT transfers.

### 4.4.3.2 ModbusRTUTx

This usage transmits Modbus responses from the server (Modbus RTU device) to the host PC via interrupt IN transfers.

### 4.4.4 Framing

Modbus RTU-USB requests and responses are identical to RS232-serial requests and responses, with two differences.

1. The USB 2.0 Full Speed interrupt transfer packets have a maximum payload of 63 bytes (after

subtracting one byte for the report ID), requiring longer Modbus frames to be broken into several interrupt transfers. The Modbus client and server will need to reassemble the fragmented frame before it can be processed. The unfragmented frame length is determined from the Modbus packet, see section 6 of the Modbus application protocol: [http://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf)

2. The 16-bit checksum is not included in Modbus RTU-USB. Data integrity is provided by the USB interface.

Note that the Modbus RTU device address is also used in Modbus RTU-USB. A Modbus RTU-USB packet looks like this:

USB HID Report ID	Slave Address	Function Code	Data
1 byte	1 byte	1 byte	0 up to 61 byte(s)

If the data payload is longer than 61 bytes, the following interrupt transfers will contain the remainder of the message, broken into 63-byte fragments:

USB HID Report ID	Data
1 byte	0 up to 63 byte(s)

A maximum of 5 interrupt transfers may be required to transmit the entire Modbus RTU message.

USB HID Report ID	Data
1 byte	0 up to 63 byte(s)

USB HID Report ID	Data
1 byte	0 up to 63 byte(s)

USB HID	Data	Unused

Report ID		
1 byte	0 up to 2 byte(s)	

## 5 Modbus Register Organization

All data is in holding registers, read with function code 3 (Modbus addresses 4xxxx). Single register writes use function code 6, and multiple register writes use function code 16. The following section describes the regions of the Modbus register map.

### 5.1 Status Data (0x0000 – 0x0007F)

Status and Alarm Registers.

### 5.2 Dynamic Data (0x0080 – 0x01FF)

The dynamic data consists of information that changes during the operation and consists of measurement information.

### 5.3 Inventory Information (0x0200 – 0x03FF)

This is information about the particular device, including SKU Name, Serial

Number, Firmware version, and Manufacturing Date. User Configuration strings will be in this region as well.

### 5.4 Static Data (0x0400 – 0x5FF)

The static data consists of string information, and configuration data that does not change frequently.

### 5.5 Commands (0x0600 – 0x6FF)

Write only commands that affect operation..

### 5.6 Protocol Verification (0x0800 – 0x8FF)

Read only data to verify protocol communications.

## 6 Enabling Modbus on Smart-UPS

### 6.1 Model compatibility

Smart-UPS in the SMT family will be able to use the new Modbus functionality, with exception of the SMT Rack Mount 1U models. Please visit [APC.com](http://www.apc.com) to view the SMT models currently available.

### 6.2 Firmware Update

UPS Firmware support for Modbus was added at version UPS 09.0 for the SMT product family. To use Modbus communications on the SMT family you must be running version UPS 09.0 or higher.

To find your current Firmware level, from the main menu of the LCD screen select "About" and scroll down to "UPS Firmware".

You may also run the Firmware Update Wizard and it will identify the version before performing the update. The Wizard can be found in the article referenced below.

For more information on firmware

updates, please consult the Knowledge Base Article FA164737 (How do I update the firmware of my SMT, SMX, or SMC series Smart-UPS?) in the Support section of APC.com (<http://www.apc.com/site/support/index.cfm/faq/>)

### 6.3 Enabling Modbus Communications

All APC Smart-UPS that support the Modbus communication interface ship with this feature disabled. To enable the Modbus communication functionality, in the front control console of the UPS, go to Advanced Menu mode, under Configuration and enable Modbus.

## Appendix A – Register List

Binary points are provided for most registers with a two-character designator. The first character will be either U, for unsigned, or S, for signed. The second character will be a number denoting the actual Binary Point Location.

Register	Size (bytes)	Register Name	Customer Description	Bin pt
0	4	UPSStatus_BF	The purpose of this register is to convey the mode of operation of the UPS at macro level. Anytime the value of this usage changes the UPSStatusChangeCause_EN usage will change as well. This usage is NOT intended to be a direct mapping to the internal UPS state machine. <b>See bit descriptions in Appendix B</b>	
2	2	UPSStatusChangeCause_EN	<b>See bit descriptions in Appendix B</b>	
3	4	MOG.OutletStatus_BF	<b>See bit descriptions in Appendix B</b>	
6	4	SOG[0].OutletStatus_BF	<b>See bit descriptions in Appendix B</b>	
9	4	SOG[1].OutletStatus_BF	<b>See bit descriptions in Appendix B</b>	
12	4	SOG[2].OutletStatus_BF	<b>See bit descriptions in Appendix B</b>	
15	4	SOG[3].OutletStatus_BF	<b>See bit descriptions in Appendix B</b>	
18	2	SimpleSignalingStatus_BF	<b>See bit descriptions in Appendix B</b>	
19	2	GeneralError_BF	<b>See bit descriptions in Appendix B</b>	
20	4	PowerSystemError_BF	<b>See bit descriptions in Appendix B</b>	
22	2	BatterySystemError_BF	<b>See bit descriptions in Appendix B</b>	
23	2	ReplaceBatteryTestStatus_BF	<b>See bit descriptions in Appendix B</b>	
24	2	RunTimeCalibrationStatus_BF	<b>See bit descriptions in Appendix B</b>	
25	2	Battery.LifeTimeStatus_BF	<b>See bit descriptions in Appendix B</b>	
26	2	UserInterfaceStatus_BF	<b>See bit descriptions in Appendix B</b>	

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
128	4	RunTimeRemaining	The number of seconds until power will go out when running on battery. This should never be compared as an actual value, but should be compared as LTE.	U0
130	2	StateOfCharge_Pct	This represents the % state of charge in the battery. (known as Battery Capacity in UPS-link).	U9
131	2	Battery.Positive.VoltageDC	This is the measured DC voltage. (Positive Half if there are 2 parts)	S5
133	2	Battery.Date	This is the number of days since 1999 (0=1-jan-2000). This is the theoretical battery replacement date, it should not be interpreted to be more accurate than a month.	U0
135	2	Battery.Temperature	This is the temperature in Degrees C.	S7
136	2	Output[0].RealPower_Pct	Phase 1 - Measured real power as a percent of full rating.	U8
138	2	Output[0].ApparentPower_Pct	Phase 1 - Measured apparent power as a percent of full rating. Never be implemented with ApparentPower usage.	U8
140	2	Output[0].CurrentAC	Phase 1 - This is the measured AC RMS Current.	U5
142	2	Output[0].VoltageAC	Phase 1 - This is the measured Output Voltage	U6
144	2	Output.Frequency	This is the measured AC frequency on the output.	U7
145	4	Output.Energy	This is the number of Watt Hours consumed	U0
150	2	Input.InputStatus_BF	<b>See bit descriptions in Appendix B</b>	
151	2	Input[0].VoltageAC	Phase 1 - This is the measured Input Voltage (0xFFFF if not applicable)	U6
154	2	Input.Efficiency_EN	Efficiency is defined as RealPowerOut / RealPowerIn. Apparent Power (VA) measurements should not be used.	S0

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
155	2	MOG.TurnOffCountdown_EN	Time remaining till output off -1-NotActive -no countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	S0
156	2	MOG.TurnOnCountdown_EN	Time remaining till output on- 1-NotActive - no countdown in progress.0- CountdownExpired-Countdown has ended.(1)- (32767)-Seconds remaining for countdown.	S0
157	4	MOG.StayOffCountdown_EN	Minimum time to remain off after a shutdown. -1-NotActive-No countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(2147483647)-Seconds remaining for countdown.	S0
159	2	SOG[0].TurnOffCountdown_EN	Time remaining till output off SOG 0 -1-NotActive_Cancel-Reading: no countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	S0
160	2	SOG[0].TurnOnCountdown_EN	Time remaining till output on SOG 0 -1-NotActive_Cancel-Reading: no countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	S0
161	4	SOG[0].StayOffCountdown_EN	Minimum time to remain off after a shutdown. SOG 0 -1-NotActive-No countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(2147483647)-Seconds remaining for countdown.	S0

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
163	2	SOG[1].TurnOffCountdown_EN	Time remaining till output off SOG 1 -1-NotActive_Cancel- Reading: no countdown in progress. 0-CountdownExpired- Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	S0
164	2	SOG[1].TurnOnCountdown_EN	Time remaining till output on SOG 1 -1-NotActive_Cancel- Reading: no countdown in progress. 0-CountdownExpired- Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	S0
165	4	SOG[1].StayOffCountdown_EN	Minimum time to remain off after a shutdown. SOG 1 -1-NotActive-No countdown in progress. 0-CountdownExpired- Countdown has ended. (1)-(2147483647)-Seconds remaining for countdown.	S0
167	2	SOG[2].TurnOffCountdown_EN	Time remaining till output off SOG 2 -1-NotActive_Cancel- Reading: no countdown in progress. 0-CountdownExpired- Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	
168	2	SOG[2].TurnOnCountdown_EN	Time remaining till output on SOG 2 -1-NotActive_Cancel- Reading: no countdown in progress. 0-CountdownExpired- Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	

©2013 Schneider Electric. All Rights Reserved. Schneider Electric and APC are trademarks owned by Schneider Electric Industries SAS or its affiliated companies. All other trademarks are property of their respective owners.

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
169	4	SOG[2].StayOffCountdown_EN	Minimum time to remain off after a shutdown. SOG 2 -1-NotActive-No countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(2147483647)-Seconds remaining for countdown.	
171	2	SOG[3].TurnOffCountdown_EN	Time remaining till output off SOG 3 -1-NotActive_Cancel-Reading: no countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	
172	2	SOG[3].TurnOnCountdown_EN	Time remaining till output on SOG 3 -1-NotActive_Cancel-Reading: no countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(32767)-Seconds remaining for countdown.	
173	4	SOG[3].StayOffCountdown_EN	Minimum time to remain off after a shutdown. SOG 3 -1-NotActive-No countdown in progress. 0-CountdownExpired-Countdown has ended. (1)-(2147483647)-Seconds remaining for countdown.	
516	16	FWVersion_STR	UPS Firmware Version	
532	32	Model_STR	Model Name	
548	32	SKU_STR	SKU Name	
564	16	SerialNumber_STR	Serial Number	
572	16	Battery.SKU_STR	The replacement Battery Pack for the internal battery pack (or the system if there is only one type)	
580	16	Battery.ExternalBattery.SKU_STR	The replacement Battery Pack for the external battery pack (or the system if there is only one type)	
588	2	Output.ApparentPowerRating	The rated apparent full power.	U0
589	2	Output.RealPowerRating	The rated real full power.	U0

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
590	2	I_SOGRelayConfigSetting_BF	<b>See bit descriptions in Appendix B</b>	
591	2	Date	Manufacture Date - This is the number of days since 1999 (0=1-jan-2000).	U0
592	2	Output.VoltageACSetting_BF	<b>See bit descriptions in Appendix B</b>	
595	2	Battery.DateSetting	Battery Installation Date, Days since 1999 (1/1/00 = 0)	
596	16	Name_STR	The name assigned to the UPS.	
1029	2	MOGTurnOffCountdownSetting_EN	Seconds of delay to use for a off. This value will be loaded into the TurnOffCountdown_EN when a delayed off command is requested.	S0
1030	2	MOGTurnOnCountdownSetting_EN	Seconds of delay to use for a on. This value will be loaded into the TurnOnCountdown_EN when a delayed on command is requested.	S0
1031	4	MOGStayOffCountdownSetting_4B	Seconds to keep an output off before starting it again. Typically minimum value of 4, maximum of 300.	S0
1034	2	SOG[0]TurnOffCountdownSetting_EN	Seconds of delay to use for a off. This value will be loaded into the TurnOffCountdown_EN when a delayed off command is requested.	S0
1035	2	SOG[0]TurnOnCountdownSetting_EN	Seconds of delay to use for a on. This value will be loaded into the TurnOnCountdown_EN when a delayed on command is requested.	S0
1036	4	SOG[0]StayOffCountdownSetting_4B	Seconds to keep an output off before starting it again. Typically minimum value of 4, maximum of 300.	S0
1039	2	SOG[1]TurnOffCountdownSetting_EN	Seconds of delay to use for a off. This value will be loaded into the TurnOffCountdown_EN when a delayed off command is requested.	S0

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
1040	2	SOG[1]TurnOnCountdownSetting_EN	Seconds of delay to use for a on. This value will be loaded into the TurnOnCountdown_EN when a delayed on command is requested.	S0
1041	4	SOG[1]StayOffCountdownSetting_4B	Seconds to keep an output off before starting it again. Typically minimum value of 4, maximum of 300.	S0
1044	2	SOG[2]TurnOffCountdownSetting_EN	Seconds of delay to use for a off. This value will be loaded into the TurnOffCountdown_EN when a delayed off command is requested.	S0
1045	2	SOG[2]TurnOnCountdownSetting_EN	Seconds of delay to use for a on. This value will be loaded into the TurnOnCountdown_EN when a delayed on command is requested.	S0
1046	4	SOG[2]StayOffCountdownSetting_4B	Seconds to keep an output off before starting it again. Typically minimum value of 4, maximum of 300.	S0
1049	2	SOG[3]TurnOffCountdownSetting_EN	Seconds of delay to use for a off. This value will be loaded into the TurnOffCountdown_EN when a delayed off command is requested.	S0
1050	2	SOG[3]TurnOnCountdownSetting_EN	Seconds of delay to use for a on. This value will be loaded into the TurnOnCountdown_EN when a delayed on command is requested.	S0
1051	4	SOG[3]StayOffCountdownSetting_4B	Seconds to keep an output off before starting it again. Typically minimum value of 4, maximum of 300.	S0
1536	4	UPSCCommand_BF	<b>See bit descriptions in Appendix B</b>	
1538	4	OutletCommand_BF	<b>See bit descriptions in Appendix B</b>	
1540	2	SimpleSignalingCommand_BF	<b>See bit descriptions in Appendix B</b>	
1541	2	ReplaceBatteryTestCommand_BF	<b>See bit descriptions in Appendix B</b>	

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Register	Size (bytes)	Register Name	Customer Description	Bin pt
1542	2	RunTimeCalibrationCommand_BF	<b>See bit descriptions in Appendix B</b>	
1543	2	UserInterfaceCommand_BF	<b>See bit descriptions in Appendix B</b>	
2048	4	ModbusMapID	Reports the Modbus map ID as a string, no null terminator.	STRING
2050	8	TestString	Always Reports "12345678" - included to debug end customer protocol byte order.	STRING
2054	4	Test4BNumber1	Always Reports 0x12345678 - included to debug end customer protocol byte order.	U0
2056	4	Test4BNumber2	Always Reports 0x-5 - included to debug end customer protocol byte order.	S0
2058	2	Test2BNumber1	Always Reports 0x1234 - included to debug end customer protocol byte order.	U0
2059	2	Test2BNumber2	Always Reports 0x-5 - included to debug end customer protocol byte order.	S0
2060	2	TestBPINumber1	Always Reports 128.5 - included to debug end customer protocol byte order.	S6
2061	2	TestBPINumber2	Always Reports -128.5 - included to debug end customer protocol byte order.	S6

©2013 Schneider Electric. All Rights Reserved. Schneider Electric and APC are trademarks owned by Schneider Electric Industries SAS or its affiliated companies. All other trademarks are property of their respective owners.

## Appendix B – Register Bit Descriptions

### UPSStatus\_BF

Bit Index	Description
0	Reserved
1	StateOnline-State - Indicating that the power for the output is being sourced from the input. Mutually exclusive with other state bits.
2	StateOnBattery-State - Indicating that the power for the output is being sourced from the battery. Mutually exclusive with other state bits.
3	Reserved
4	StateOutputOff-State - Indicates that the output is not powered through the UPS (Including any internal bypass paths). Some examples are: Off because of Fault or Low-Battery. Mutually exclusive with other state bits.
5	Fault-Modifier - Indicates that a fault of any severity (Warning, or Critical) is present in the system, which may have caused a transition.
6	6-InputBad-Modifier - Indicates that the input is not acceptable.
7	Test-Modifier - Indicates that a test is in progress.
8	PendingOutputOn-Modifier - Indicates that the state is pending output on (either on line, on battery, or bypass). Should only be set in combination with StateOutputOff.
9	PendingOutputOff-Modifier - Indicates that the state is pending output off. Master sets this whenever the ups is in process of turning off, or immediately when on battery for input bad. Will never be set in combination with StateOutputOff. When set the slave should monitor RunTimeRemaining. When / if run time is less than or equal to the slave's minimum run time threshold, the slave should start the shutdown process. This bit may also be set in conditions other than above, e.g., in bypass due to fault.
10	Reserved
11	Reserved
12	Reserved
13	HighEfficiency-Modifier - Indicates that the UPS is operating in a high efficiency mode (eg. green mode, Economy Mode, ECO Mode).
14	InformationalAlert-Modifier - Indicates that the UPS has an informational alert active. (eg. Lifetime Status near end).
15	Reserved

### SimpleSignalingStatus\_BF

Bit Index	Description
0	0-PowerFailure-Indicates that the input power has failed. Signal will be driven with output on or off. Complement of InputStatus.Acceptable.
1	ShutdownImminent- Indicates that the UPS is committed to disconnecting power from its output(s). The bit is set when UPSStatus_BF.PendingOutputOff is set AND RunTimeRemaining is less than or equal to LowRunTimeWarningSetting OR any of the following depending upon the UPS configuration.  * For UPS with an unswitched outlet group - when the UPSSystem.UnswitchedOutletGroup.TurnOffCountdown_EN is greater than -1

[ APPLICATION NOTE #176 ] **Modbus Implementation in Smart-UPS**

Bit Index	Description
	<p>* For UPS with no unswitched outlet group and with switched outlet group(s) - when the "last commanded" UPSSystem.SwitchedOutletGroup[x].TurnOffCountdown_EN is greater than -1.</p> <p>* For UPS with no unswitched outlet group and with no switched outlet groups - when the UPSSystem.OutputSystem.TurnOffCountdown_EN is greater than -1.</p> <p>In response to this bit becoming set, the device using the simple signaling interface should drive request to shutdown if it hasn't already done so (this ensures that TurnOffCountdown_EN timer will be set to at least the minimum time needed by the simple signaling host).</p>

**GeneralError\_BF**

Bit Index	Description
0	SiteWiring-A site wiring fault exists.
1	EEPROM-A eeprom fault exists.
2	ADConverter-An A/D converter fault exists.
3	LogicPowerSupply-A logic power supply fault exists.
4	InternalCommunication-A fault in the processor communication system.
5	UIButton-One (or more) of the Front Panel Buttons is not working properly.
6	Reserved
7	EPOActive-There is an active or unacknowledged Emergency Power Off signal.
8	Reserved

**PowerSystemError\_BF**

Bit Index	Description
0	OutputOverload-The output is overloaded (either real or apparent power).
1	OutputShortCircuit-The output is short circuited. [Put here for SU5G ease of implementation.]
2	OutputOvervoltage-The output voltage is too high.
3	Reserved
4	Overtemperature-Indicates that a component's temperature is too high.
5	BackfeedRelay-The backfeed relay (or its driver) has a fault.
6	AVRRelay-An AVR relay (or its driver) has a fault.
7	PFCInputRelay-A PFC input relay (or its driver) has a fault.
8	OutputRelay-An output relay (or its driver) has a fault.
9	BypassRelay-A bypass relay (or its driver) has a fault.
10	Reserved
11	PFC-A PFC fault exists.
12	DCBusOvervoltage-A DC bus voltage is too high.
13	Inverter-An inverter fault exists.
14	Reserved

**BatterySystemError\_BF**

Bit Index	Description
0	Disconnected-Indicates that the battery is electrically disconnected

Bit Index	Description
	(missing).
1	Overvoltage-Indicates that the battery voltage is too high.
2	NeedsReplacement-Indicates that the battery is at the end of its service life.
3	OvertemperatureCritical-Indicates that the battery temperature has exceeded a critical level. (Exclusive with OvertemperatureWarning)
4	Charger-A battery charger fault exists
5	TemperatureSensor-A battery temperature sensor fault exists.
6	BusSoftStart-A battery bus soft start fault exists.
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved

### ReplaceBatteryTestStatus\_BF

Bit Index	Description
0	Pending-Replace battery test is pending (high level acknowledgement of command).
1	InProgress-Replace battery test is in progress.
2	Passed-Replace battery test passed (completed successfully).
3	Failed-Replace battery test failed (completed unsuccessfully).
4	Refused-Replace battery test was refused (check "result modifier" bits for potentially additional details). Note: should not change source modifier when refusing a test as the refusal is always internal and the origin of the test would be lost any time the test is refused.
5	Aborted-Replace battery test was aborted (check "result modifier" and "source modifier" bits for potentially additional details).
6	Protocol-Source modifier: the protocol is the origin for initiation or abortion of the replace battery test.
7	LocalUI-Source modifier: the local user interface is the origin for initiation or abortion of the replace battery test. Includes local terminal mode interface if applicable.
8	Internal-Source modifier: internal control is the origin for initiation or abortion of the replace battery test.
9	InvalidState-Result modifier: invalid UPS operating state (e.g., shutdown pending, output off, ups in bypass, input voltage not acceptable).
10	InternalFault-Result modifier: an internal fault exists (e.g., battery is missing, inverter failure). Also, overload in progress which is not in the error usages.
11	StateOfChargeNotAcceptable-Result modifier: the battery state of charge is not acceptable.

### RunTimeCalibrationStatus\_BF

Bit Index	Description
0	Pending-Run time calibration is pending (high level acknowledgement of command).
1	InProgress-Run time calibration is in progress.
2	Passed-Run time calibration passed (completed successfully).
3	Failed-Run time calibration failed (completed unsuccessfully).
4	Refused-Run time calibration was refused (check "result modifier" bits for potentially additional details). Note: should not change source modifier

## [ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Bit Index	Description
	when refusing a test as the refusal is always internal and the origin of the test would be lost any time the test is refused.
5	Aborted-Run time calibration was aborted (check "result modifier" and "source modifier" bits for potentially additional details).
6	Protocol-Source modifier: the protocol is the origin for initiation or abortion of the run time calibration.
7	LocalUI-Source modifier: the local user interface is the origin for initiation or abortion of the run time calibration. Includes local terminal mode interface if applicable.
8	Internal-Source modifier: internal control is the origin for initiation or abortion of the run time calibration. Note: Internal should be reported if there is a "scheduled" internal test eg. every 3 months. Internal should also be used when a "natural" test completes successfully.
9	InvalidState-Result modifier: invalid UPS operating state (e.g., shutdown pending, output off, ups in bypass, input voltage not acceptable).
10	InternalFault-Result modifier: an internal fault exists (e.g., battery is missing, inverter failure). Also, overload in progress which is not in the error usages.
11	StateOfChargeNotAcceptable-Result modifier: the battery state of charge is not acceptable.
12	LoadChange-Result modifier: the load changed. (previous version has this incorrectly as bit 13 with no bit 12).
13	ACInputNotAcceptable-Result modifier: the AC input is not acceptable so the run time calibration was aborted.
14	LoadTooLow-Result modifier: the load is too low to recalibrate the run time accurately.
15	OverChargeInProgress-Result modifier: a battery overcharge is currently in progress therefore the run time calibration is refused (to prevent an inaccurate result).

### LifeTimeStatus\_BF

Bit Index	Description
0	LifeTimeStatusOK-Lifetime is OK. Mutually exclusive with bits 1-2
1	LifeTimeNearEnd-Lifetime is near end. Mutually exclusive with bits 0 and 2
2	LifeTimeExceeded-Lifetime is exceeded. Mutually exclusive with bits 0-1
3	LifeTimeNearEndAcknowledged-Alert has been acknowledged but still exists
4	LifeTimeExceededAcknowledged-Alert has been acknowledged but still exists

### UserInterfaceStatus\_BF

Bit Index	Description
0	ContinuousTestInProgress-The continuous local UI test is in progress.
1	AudibleAlarmInProgress-There is an active alarm that is causing the local UI beeper to sound. This bit indicates that the command to mute is available.
2	AudibleAlarmMuted-There is an active alarm that is currently being muted. This bit indicates that the command to cancel mute is available.
3	Reserved

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Bit Index	Description
-----------	-------------

Input.InputStatus\_BF

Bit Index	Description
0	Acceptable-Input (both voltage and frequency) is acceptable and all other system constraints are met so that the UPS can power the output with this input source.
1	PendingAcceptable-Input (both voltage and frequency) is acceptable but at least one other system constraint is not met preventing the line from being declared acceptable (e.g. line is not stable long enough).
2	VoltageTooLow-Indicates that the input voltage is too low to be acceptable.
3	VoltageTooHigh-Indicates that the input voltage is too high to be acceptable.
4	Distorted-Indicates a distorted input waveform. The input voltage is too different from reference waveform, the frequency is moving too fast to track, or the frequency is out of measurable range.
5	Boost-Indicates that the UPS is attempting to amplify the input voltage.
6	Trim-Indicates that the UPS is attempting to attenuate the input voltage.
7	FrequencyTooLow-Indicates frequency is measurably too low.
8	FrequencyTooHigh-Indicates frequency is measurably too high.
9	FreqAndPhaseNotLocked-Indicates that the system is not frequency and phase locked to the input frequency and phase.
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved

VoltageACSetting\_BF

Bit Index	Description
0	Reserved
1	VAC120-Output voltage 120VAC.
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved

AcceptableFrequencySetting\_BF

Bit Index	Description
0	Reserved

## [ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Bit Index	Description
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved

### UPSCommand\_BF

Bit Index	Description
0	Reserved
1	Reserved
2	Reserved
3	RestoreFactorySettings-Restore factory default settings for all operational parameters that can be safely returned to factory defaults. Output Voltage Setting and Output Frequency Setting are not altered. Strings, User Language settings, logs, and statistical information are not reset with this command.
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved

### SimpleSignalingCommand\_BF

Bit Index	Description
0	RequestShutdown - If there is no "shutdown" action in process this bit indicates a command to the master to shutdown. The UPS will accept this command regardless of the UPS State (Online or On Battery) It is the responsibility of the slave to only issue this command at the appropriate time.
1	RemoteOff - This is the equivalent of pressing and holding the power off button. This will execute an immediate off function of all outlets that are on and the UPS output.
2	RemoteOn - This is the equivalent of pressing the ON power off button. This will execute a sequenced on.

### Register: UPSStatus\_BF

Bit Index	Description
0	Start-Start the test.
1	Reserved

### RunTimeCalibrationCommand\_BF

Bit Index	Description
0	Start-Start the run time calibration.
1	Abort-Cancel the run time calibration if InProgress.

### UserInterfaceCommand\_BF

Bit Index	Description
-----------	-------------

[ APPLICATION NOTE #176 ] **Modbus Implementation in Smart-UPS**

Bit Index	Description
0	ShortTest-Perform the momentary local UI test, e.g., light all the LEDs and sound the beeper.
1	ContinuousTest-Perform the continuous local UI test, e.g., light all the LEDs and sound the beeper until canceled. To cancel, set UICommand_BF.ShortTest. Local muting should cancel this as well.
2	MuteAllActiveAudibleAlarms-Mute all the active alarms in the UPS. Will not silence the beeper during the short or continuous test or under other implementation specific reasons (for example, key click).
3	CancelMute-Cancels any muting (same as audible disabled then enabled).
4	Reserved
5	AcknowledgeBatteryAlarms-Acknowledge active battery alarms.

OutletStatus\_BF

Bit Index	Description
0	StateOn-State - indicates the outlet is powered. Mutually exclusive with other state bits.
1	StateOff-State - Indicates the outlet is not powered. Mutually exclusive with other state bits.
2	ProcessReboot-Modifier - indicates that a reboot command was issued and is still in progress. A reboot command can be issued by writing to the command bitfield or by writing timers. Mutually exclusive with other process bits.
3	ProcessShutdown-Modifier - indicates that shutdown command was issued and is still in progress. A shutdown command can be issued by writing to the command bitfield or by writing timers. Mutually exclusive with other process bits.
4	ProcessSleep-Modifier - indicates that a sleep command was issued and is still in progress. A sleep command can be issued by writing to the command bitfield or by writing timers. A master should indicate sleep rather than reboot if the StayOffCountdown_EN_4B timer is initially loaded with a value greater than 300 seconds. Mutually exclusive with other process bits.
5	Reserved
6	Reserved
7	PendingOffDelay-Modifier - indicates the outlet has an active process that requires an off delay when switching an outlet from on to off.
8	PendingOnACPresence-Modifier - indicates the outlet will not turn on unless AC input power is available.
9	PendingOnMinRuntime-Modifier - indicates the outlet will not turn on unless sufficient runtime is available.
10	MemberGroupProcess1-Modifier - indicates the outlet is participating in the 1st "group process command".
11	MemberGroupProcess2-Modifier - indicates the outlet is participating in the 2nd "group process command".
12	LowRuntime-Modifier - indicates the run time is below the setting for the outlet group.
13	Reserved
14	Reserved
15	Reserved

[ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Bit Index	Description
-----------	-------------

OutletCommand\_BF

Bit Index	Description
0	Cancel-Cancels pending actions to the targets selected. No modifiers are allowed.
1	OutputOn-Command the output to turn on. The only valid modifiers (in any combination) are UseOnDelay and ColdBootAllowed.
2	OutputOff-Command the output to turn off (and not come back on automatically). The only valid modifier is UseOffDelay.
3	OutputShutdown-Command the output to turn off and then back on automatically when AC input power is restored. The only valid modifiers (in any combination) are UseOffDelay and UseOnDelay. MinimumReturnRuntimeSetting is enforced when turning on.
4	OutputReboot-Command the output to turn off and then back on automatically. The only valid modifiers (in any combination) are UseOffDelay, UseOnDelay and ColdBootAllowed. MinimumReturnRuntimeSetting is not enforced when turning on. A Reboot command is interpreted as a sleep command when the stayofftime countdown is greater than 300 seconds.
5	ColdBootAllowed-Modifier - allow the output to turn on without AC input power conditions met.
6	UseOnDelay-Modifier - use the on delay settings for the applied command
7	UseOffDelay-Modifier - use the off delay settings for the applied command
8	UnswitchedOutletGroup-Target - command applies to the unswitched outlet group
9	SwitchedOutletGroup0-Target - command applies to switched outlet group 0
10	SwitchedOutletGroup1-Target - command applies to switched outlet group 1
11	SwitchedOutletGroup2-Target - command applies to switched outlet group 2
12	USBPort-Source - Command came from a device connected to the USB port
13	LocalUser-Source - Command came from a local user interface.
14	RJ45Port-Source - Command came from a device connected to the Computer Interface port (Typically RJ45), This includes software over the serial RJ45 and simple signal via RJ45
15	SmartSlot1-Source - Command came from a device in SmartSlot 1.
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved

## [ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Bit Index	Description
29	Reserved
30	Reserved
31	Reserved

### I\_SOGRelayConfigSetting\_BF

Bit Index	Description
0	MOGPresent-A user accessible MOG is present
1	SOG0Present-SOG 0 is present
2	SOG1Present-SOG 1 is present
3	SOG2Present-SOG 2 is present
4	SOG3Present-SOG 3 is present
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved

## Appendix C – Enumeration Descriptions

Enumeration for UPSStatusChangeCause\_EN

Value	Description
0	SystemInitialization-Indicates that the present state is achieved due to microprocessor reset. Value at start-up.
1	HighInputVoltage-A high input voltage condition caused the transition.
2	LowInputVoltage-A low input voltage condition caused the transition.
3	DistortedInput-A bad input condition (distorted voltage or unstable frequency, "turbo") caused the transition.
4	RapidChangeOfInputVoltage-A rapid change in the input voltage ("dV/dt") caused the transition.
5	HighInputFrequency-A high input frequency caused the transition.
6	LowInputFrequency-A low input frequency caused the transition.
7	FreqAndOrPhaseDifference-A difference in frequency and/or phase between the input and the system caused the transition.
8	AcceptableInput-An acceptable input (both voltage and frequency) caused the transition.
9	AutomaticTest-Indicates that a test has been initiated via the automatic timer in the UPS (or other programatic determination, e.g., power on). This can be any test, e.g., replace battery test or run time calibration.
10	TestEnded-Indicates that a test has been either completed (successfully or unsuccessfully) or aborted to cause the transition. Note that the only aborted causes that will be captured with this value are the ones that result in the same status after the test has been aborted. For example, a load change during a run time calibration that causes the test to abort and the status to return to on-line. As opposed to a local UI command (off button) that causes the run time calibration to be aborted but the status does not change to on-line.
11	LocalUICommand-Indicates the user pressed the on/off or other button locally to cause the transition. Includes local terminal mode interface if applicable.
12	ProtocolCommand-Indicates that a command received over the smart interface has caused the state change.
13	LowBatteryVoltage-A low battery voltage caused the transition. This would be used for low battery shutdown but may also be used when transitioning between other states due to a low battery voltage criteria.
14	GeneralError-A general error caused the transition. GeneralError_BF usage contains the specific fault if still valid.
15	PowerSystemError-A power system error caused the transition. PowerSystemError_BF usage contains the specific fault if still valid.
16	BatterySystemError-A battery system error caused the transition. BatterySystemError_BF usage contains the specific fault if still valid.
17	ErrorCleared-Indicates that the system changed states due to an error clearing. (Some errors may still exist but a state change occurred even with those errors present.)
18	AutomaticRestart-Indicates that internal conditions have met to allow the output to turn on, after a battery depletion. (8051 may not use this one because it requires EEPROM storage of the state).
19	DistortedInverterOutput-Indicates that the system changed states due to a distorted waveform detected on the output ("turbo").
20	InverterOutputAcceptable-Indicates that the system changed states due to no further distortion on the output waveform.

## [ APPLICATION NOTE #176 ] Modbus Implementation in Smart-UPS

Value	Description
21	EPOInterface-Indicates that an input was received at the UPS through the EPO interface to turn off the output.
22	InputPhaseDeltaOutOfRange-Indicates input phase delta is out of limit
23	InputNeutralNotConnected-Indicates that neutral leg is missing
24	ATSTransfer-Indicates that state change was caused due to ATS operation
25	ConfigurationChange-Indicates that state change was caused by a configuration change. (eg. a change in AllowedOperatingMode_BF)
26	AlertAsserted-An informational alert has caused the transition.
27	AlertCleared-Indicates that the system changed states due to an Informational alert acknowledge or cleared.
28	PlugRatingExceeded-Indicates transition happened because Input current exceeded plug rating. Example: when operating in "boost" mode when input current exceeds line cord rating transition to battery.
29	OutletGroupStateChange-Indicates the transition occurred due to MOG or SOG state change.
30	FailureBypassExpired-Indicates that load was turned off due to inability to continue operating in failure bypass.