

## pv - Python-based PV Inverter Monitoring Library

[HOME](#)[SOURCE CODE](#)[DOWNLOADS](#)[DOCUMENTATION](#)[DISCUSSIONS](#)[ISSUE TRACKER](#)[PEOPLE](#)[LICENSE](#)[Page Info](#) | [Change History \(all pages\)](#)★ [Follow \(3\)](#) [Subscribe](#)[Documentation](#) > [Protocol](#)

Search Wiki &amp; Documentation

## Communication Protocol

### Caution

Note that this protocol is not the manufacturer's official specifications. In fact, we know that while this protocol is consistent with the CMS2000 inverter, it is slightly inconsistent with the CMS5000A's protocol. This has yet to be resolved.

Let me know if you have a CMS inverter connected via serial and would like to help figuring out the CMS inverter communication protocol..

### Serial port parameters

The inverter uses a standard set of parameters (9600 8N1) for serial communication:

Baud rate = 9600  
 Byte size = 8  
 Parity = None  
 Stop bits = 1  
 Software/Hardware flow control off

### Frame structure

Let's begin by looking at an example serial communication frame (hex encoded):

**aaaa 0000 0100 0080 0a 31303034424f31323530 03fc**

**aaaa** - two preamble bytes for synchronisation  
**0000** - frame source address  
**0100** - frame destination address  
**0080** - frame type; represents a command and the corresponding structure of the payload.  
**0a** - frame payload size as an unsigned byte  
**31303034424f31323530** - frame payload, the actual data carried in this frame  
**03fc** - frame checksum, as a modular sum of all bytes in the frame, not including the checksum itself

### Protocol

This is the protocol as understood from observing the communication between the inverter and the client via an RS-232 serial interface. SEND and RECV is as viewed from client side.

#### 1. Reset the connected device

SEND aaaa 0100 0000 0004 00 0158



No reply is expected. The source address is not limited to 0x0100, but can be anything.

#### 2. Discover if there is an inverter connected

SEND aaaa 0100 0000 0000 00 0154

If an inverter is connected and is online, it replies with its serial number, so we receive:

RECV aaaa 0000 0100 0080 0a 31303034424f31323530 03fc

The payload of this frame contains the serial number of the inverter in ASCII encoding. In this example, the serial number is "1004BO1250"

#### 3. Associate with the discovered inverter

Having found the inverter's serial number, we must register that inverter with us to perform further operations. In the following broadcast (dst=0x0000) frame payload, we are first specifying the serial number of the device we wish to register with us, and at the same time, we assign an address of 0x0001 to that device.

```
SEND aaaa 0100 0000 0001 0c 31303034424f31323530 0001 0381
RECV aaaa 0001 0100 0081 01 06 01de
```

The inverter confirmed its registration by replying with an OK, using its newly assigned address.

#### 4. (Optional) Query the inverter for an extended identification string

```
SEND aaaa 0100 0001 0103 00 015a
RECV aaaa 0001 0100 0183 40
3120203230303041312e303020202020434d5320323030302020202050484f454e49585445432020202020203
1303034424f313235300000000000000033363030 0db0
```

In this frame, the inverter sends a 64-character ID string "1 2000A1.00 CMS 2000 PHOENIXTEC 1004BO1250\x00\x00\x00\x00\x00\x00\x003600"

#### 5. (Optional) Query the inverter's parameters

Before we can read parameters from the inverter, we need to find out what parameters the inverter has and how the inverter is going to send the values.

```
SEND aaaa 0100 0001 0101 00 0158
RECV aaaa 0001 0100 0181 06 404144454647 0375
```

The inverter replies with a sequence of bytes, each representing a data field. Knowing the message structure, we can use that information to interpret the inverter's parameters.

```
SEND aaaa 0100 0001 0104 00 015b
RECV aaaa 0001 0100 0184 0c 05dc003c080c0a50133d13d3 04a8
```

Interpreting the fields and values also require some additional information about their mapping.

Fac-Max	50.75
Fac-Min	49.25
Vac-Min	206.0
Vac-Max	264.0
Vpc-Start	150.0
T-start	60

#### 6. Query the inverter's status

Before we can read the inverter status, we need to find out what status information the inverter can provide. We query its structure:

```
SEND aaaa 0100 0001 0100 00 0157
RECV aaaa 0001 0100 0180 14 004041424344454748494a4c78797a7b7c7d7e7f 08c4
```

Likewise with the inverter parameters, we can use the layout information to interpret the status information we obtain from the inverter:

```
SEND aaaa 0100 0001 0102 00 0159
RECV aaaa 0001 0100 0182 28
01670bf9000309f1138b0036ffff00002cfb000007a900010000000000000000000000000000 0914
```

Vpv	306.5
Vac	254.5
Error	0
Zac	65535
Iac	0.3
Pac	54
Fac	50.03
h-Total	1961
Mode	1
Temp-inv	35.90

E-Total	1151.5
---------	--------

Last edited Dec 5, 2010 at 12:57 PM by [EdmundTse](#), version 1

## COMMENTS

No comments yet.

[Sign in to add a comment](#)

---

© 2006-2012 Microsoft   [Get Help](#)   [Privacy Statement](#)   [Terms of Use](#)   [Code of Conduct](#)   [Advertise With Us](#)   Version 12.12.2012.19766