Sourced: <https://reisfun.wordpress.com/>

## [Reverse engineering is fun, part 1](https://reisfun.wordpress.com/2010/09/29/re_part1/)

Posted on [September 29, 2010](https://reisfun.wordpress.com/2010/09/29/re_part1/) by [reisfun](https://reisfun.wordpress.com/author/reisfun/)

Three months ago, we got a 10 kW solar power plant installed on our roof. (Actually it was in June, a mere week before the guaranteed prices for generated PV energy were lowered here in Germany.)

It generated a whole lot of electricity during July and some more in August (lousy weather then), but it became boring to read the energy meter each evening to see results. Well, both inverters have a RS-232 connector and being an electronic engineer and embedded Linux software developer this sounded like an easy task for a small Linux server. (Honestly – really no one would consider using a Windows PC consuming 1% of the generated energy and rebooting every other day?) So, just get some software for logging the solar inverter’s internal meters, install it and put the results on an Apache server.

That’s the theory, in reality I got two different brands of solar inverters installed. One is called Effekta, the other Solpower. Both are German companies, but I already suspected, that at least one of them just buys some Asian inverters and put their logo on it.

So let’s start with the Effekta, a quick visit to their website and voilà, there’s a Windows logging software for free but no Linux version. OK, I really didn’t expect to find finished software for **apt-get** of **yum**, but… So, starting up a VM, booting Windows there and install the software. Hmm, at least it works and shows me what data the inverter has to offer. Firing up **portmon** and restarting the software shows data packets like this:

01 03 C0 20 00 10 plus some changing checksum/CRC

Firing up Google to look for a tool to calculate checksums… Found a [nice site](http://www.lammertbies.nl/comm/info/crc-calculation.html) calculating different checksums for data packets – and bingo. The site says the crc looks like a CRC16 in modbus flavour. OK, wikipedia says that modbus is a standard for industrial control and I soon find a link to a Linux lib called libmodbus. Fine, Effekta seems to be solved. To be sure, I wrote an email to Effekta and asked for technical specs of their inverter’s interface. The next morning (!) I got a reply and a detailed PDF confirming my findings

Thanks a lot again here!

## [Reverse engineering is fun, part 2](https://reisfun.wordpress.com/2010/09/29/re_part/)

Posted on [September 29, 2010](https://reisfun.wordpress.com/2010/09/29/re_part/) by [reisfun](https://reisfun.wordpress.com/author/reisfun/)

Well, the Solpower inverters proved to be much more complicated. I tried the same (Effekta) strategy here and visited the Solpower website. No mention of the inverter here. OK, write an email to the guys who installed the stuff on my house, but no luck, just an apologetic reply and no result after a couple of days

Seems I need to put more efford into this:

First try: connecting the Effekta Windows software on the Solpower inverter – no luck.

Second try: firing up a terminal program, connecting it and try some characters – no luck, not even an echo of my sent data.

Third try: the Solpower inverter has an USB port. Fine, let’s connect my MacBook to it and see what happens. Nothing, OK, I expected this. But let’s look what Mac OS X says about the device. Manufacturer ID $0D9F? Google tells me this is a Powercom Ltd. device. Fine, first useful hint…

Google again and I find a couple of powercom companies but only one with solar inverters. It’s Powercom Ltd. from Taiwan (<http://www.pcmups.com.tw>). Looking around on their website I find the solar king inverters which look really, really like my Solpower inverters, just a different color. OK, got you

They even offer some Windows software but again, no Linux solution. OK, download, startup virtual Windows, drink espresso and install it (did I mention I drink way too much espresso?). Hmm, ugly software (you know that kind of chinese software with english texts squeezed into buttons sized for chinese characters?), but it connects to my inverters on the first try. Nice, so fire up **portmon** again and look at it. No modbus protocol this time. But it looks like a simple question-answer game again:

Question: BB BB 01 00 00 01 01 02 00 01 7B

Answer: BB BB 00 01 01 00 01 82 2A 02 B2 0A 61 00 2B 00 00 00 F4 00 01 00 00 00 00 00 00 00 00 00 00 00 2F 09 31 13 86 04 3F 00 38 00 00 0A 52 00 00 00 00 00 00 06 3D

A lot of data in one packet, but a short look at the decoded data in the Windows software helps decoding at least a couple of byte. So, let’s try if they tell me how it works in reality. There’s a support page on their website and a couple of seconds later a sent my polite request.

Hello,  
I’ve got a Solarking SLK-3000 inverter. I want to log the machine with Linux. Would you please send my documentation about the serial protocol?

Thanks!

OK, let’s wait and see (drink some espresso in the mean time

Three days later I get a response from Powercom’s [ELIZA](http://en.wikipedia.org/wiki/ELIZA):

Dear Sir,

Solar Control system for SLK inverter can only work under Windows system.

Thank you.

Regards,  
Amos

Well Amos, that was the reason for asking in the beginning. OK, maybe Amos had a bad day with his english and I retry it very politely explaining my need more elaborated … One week later – no answer… (In fact, no answer up to today, about six weeks later.)

This looks like one of these hard cases of chinese suppliers we see regularly at my daytime job: Wanna buy? Here it is? Need support, have question? Sorry, don’t understand english… The chinese really need to learn that having a manufacturer-customer relationship doesn’t end after paying (our Japanese suppliers understood this!), so that’s why I decided to decode the protocol and publish it…

## [Reverse engineering is fun, part 3](https://reisfun.wordpress.com/2010/09/29/rev_part3/)

Posted on [September 29, 2010](https://reisfun.wordpress.com/2010/09/29/rev_part3/) by [reisfun](https://reisfun.wordpress.com/author/reisfun/)

OK, basically this inverter seems to work the same way as the modbus protocol, open RS-232, write request, wait for answer, decode, close RS-232. So I decided to clone the libmodbus, rename it libpowercom and replace some parts like the CRC with some other checksum code.

Let’s begin with the checksum. Playing around a little bit with the online CRC tool from part 1, it looks like Powercom doesn’t use a sophisticated CRC calculation but a simple 16 bit checksum. Hey guy over there – didn’t anybody at engineering school tell you why one should use a CRC instead of checksums? You wouldn’t even notice a dropped zero data byte or swapped bytes, come on guys, you can do better…

The remaining parts of the packet seem to be some kind of command with parameters. There are basically three different types:

BB BB 01 00 00 01 01 02 00 plus checksum asks for data

BB BB 00 00 00 00 00 00 00 plus checksum asks for the serial number of the inverter

BB BB 00 00 00 00 00 01 0C plus the serial number plus checksums “logs into” the inverter

The answers can be decoded like this:

Log in packet: BB BB 00 01 00 00 00 81 01 06 plus checksum means “log in OK”

Serial number request packet: BB BB 00 00 00 00 00 80 0B 31 30 31 30 30 32 36 31 30 30 35 plus checksum, meaning a serial number of “10100261005”

Data packet:BB BB 00 01 01 00 01 82 2A plus 21 words big endian meaning:  
0    02 B2 = 690d -> 69,0 C heat sink temperature  
1    0A 1C = 2588d -> 258,8V DC (panel) voltage  
2    00 2C = 44d -> 4,4A DC (panel) current  
3    00 00  
4    00 F4 = 244d -> 244 working hours  
5    00 01 ??  
6    00 00  
7    00 00  
8    00 00  
9    00 00  
10    00 00  
11    00 2F = 47d -> 4,7A AC (line) current  
12    09 31 = 2353d -> 235,3V AC (line) voltage  
13    13 85 = 4997d -> 49,97Hz (line) frequency  
14    04 3F = 1087d -> 1087W AC (line) power  
15    00 37 = 55d another temperature?  
16    00 00  
17    0A 52 = 2642d -> 264.2kWh accumulated energy  
18    00 00  
19    00 00  
20    00 00 plus checksum…

So, fine. If anybody out there know what words 5 or 15 mean, don’t hesitate to write me

It took me less than an hour to modify the libmodbus into libpowercom. (I will put this on sourceforge the next days, promised…) Checking it was much more difficult, as the sun has set already and the inverters don’t talk to you when it’s dark. So wait for the weekend.

On weekend it took me some hours to install three RS-232 cables, install a fresh Debian on an old PC, connect an USB hub with three USB-RS-232-converters and put everything together. After way too much espresso, it works. **crond** starts my new tool every five minutes and the actual power and accumulated energy is logged into a csv file. Putting this into some nice PNG graphics, embedded in nice HTML+CSS into is another story. However, it works. The next task will be to put that code into one ARM linux boards from work (consuming less than 1W fully running), but this is yet another story.

## [Reverse engineering is fun, update](https://reisfun.wordpress.com/2010/10/01/reverse-engineering-is-fun-update/)

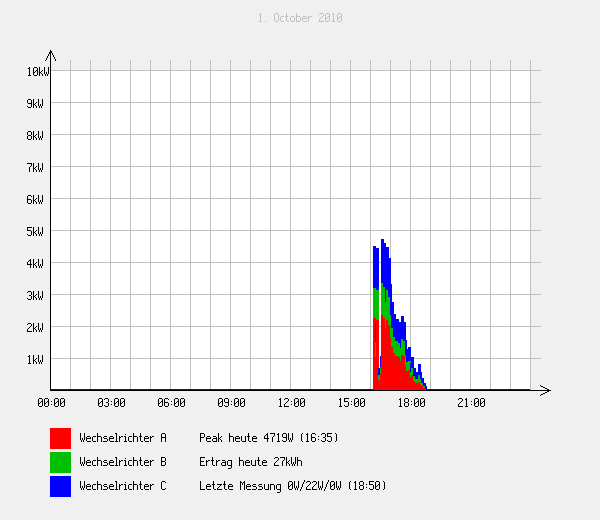
Posted on [October 1, 2010](https://reisfun.wordpress.com/2010/10/01/reverse-engineering-is-fun-update/) by [reisfun](https://reisfun.wordpress.com/author/reisfun/)

I did some more work on it and I can now read my Effekta and Powercom internal data. I attached an automatically generated PNG from the read data (sorry for the texts being German). A friendly G’day to Darryl from Australia for giving me some pointers about the unknown registers. I also found out the algorithm the Powercom software uses to connect to the inverter:

1. try to read the inverter’s register -> OK? end then

2. try to read the inverter’s serial number, possibly repeat this several times until an answer comes -> no answer? bad luck…

3. use the received serial number to log into the inverter -> got an ACK (ASCII $06)? fine, so restart at 1

[](https://reisfun.files.wordpress.com/2010/10/tag.png)

## [Reverse engineering is fun, publishing results](https://reisfun.wordpress.com/2010/10/03/reverse-engineering-is-fun-publishing-results/)

Posted on [October 3, 2010](https://reisfun.wordpress.com/2010/10/03/reverse-engineering-is-fun-publishing-results/) by [reisfun](https://reisfun.wordpress.com/author/reisfun/)

OK folks – the library for talking to the Powercom solar inverter is working. As it’s derived from another library there’s a lot of overhead which is no longer used, but it’s a working beta… I’ve tested it on Ubuntu 10.04 and Debian 5.0.

Visit <http://libpowercom.sourceforge.net> for downloading.

Feedback appreciated…