

HSPY-1000-001:

1000V 0.1A Digital Adjustable Variable Lab Bench Power Source High Voltage Programmable DC Regulated Power Supply (The one on the photo is 0,3 A)



I bought mine to calibrate duspols on 24, 50, 120, 230, 400 and 690 Vdc. So I choose the 0,1 A type, since I only want to check the voltage, more current makes it more expensive.

I bought mine for € 236,95 + local taxes. It is now 1 month later € 275-

Since I'm going to change the current 6 times every measurement, I wanted to use the programmable part, since using the buttons is only nice for a single setting. But continuously changing the setting using the device interface, is rubbish.

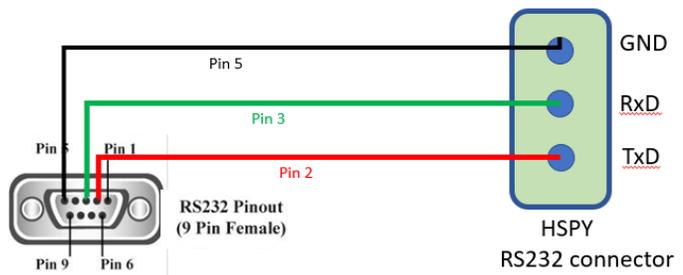
Only one place on the internet with nice details: <https://www.eevblog.com/forum/testgear/hspy-power-supply/>. DPENKLER from this blog posted a very useful modbus-RTU overview on the HSPY-xxxx, I still could not program the device. Since it took me way to much time to figger it out, I'm going to share this, this device really is intended for quick shifting.

Use windows to program the HSPY.

You will need a serial cable, the one that came with my device is wired like the drawing on the right.

On the device ground (black) is on top,

Rx in the middle (Green) en Tx down (Red).



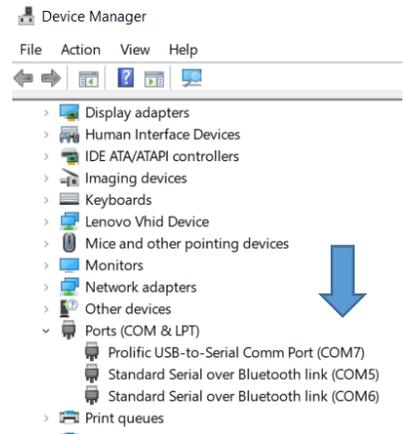
Connect the power supply to windows



Since I wanted to connect to windows, I used a standard **usb to serial cable**.

Those are about € 8.00 en are sold everywhere.

- Connect the cable to the USB port.
- Start the device manager.
- Open ports (Com & LPT)
- Look for the COMx port name belonging to your usb-serial adapter.
- Remember the port name, close the device manager.
- Connect the USB-Serial cable with the Device serial cable.



Your hardware is now connected for programming.

Setting up the power supply HSPY-xxxx-yyy.

The power supply comes with some default setting as shown to the right.

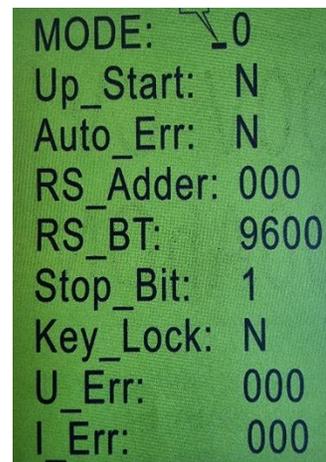
You can enter the settings, by turning on the device HSPY-xxx en then press the rotary-button once.

You can scroll through them by pressing the **U/I**-button.

By turning the rotary-button, you change the value.

When done, press the rotary-button again and your back to normal display.

1. Set Mode to N
2. Set the device (slave) address (RS-Adder) to 001, or whatever might make you happy
3. Set speed (RS_BT) to 9600
4. Set Stop_Bit to 1



Your device is now ready.

I will create a separate document from the original Chinese manual later.

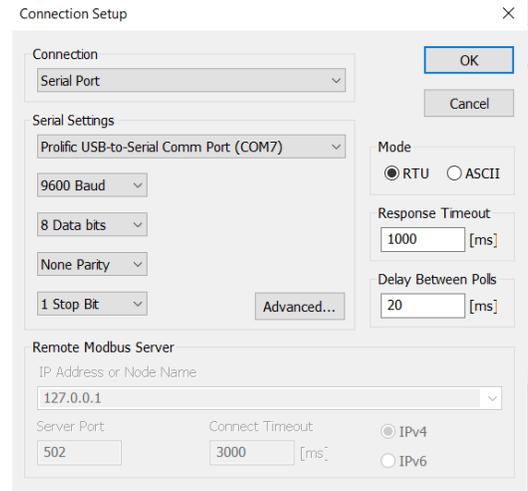
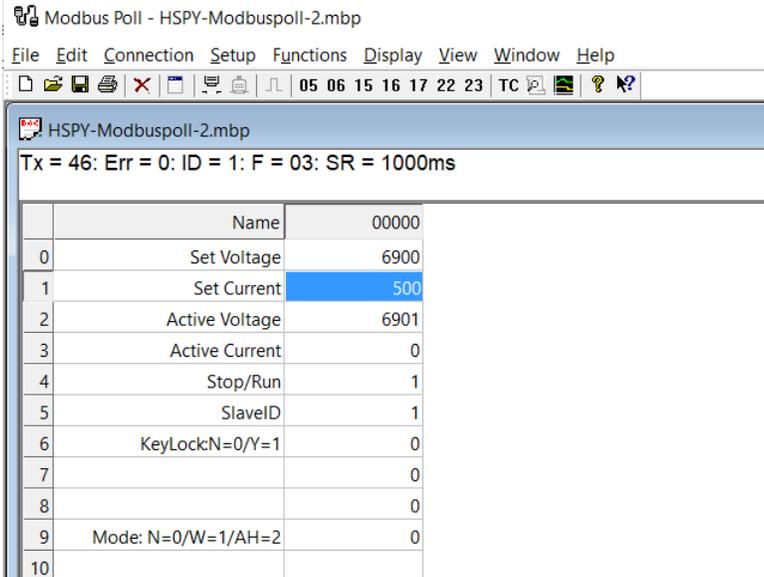
Setting up windows

I used 2 programs, to program the device.

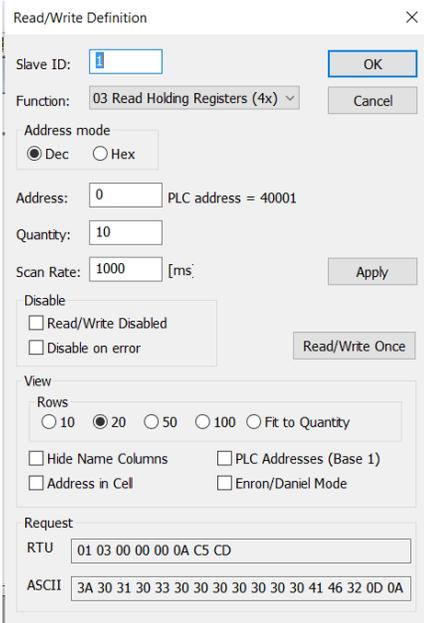
1. Windows based
2. Command-prompt based

Windows based software:

Modbus Poll, license based, 30 days free, need to restart after 10 minutes. Still very useful while experimenting. Download it from <https://www.modbustools.com/>



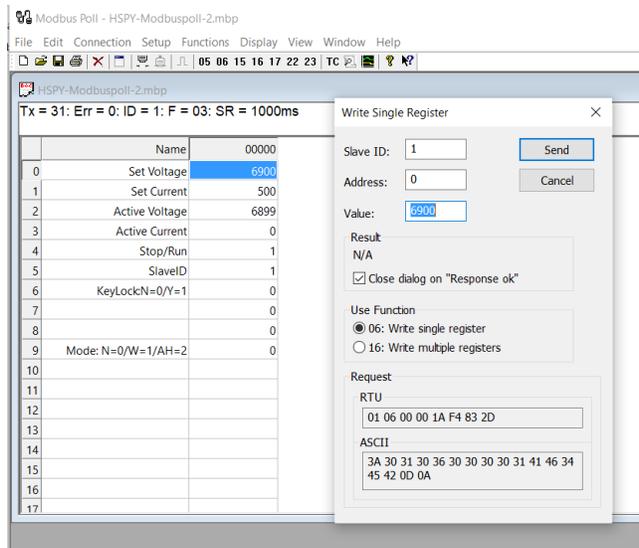
You need to do the connection setup, since you have set the power supply earlier, you only need to set the same settings to this software, remember to set the RTU mode (Modbus over RS232 remember?)



As you can see in the overview on the topleft, , you can name the registers and save this setting. I like it, since I forget that the fifth register (Line nr4) is on/off for the Aux clamps on the device, etc

You get this list by setting the Read/Write definition to “03-Read Holding Registers”

Start at adres 0, ask for 10 register (quantity) and I left the scanrate default.



I default set the focus on register 0 (Set Voltage) data field.

As soon as you start typing numbers, the “Write Single Register” pops up. The first time you must set the checkmark “Close Dialog on response OK”.

Enter the desired value. You must specify the voltage you want, including the digit behind the comma, without using the comma.

So if you want 15,5 volt, you type 155.

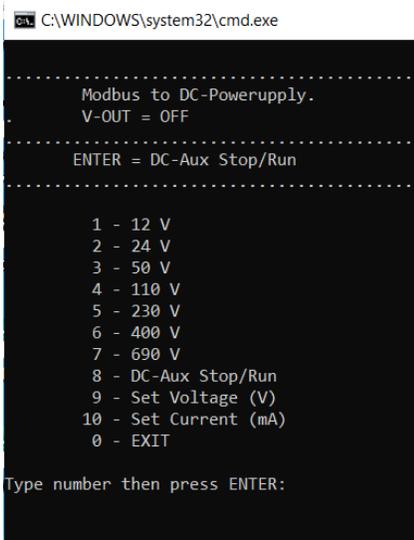
You want 400V, you type 4000 and hit enter.

The dialog closes. As soon as you start typing a new number, the dialog pops up from the field that has focus.

You can set the current, the voltage, Stop/Run, KeyLock (Rotary-button off) and Mode .You can only read the active current and voltage.

That’s it, I like this program a lot, it save you tons of time not using the knobs. I did not actually buy it, since I want to trigger program-changes to the device from an excel spreadsheet, where I log my measurements for customers. Or from a Arduino, being able to use the power supply without a PC. More over that, in the next chapter.

Using a commandline tool



I did have trouble finding a free active-x control, or other tool , simple to use. In the end, I found Modpoll.exe. A standalone executable, no install needed and FREE (Read the license).

You can download it from :

<https://www.modbusdriver.com/modpoll.html>

Since I’m calibrating Meters and duspols, I use standard volt-settings.

I made a dos batch file to switch through the several settings.

I made a vba Form in Excel, that exits to the shell for a moment.

Being able to switch without effort, makes this power supply a multitool.

I will also supply the batch file, you need to adapt the path to the executable, the Com-port and the slave number. I added some error trapping.

How to use Modpoll?

Reading: How to use? Run Modpoll.exe -h from the command line, you only need a few parameters:

- -b The baudrate, for this device mandatory 9600
- -p Parity NONE
- -m rtu Modbus RTU protocol (default if SERIALPORT contains \ or COM)
- -1 If retrieving data, otherwise the program keeps polling. Not needed when writing
- -a Slave ID, default 1
- -c Count off items to read, for this device always 1.
- -r The register to read or write, the HSPY know the following
 - ➔ 1 = Set Voltage (Read/Write)
 - ➔ 2 = Set current in milliamp (Read/Write)
 - ➔ 3 = Active Voltage (Read)
 - ➔ 4 = Active Current (read)
 - ➔ 5 = Stop/Run (Read/Write) value 0/1
 - ➔ 6 = Keylock (Read/Write) value 0/1 Locks the rotary part of the rotary button, does not lock the push function, since you need that to alter the settings
 - ➔ 7 & 8 = Did not find that yet .
 - ➔ 9 = Device Mode (Read/Write), value 0,1,2
Where 0 = normal, Voltage & current cooperate, Voltage drops if current to low
I did not find Mode W and AH yet, will be covered later.
- COMx The port your USB-Serial uses.

So for reading the commandline looks like this:

```
Modpoll.exe -b 9600 -p none -m rtu -1 -a %SlaveID% -r %Register% -c 1 %ComPort%
```

Reading the Run/Stop button: Register =5, my Comport =7, My slaveID=3

```
Modpoll.exe -b 9600 -p none -m rtu -1 -a 3 -r 5 -c 1 com7
```

```
C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch>modpoll -b 9600 -p none -m rtu -1 -a 3 -r 5 -c 1 COM7
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC3
Slave configuration...: address = 3, start reference = 5, count = 1
Communication.....: COM7, 9600, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, output (holding) register table

-- Polling slave...
[5]: 0
```

At the end you see “[5]: 0”, meaning My stop/run button is off.

```
C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch>modpoll -b 9600 -p none -m rtu -1 -a 3 -r 5 -c 1 COM7
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC3
Slave configuration...: address = 3, start reference = 5, count = 1
Communication.....: COM7, 9600, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, output (holding) register table

-- Polling slave...
[5]: 1
```

Same run, but now the button is on.

Writing: How to use?

The writing of a register is nearly the same as reading it. You omit the -1 and at the end you set the value needed.

```
Modpoll.exe -b 9600 -p none -m rtu -a %SlaveID% -t 4 -r %Register% -c 1 %ComPort% %Value%
```

So I want to set the voltage: Register =1, my Comport =7, My slaveID=3, Voltage=50V
Voltage and current are written with the digit behind the comma, without the comma

So writing 17,5 V means you type 175

So writing 400 V, means you write 4000.

You can write any number, the device is limited to its max.

```
Modpoll.exe -b 9600 -p none -m rtu -a 3 -r 1 -c 1 com7 500
```

```
C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch>modpoll -b 9600 -p none -m rtu -a 3 -r 1 -c 1 COM7 500
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC6
Slave configuration...: address = 3, start reference = 1, count = 1
Communication.....: COM7, 9600, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, output (holding) register table

Written 1 reference.
```

Reading this value:

```
C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch>modpoll -b 9600 -p none -m rtu -1 -a 3 -r 1 -c 1 COM7
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC3
Slave configuration...: address = 3, start reference = 1, count = 1
Communication.....: COM7, 9600, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, output (holding) register table

-- Polling slave...
[1]: 500
```



As you see, 500, but on display 50,0V

Writing 1500V (my device = 1000V)

```
C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch>modpoll -b 9600 -p none -m rtu -a 3 -r 1 -c 1 COM7 15000
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC6
Slave configuration...: address = 3, start reference = 1, count = 1
Communication.....: COM7, 9600, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, output (holding) register table

Written 1 reference.
```



On display the max my device is limited to: 999,9V

DOS-Batch program

As shown earlier I wrote a dos-batch for my own purpose. It does some extra handling, like checking if Aux is on or off and displaying that on the second line.

It does some basic checking. You can select preset voltages, but you can also set Voltage (9) and Current (10)

I will upload this batch file (**PowerDC.cmd**) as a separate file.

You will need, to download the executable this batch uses.

You can find it: <https://www.modbusdriver.com/modpoll.html>

Put the downloaded modpoll.exe somewhere on your PC.

Open PowerDC.cmd and modify the following lines:

```
REM *** PLEASE ADAPT NEXT 3 SETTINGS TO YOUR SITUATION ****  
set ModExe=C:\Users\Johan_2\Documents\Arduino\HSPY-Voeding\Batch\modpoll.exe  
set ComPort=Com7  
set SlaveID=1
```

Specify you settings, and your ready to go!

Enjoy! The lack off information on internet over this device is strange, since it is sold at many places. Please share!

Arduino:

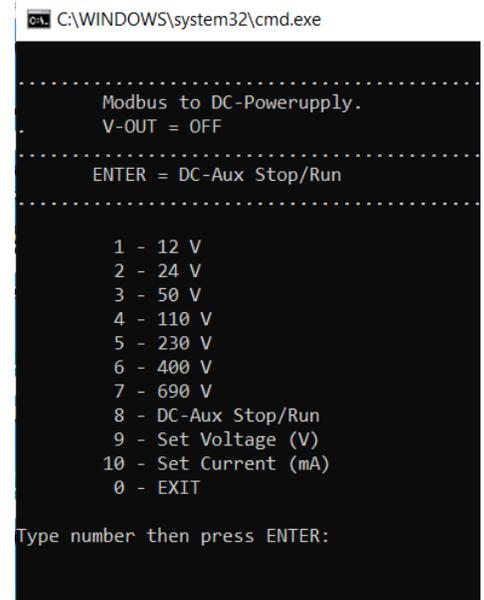
I am busy, getting an Arduino programming the device. I'm not really a network specialist, so I'm still fighting. If finished, I will publish the used components and the code.

Using the Arduino with a display, makes unattended use possible. I still want to select the Voltages, but also want Step-up en step-down buttons.

I want to use relais, to switch between my AC and DC power supply and also connect to my resistance testing, doing so, it is no longer needed to switch connection to the testing device.

Have a nice day and be happy with your HSPU-xxx-yyy!

Kind regards, Johan Mulder / The Netherlands.



```
C:\WINDOWS\system32\cmd.exe  
-----  
Modbus to DC-Powerupply.  
V-OUT = OFF  
-----  
ENTER = DC-Aux Stop/Run  
-----  
1 - 12 V  
2 - 24 V  
3 - 50 V  
4 - 110 V  
5 - 230 V  
6 - 400 V  
7 - 690 V  
8 - DC-Aux Stop/Run  
9 - Set Voltage (V)  
10 - Set Current (mA)  
0 - EXIT  
Type number then press ENTER:
```