Monitoring a solar installation by tapping into a SunPower PVS6

The following describes how to setup a way to issue http commands to a SunPower PVS6 monitoring system. This information was gather from a number of different places on the Internet, but is presented her in an attempt to provide a single comprehensive document. The most notable sources used were:

- Monitoring a SunPower Solar System (https://blog.gruby.com/2020/04/28/monitoring-a-sunpower-solar-system/)
    - Describes Scott's experience, and configuring a Raspberry Pi3, and how he uses Home Assistant with Grafana to build a dashboard
- SunPower PVS5x/PVS6 Notes (https://github.com/ginoledesma/sunpower-pvs-exporter/blob/master/sunpower_pvs_notes.md)
    - Documents using a Raspberry Pi3 as a bridge to the PVS5x or PVS6 system as well as some of the API commands (with a few errors)
    - The GitHub repository also contains some code
- Sunpower PVS6 Installation Instructions (https://us.sunpower.com/sites/default/files/sunpower-pv-supervisor-6-installation-instructions-531566-reva_0.pdf)
- Commissioning EquinoxTM Systems using WiFi (https://us.sunpower.com/sites/default/files/tech-note-commissioning-equinoxtm-systems-using-wifi-534241-004_0.pdf)
- Equinox Installation Support (https://us.sunpower.com/support/install/equinox)

The basic reason for even wanting to bother with all this is the finding that the SunPower mobile app or web site that is offered to customers is extremely limited in functionality and, basically, only offers insights into current solar production, current consumption (only if a metering kit was installed which often times is not the case, so you would have to ask for it), and some historical graphs showing kWh produced or power output. There is no information

about individual system components regarding their functional state, as well as performance metrics. Additionally it appears the the SunPower data on the app or web site is only updated every 15 minutes.

Using the approach describe here, more detail and data with higher frequency is available. I first describe the basic functioning of the PVS system and how one can "tap" into it, and then discuss the specific solution using a Raspberry Pi3. Finally I document the API that becomes available using this approach. The API can be used by custom code you or somebody else writes. One such example is an integration called SunPower, for Home Assistant (https://github.com/krbaker/hass-sunpower).

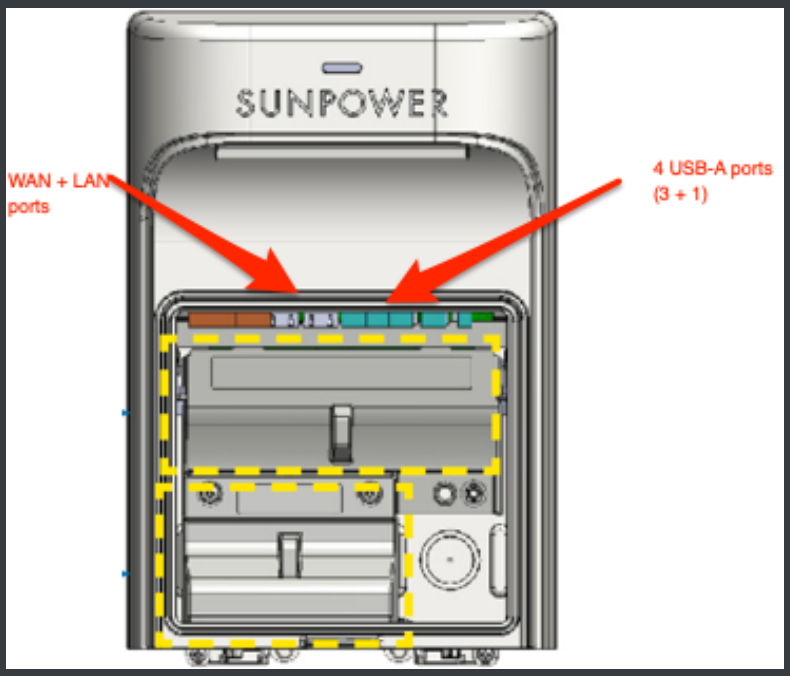
The PVS6 system

What follows may not precisely describe how things are for the PVS5 series. Although it seems very similar with respect to the functionality described here, I don't have one so could not test it. The system is a separate box, typically installed near the (sub)panel where the wiring from the solar array is hooked up. Inside it generally contains one or two circular magnets/coils installed around the PVS wires to the the energy produced can be monitored. There may also be a set around the incoming wires from the utility (a so-called metering kit). The latter is often times omitted, unless you have battery storage, in which case it is needed to properly decide where to divert energy to. The system will be power via its own private breaker circuit.

The system has a number of I/O ports inside (remove front cover). Most notably there are two Ethernet ports labelled WAN/LAN2 and LAN1, and 4 USB-A ports. Also available is a wireless radio (for WiFi access), and a cellular system, which will most of the time not be equipped with a SIM unless you have cellular as your only option to connect to Internet for monitoring by, and uploading to SunPower. Finally there is also a facility to access the PVS monitoring system through PLC (Power Line Communication). The latter is also how the monitoring system communicates with all the micro inverters for the panels. The USB ports do not appear to be used (at least in my case), and probably comes as part of a fairly generic motherboard used inside. They can come in handy though, as you will see later.

In the most basic setup, the PVS monitoring system needs Internet access to communicate with SunPower's cloud. In many customer installations this is achieved by connecting to the customer's WiFi network. Where this is not possible, or not desirable, an Ethernet cable from the WAN port to the customer network is another option (but often a cumbersome one as the panel will likely be outside, so a cable will have to be routed to a suitable location inside). In both cases the PVS monitoring system will act as a DHCP client on the customer network to obtain its IP address, although it is also possible, using installer access via the specialized mobile app, to configure a static ip address. Generally it is advisable to leave things as DHCP but configure your home router or WiFi with a statically reserved DHCP address so the box always gets the same IP address. Failing the above two solutions a PLC based connection is the next option, and finally there is cellular. Only the latter will not make the PVS monitoring system available on the customer network and everything discussed here cannot be used.

While the WAN port is for wired communication between the PVS monitoring system and customer network, the LAN port is meant for the installer of the system. A laptop can be plugged into it directly. To support this mode of operation, the PVS monitoring system runs a DHCP server on this port, so the laptop can obtain a suitable IP address, routing information (the PVS monitoring system is the gateway) and DNS (so that a lookup of www.sunpowerconsole.com will work). The port presents a network with address 172.27.153.0/24 (netmask 255.255.255.0) and the PVS monitoring system will be at the gateway address supplied by DHCP (this seems to always be 172.27.153.1). The laptop will get a suitable address on the same network.


SUNPOWER
WAN + LAN ports
4 USB-A ports (3 + 1)

The black port is LAN1, the yellow port is WAN/LAN2! Raspberry powered from USB port 1. Using short cables, plenty of space.

The laptop tapping into this port is the legacy way of configuring the system. These days installers can use a mobile app available from the iOS app store, and I assume there is also an Android version. The app initially uses BlueTooth to connect. It will obtain the name of a WiFi network presented by the system and subsequently you mobile device will need to be configured to connect to that network to access the needed functionality. While you can download this app yourself, soon after you start it it requires you to login the the SunPower portal with a qualified installer account. An account that they presumably will not give to an

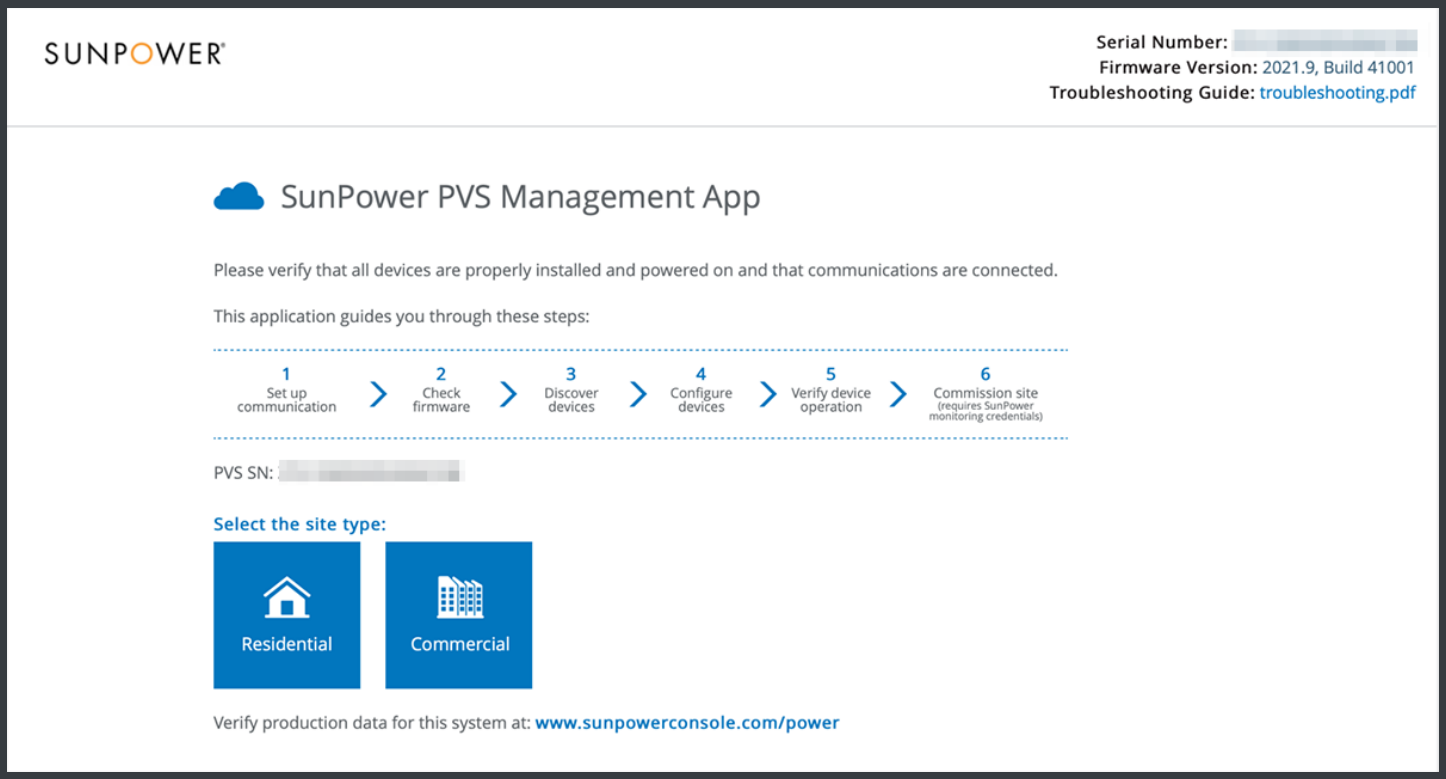
end user.

I also found a https://us.sunpower.com/sites/default/files/tech-note-commissioning-equinoxtm-systems-using-wifi-534241-004_0.pdf (from August 2019) detailing how to connect to the PVS6 using laptop and WiFi. Apparently during commisioning the box provides a WiFi network. Using the serial number found on sticker(s) on the box, numbering its characters from left to right, starting at 1 (we'll use sample `ZT190585000549A6185`):

- Characters 5, 6, followed by last 3 characters appended to "SunPower" will be the SSID to connect to (Example: `SunPower0518)
- Characters 3, 4, 5, 6 followed by last four characters will be the password to use (Example: `19056185`)

This WiFi network is apparently available for four hours after power up unless commisioning is complete. If commisioning is started in that window, it will stay active. If you go beyond the window, you can reboot (power cycle) the PVS6 and a new 4 hour window will start.

The way the LAN port works is that the monitoring system runs a web server, which listens on port 80 (standard http port) and 443 (standard SSL port). If one accesses www.sunpowerconsole.com from a browser on the laptop you will see something like this:



Unless you can see this, something is wrong with your setup! Other urls available:

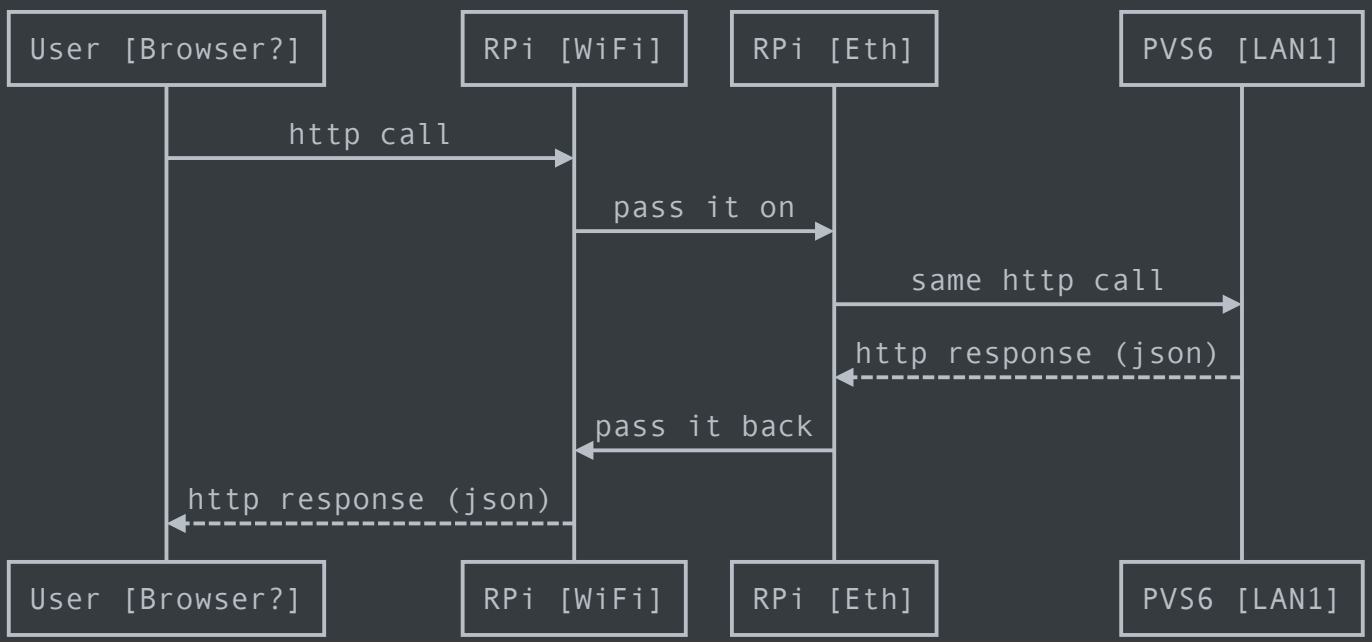
- http://www.sunpowerconsole.com/#/network/config: leads to network setup page
- http://www.sunpowerconsole.com/#/firmware: checks for an applies firmware update
- http://www.sunpowerconsole.com/#/rma-device-selection: allows request equipment RMA
- http://www.sunpowerconsole.com/#/devices/inverter-micro: returns empty results for me, but my system is powered off (waiting for PTO)

Now, you could stand out there, with your laptop, and play with this console or issue API commands, but it is not very practical beyond some basic testing. So, how to proceed? You basically want to create a situation where you can directly access this port from without your own network. If you are connected via WiFi you are, effectively, talking to the WAN port, where neither console nor API is available. So, somehow you must connect to the LAN port. Forgetting the inconvenience (to some) of installing a cable, you can use a cable, but there are some problems you will need to get around. You cannot just plug this cable into a switch on your network as the PVS monitoring system has its own address and runs a DHCP server. You probably already have one, and chaos with networking problems will ensure. It is possible to get around this by using a dedicated port on your router or switch and configuring proper routing from your network to this network (possibly involving VLAN solutions). Since your

computer stays on your network, nobody will be using the PVS supplied DHCP server and you should be fine. You will have to make assumptions about network addresses and setup, and you will not have DNS to resolve sunpowerconsole.com for you. There are workarounds to all of this, but I consider them out of scope for this writeup.

There is another solution which is relatively cheap and easy. Use a very small form factor computer with at least one ethernet port and with WiFi and install it inside, or near the PVS monitoring box. Ethernet connects to the PVS lan port, and WiFi needs to be configured to connect to your WiFi. With some proper configuration that computer can act as a proxy between both networks. The address of that little box on your WiFi becomes, effectively the http access to the PVS system. Any http query sent to the little box is forwarded to the LAN port, and any responses are forwarded back to whomever asked. A very commonly suggested solution involves a Raspberry Pi3. Principally because it is small, cheap, and comes with Ethernet and WiFi. Older model Raspberry can be equipped with a USB based WiFi dongle fairly cheaply too, so if you have one of those lying around it can also be made to work. In the remainder I will describe the Raspberry based solution, but the principles should apply to almost any other small computer.

In this setup the small computer acts as bridge (or proxy) to the PVS6 internal network (RPi = Raspberry, and [] denotes the network interface):



The following ports/interfaces will be in use:

- PVS6 LAN1: Installer/Console port, represented with IP address: `172.27.153.1`
- PVS6 WAN/LAN2: Customer/WAN port using WiFi (IP addressed assigned by your WiFi network and used for uploading to SunPower, not otherwise relevant in this document)
- RPi LAN1: Ethernet port, directly connected too PVS6 LAN1. Will receive an IP address from the PVS6 system in the network `172.27.153.0/24`.
- RPi LAN2/WiFi: Raspberry customer facing interface. This is where you will browse to, or execute API commands. IP address assigned by customer WiFi (static reservation?)

Setting up the Raspberry

For hardware you will need:

- Raspberry Pi3, preferable in a small enclosure. I mentions this model because it has built-in WiFi and an Ethernet port. Other models may well be used for this function as well. For example the Pi Zero 2 W, at US $15, combined with an Ethernet dongle for another US$ 15 or so may well do the job. The Zero 2 W has the same processor as the Pi3, but less memory. I have not actually tested this though.
- A power supply ending in a micro USB-B connector. Usually you buy this with your Raspberry. You can also use a USB-A to micro USB-B cable with the A end plugged into a suitably equipped USB port (must be able to provide enough power, often 700mA is enough). Frankly, for a permanent installation you will need the latter anyway.
- A micro SD card to load the software on. It seems an 8GB card is enough, may be even a 4GB. I used a 32GB and have not specifically tested with the smaller sizes!
- A (short) ethernet cable to connect the Raspberry to the PVS monitoring system

Prepare the software on the Raspberry

1. Download the Raspberry Pi Imager
2. Select the Raspbian Lite image (or any other the you like and prefer, but this is good enough).
3. Write the image to an SD card using the imager.
4. Your resulting SD card wil now be mounted on your system and wil be accessible there.

5. Create a file, at the root level of the SD card (it will be copied into the right place once the Raspberry boots). Call it `wpa_supplicant.conf`, and give it the contents listed below. This will enable the Raspberry to connect to your WiFi.
6. Create an empty file named `ssh` at the root level also. This too will be removed after boot, but causes ssh access to be configured.
7. Eject the SD card from your desktop or laptop
8. Install SD card in Raspberry
9. Connect Raspberry Ethernet port to your network
10. Boot the Raspberry by applying power to it

The `wpa_supplicant.conf` contents:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
 ssid="<Name of your WiFi>"
 psk="<Password for your WiFi>"
}
```

Setup networking

You will need to configure things so that you know what the IP address is for your Raspberry on your network. There will be two addresses, one on the Ethernet cable and one on the WiFi (if you set everything up correctly). The best way is to make so-called static reservations for both connections in your DHCP server (typically on your WiFi router, or other router). To do this you need the MAC addresses of the Ethernet and WiFi.

If you do not have those you may have to figure them out first. There are a variety of options to do so and methods vary a little between Windows and Mac or Linux, although the latter two may be quite similar. I am not going into all detail here, but suggest looking into `arp -a` or using the command `192.168.1.255` or similar depending on your network settings. Once you

have found the appropriate MAC addresses you can make the static DHCP reservations. How depends on your equipment and I won't discuss it here.

If you can find the IP address to either of the two ports, you should be able to login to the Raspberry using the username `raspberry` and password `raspberry`. If you did the SSH step above correctly this should work. Now is a good time to change the default password to something of your liking: `passwd` and follow the prompts.

Reboot the Raspberry to have it find its now reserved IP address. From now on your should be able to use SSH with that address. You might even go one step further and setup your router to "publish" a name for it, for your convenience. This typically involves the DNS configuration of your router. Verify you can connect to either of the two IP addresses.

Setup networking for PVS6

To allow the Raspberry to be the proxy we described above:

1. Login to the Raspberry and update the OS: `sudo apt-get update`
2. Install `ha-proxy`: `sudo apt-get install haproxy` (note: no dash in the name here!)
3. Modify the networking setup for the ethernet port so that it will not configure the gateway address supplied by the PVS6 DHCP server. If you do not do this, or do this incorrectly, the WiFi connection will not work. If that happens, use an ethernet cable to login and fix it. The change requires you edit `/etc/dhcpd.conf` and add two lines to it and save:
    1. `interface eth0`
    2. `nogateway`
4. Next add the content shown below to `/etc/haproxy/haproxy.cfg`
5. Reboot the Raspberry: `sudo shutdown -r now`

Content to add to `/etc/haproxy/haproxy.cfg`:

```
frontend http-in
    bind *:80
    default_backend backend_servers

backend backend_servers
    server sv1 172.27.153.1:80

listen stats
    bind *:8080
    stats enable
    stats uri /
    stats refresh 10s
    stats admin if LOCALHOST
```

Install the Raspberry inside/along the PVS6 system

Open up the PVS monitoring system by releasing the clip at the front-bottom. This will expose the inside where the ports are. For now it will be easiest to hook everything up provisionally and worry about stashing everything away later.

Plug a (short) ethernet cable from the Raspberry into the LAN(1) port. Provide a power source to the Raspberry:

- Use a (short) USB-A to micro B. I tested this with ports 3 and 4, but I think any of this will work. It seems to supply enough power for the Raspberry (at least in the manner it is used here), or
- Use the supplied power brick. This likely won't work well for a permanent installation inside the PVS monitoring box because there is no power outlet there, but for now it can work.

Power up the Raspberry.

Testing

You should now be able to open up a browser on your desktop, laptop, tablet or phone and browse to the WiFi IP address that you setup for your Raspberry. When you do, use `http` and not `https` as we have not configured that. You should be seeing the SunPower console landing page shown earlier. If you don't, something is wrong with your setup. Retrace all steps to troubleshoot.

Now that, presumably, the Raspberry proxy setup works you can use it to try and use the console (I did not bother so I don't know if you could do everything an installer does, or whether passwords are needed). You can also access the built-in API for requesting information.

You can try this out by browsing to:

```
http://<your pi3 IP address on WiFi>/cgi-bin/dl_cgi?Command=DeviceList
```

That will produce some output that starts like this (or looks like it with some different numbers in it):


```
{
    "devices":  [{
            "DETAIL":   "detail",
            "STATE":    "working",
            "STATEDESCR":   "Working",
            "SERIAL":   "ZT01234567890ABCDEF",
            "MODEL":    "PV Supervisor PVS6",
            "HWVER":    "6.02",
            "SWVER":    "2021.9, Build 41001",
            "DEVICE_TYPE":  "PVS",
            "DATATIME": "2021,10,28,07,31,48",
            "dl_err_count": "0",
            "dl_comm_err":  "0",
```

```
            "dl_skipped_scans": "0",
            "dl_scan_time": "0",
            "dl_untransmitted": "6108",
            "dl_uptime":    "34",
            "dl_cpu_load":  "0.87",
            "dl_mem_used":  "33352",
            "dl_flash_avail":   "67546",
            "panid":    1234567890,
            "CURTIME":  "2021,10,28,07,33,52"
        }, {
...
```

There is a lot more there, but not shown. I used fake serial number and “panid”! Seeing this output (it may take a few seconds) again confirms where access to detailed data.

A browser is not necessarily the best vehicle to interact with this API. For simple testing it works and you can study the output. Below you will find documentation for everything I have found. I use a tool called “Paw” for Mac to make interaction and experimentation easier, but you can also use `curl`, or `wget` or us a programming language of your choice (Python seems quite popular for this).

Confirming the above works may be enough for you. If, for example, you now wish to integrate your SunPower information with Home Assistant you can do so.

Instructions for cabled connection

These instructions cannot be precise and complete without knowing exactly what equipment you are using. If you are going this route you are probably fairly conversant in networking and using these “hints” can figure it out.

Using an (old) intermediary router

Basically follow these steps:

- Connect router WAN port to PVS6 LAN1 (installer) port by cable
  - Configure with a static address of `172.27.153.2` and netmask `255.255.255.0`
  - Configure with gateway `172.27.153.1`
  - Configure with DNS server 1 `172.27.153.1` and optional server 2 something like `8.8.8.8` (Google DNS), or `1.1.1.1` (Cloudflare DNS)
- Connect router LAN port to your regular network
  - Configure with available static address on your local LAN, or use your configure as dynamic (DHCP) and configure your regular network's DHCP server with an appropriate static reservation
- Turn of DHCP and possibly WiFi on old router. You won't be using/needing these.
- On the old router setup a "static route" for `192.168.1.0/24` to its LAN port (as that is where your regular network is). NOTE: I used the commonly used `192.168.1.0/24` as an example here. Substitute with your correct network. This route will tell it how to route responses to requests coming from that network.
- In your regular router setup a "static route" for `172.27.153.0/24` to its WAN port (as that is where the PVS6 connection now is). This route will tell it how to send packets for the PVS6 network to the old router which will then send it to the PVS6.
- Setup your DNS to map `sunpowerconsole.com` to `172.27.153.1`

You should now be able to browse to: `http://sunpowerconsole.com` as described.

Using your regular router with extra port

If you don't have an additional router, but your regular router has an extra port you can likely configure it as described for the "old" router above (Turn off DHCP for that port only). This is not typicaly an option with provider supplied routers. The ports on those are just part of a switch and cannot be managed. I use an EdgeRouter POE which is fully manageable and could do this (but I am using the Raspberry approach).

Integrating with Home Assistant

Assuming you already have a HA instance up and running, integration is pretty straightforward. Make sure https://hacs.xyz (Home Assistant Community Store) is installed an operational. Configure the repository from "https://github.com/krbaker" with URL: https://github.com/krbaker/hass-sunpower.

Follow the steps to install, and restart HA. You can now go to "Configuration > Integrations" and click "+" in the bottom right corner and search for "SunPower". You will then be asked for a host IP address. You need to input the address of your Raspberry's WiFi connection, unless you are using a more complex network setup, where you would directly use the SunPower console IP address of `172.27.153.1`.

After this is all done, you should see your PVS monitoring system and associated devices for the supervisor, power production meter, power consumption meter and your micro inverters. Information for each individual micro inverter is available. Not that their state might be listed as "error" if your PV system is not yet switched on (breaker(s) in the off position). This might be the case between completion of installation and your utility's permission to operate is granted.

Other integrations

I have not personally investigated other integrations. If you need something that works out of the box, you'll have to Google around and give things a try. If you are willing and able to do a little programming, look at the resources mentioned at the start of this writeup. Several of them have GitHub repositories with code in them. At the very least it will show you how it is done, or you can modify the code to suit your needs.

API information

The Sunpower Console API recognizes a bunch of specific commands. The commands are issued by executing a `GET` request over `http` to the URL that looks like this:

```
http://<rpi address>/cgi-bin/dl_cgi?Command=<command name>
```

Below I document each command I know about. Some commands just require their name as the `Command` query parameter, some require an addition of your system's serial number using `SerialNumber`. See documentation. In the DeviceList` command I will show full response headers and body, but in subsequent documentation I will only show the body and explanations of what you see there.

The headers seem to indicate that not only GET commands are allowed, but also POST, DELETE and PUT. This would suggest this is a full so-called REST API. I have not attempted to do any of those interactions. If they even work, there is potential for messing something up. Since my system is not for experimentation and for power production I do not want to take that risk. Be warned!

You will note that several commands return data with some keys listed in all uppercase and others in all lowercase. It appears the UPPERCASE ones are device properties/attributes and the lowercase ones are performance indicators.

GET /cgi-bin/dl_cgi?Command=DeviceList

This command gets a complete list of all known PVS devices and their status and performance descriptors. There seem to be three device types:

1. Supervisor. This seems to be the main entity that controls everything
2. Power meter, which comes in a production and consumption version. The latter will show up even if no consumption metering kit is installed, but kWh information should stay at 0.
3. Inverter. There may be as many of these as you have micro-inverters (panels).

The output body below has fake serial numbers, and has all but the first inverter omitted. This way you can see an example of each device type in the output. Detail is discussed below the body.

You will also note the `result` part of the response. I have never seen anything but `success` but it stands to reason that if it say anything but `success` the rest of the response should not be interpreted.

> **NOTE**: This command may take quite a while. On my PVS6 system with 24 panels it seems to take between 6 and 7 seconds to return all results!

- Response 200 (application/json)
  - Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
Access-Control-Allow-Methods: GET,POST,DELETE,PUT
Pragma: no-cache
dlcgi-build-time: Sep 15 2021 00:05:00
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Cache-Control: no-cache, no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Server: lighttpd/1.4.51
```

  - Body


```
{
    "devices":  [{
            "DETAIL":   "detail",
            "STATE":    "working",
            "STATEDESCR":   "Working",
            "SERIAL":   "ZT01234567890ABCDEF",
            "MODEL":    "PV Supervisor PVS6",
            "HWVER":    "6.02",
            "SWVER":    "2021.9, Build 41001",
            "DEVICE_TYPE":  "PVS",
            "DATATIME": "2021,10,27,07,25,00",
            "dl_err_count": "0",
            "dl_comm_err":  "480",
            "dl_skipped_scans": "0",
            "dl_scan_time": "1",
```

```
            "dl_untransmitted": "5539",
            "dl_uptime":    "85428",
            "dl_cpu_load":  "0.28",
            "dl_mem_used":  "73704",
            "dl_flash_avail":   "67599",
            "panid":    1234567890,
            "CURTIME":  "2021,10,27,07,28,13"
        }, {
            "ISDETAIL": true,
            "SERIAL":   "PVS6M12345678p",
            "TYPE": "PVS5-METER-P",
            "STATE":    "working",
            "STATEDESCR":   "Working",
            "MODEL":    "PVS6M0400p",
            "DESCR":    "Power Meter PVS6M12345678p",
            "DEVICE_TYPE":  "Power Meter",
            "SWVER":    "3000",
            "PORT": "",
            "DATATIME": "2021,10,27,07,28,12",
            "ct_scl_fctr":  "50",
            "net_ltea_3phsum_kwh":  "11.34",
            "p_3phsum_kw":  "0",
            "q_3phsum_kvar":    "0",
            "s_3phsum_kva": "0",
            "tot_pf_rto":   "1",
            "freq_hz":  "60",
            "CAL0": "50",
            "origin":   "data_logger",
            "OPERATION":    "noop",
            "CURTIME":  "2021,10,27,07,28,13"
        }, {
            "ISDETAIL": true,
            "SERIAL":   "PVS6M12345678c",
            "TYPE": "PVS5-METER-C",
```

```
            "STATE":    "working",
            "STATEDESCR":   "Working",
            "MODEL":    "PVS6M0400c",
            "DESCR":    "Power Meter PVS6M12345678c",
            "DEVICE_TYPE":  "Power Meter",
            "SWVER":    "3000",
            "PORT": "",
            "DATATIME": "2021,10,27,07,28,13",
            "ct_scl_fctr":  "100",
            "net_ltea_3phsum_kwh":  "0",
            "p_3phsum_kw":  "0",
            "q_3phsum_kvar":    "0",
            "s_3phsum_kva": "0",
            "tot_pf_rto":   "0",
            "freq_hz":  "60",
            "i1_a": "0",
            "i2_a": "0",
            "v1n_v":    "116.5863",
            "v2n_v":    "116.4296",
            "v12_v":    "233.0139",
            "p1_kw":    "0",
            "p2_kw":    "0",
            "neg_ltea_3phsum_kwh":  "0",
            "pos_ltea_3phsum_kwh":  "0",
            "CAL0": "100",
            "origin":   "data_logger",
            "OPERATION":    "noop",
            "CURTIME":  "2021,10,27,07,28,14"
        }, {
            "ISDETAIL": true,
            "SERIAL":   "E00123456000001",
            "TYPE": "SOLARBRIDGE",
            "STATE":    "error",
            "STATEDESCR":   "Error",
```

```
                "MODEL":    "AC_Module_Type_E",
                "DESCR":    "Inverter E00122125092032",
                "DEVICE_TYPE":  "Inverter",
                "PANEL":    "SPR-X22-360-E-AC",
                "SWVER":    "4.21.3",
                "PORT": "",
                "MOD_SN":   "R36M20579663",
                "NMPLT_SKU":    "",
                "origin":   "data_logger",
                "OPERATION":    "noop",
                "CURTIME":  "2021,10,28,09,03,43"
            }],
        "result":   "succeed"
    }
```


Detail for supervisor

Most of the properties are self explanatory. The `DATATIME` indicates the timestamp when the presented information last changed, whereas `CURTIME` should be the timestamp of the request (or something close to it). The timestamps are in UTC! Observation of `DATATIME` on my system shows that the minutes component is always a multiple of 5 minutes and seconds is always 0. So if you are interested getting timely information, perform the call just after that start of each 5 minute interval. You can interrogate more often, but if timing is not important, don't stress the PVS6 by constantly asking for `DeviceList`.

The meaning of the performance indicators can only be guessed by observation and a little knowledge (dl probably means "data logger"):

- `dl_err_count` : Number of errors since last report?
- `dl_comm_err` : Number of communication errors? (Not sure what kind)
- `dl_skipped_scans` : Best guess this counts when the supervisor scans the PLC network for inverters and it decides it needs to skip a scan
- `dl_scan_time` : Probably related somehow to the above mentioned scan

- `dl_untransmitted` : Number of not yet transmitted (to SunPower) events/records?
- `dl_uptime` : number of seconds the data logger has been running. I observed it being lower than a previously seen value (prior value was almost 24h, new value was a little over 1.5h). I was not aware of a power outage in between so either there was a restart for power reasons or other, or perhaps the device restarts itself periodically (somewhat lousy safegurard against memory leaks), or something else. Also see remark below.
- `dl_cpu_load` : CPU load average as it existed (possibly averaged over 5 minutes) at `DATATIME`
- `dl_mem_used` : amount of memory in use. Assuming the motherboard has little memory this is probably in kiB
- `dl_flash_avail` : amount of free space on flash device. Assumed in kIB

During experimentation I occasionally encountered this response:

```
HTTP/1.0 503 Service Unavailable
Cache-Control: no-cache
Connection: close
Content-Type: text/html

<html><body><h1>503 Service Unavailable</h1>
No server is available to handle this request.
</body></html>
```

That would stay that wasy for a few minutes. After a successful retry I would see `dl_uptime` be a very low number, indicating that the supervisor/datalogger had apparently restarted. It is not clear if I cause the crash/restart, or it “just happens”.

Detail for power meter

Reports are that powermeter data is updated more frequently than the supervisor information. Reports indicate at least every 5 eceonds. The power meter’s properties are fairly self explanatory. One thing to note is that the `DESCR` property contains the meter’s serial number inside its name. The name ends with the letter “p”, indicating this is the “production” meter, or

the letter “c” indicating the consumer meter. Ditto for the `MODEL` property. Also present, as described for the supervisor are `DATATIME` and `CURTIME`.

The `CAL0` property indicates the sensor current capacity for the calibration-reference CT sensor. It will be “50” (50A) fr the production meter, and either “100” or “200” for consumption. Note that the `ct_scl_fctr` contains the capacity for the actual sensor.

The meaning of the performance indicators can only be guessed by observation and a little knowledge:

- `ct_scl_fctr` : Current capacity for the actual CT sensor used
- `net_ltea_3phsum_kwh` : Net cumulative energy, in kWh, across all three phases
- `p_3phsum_kw` : Average real power
- `q_3phsum_kvar` : Cumulative “reactive” power, in kVA or kW, across all three phases (since ?)
- `s_3phsum_kva` : Cumulative “apparent” power, in kVA or kW, across all three phases (since ?)
- `tot_pf_rto` : Power factor ratio, defined as `real power` divided by `apparant power`
- `freq_hz` : Operating frequency (typically around 60 Hz in the US)

The above performance indicators are present in both production and consumer meters. The following is additionally available in the consumption meter:

- `i1_a` : Supply current, in A, on CT1 lead
- `i2_a` : Supply current, in A, on CT2 lead
- `v1n_v` : Supply voltage CT1 lead (relative to neutral, typically in the 110-120 V range)
- `v2n_v` : Supply voltage CT2 lead (relative to neutral, typically in the 110-120 V range)
- `v12_v` : Supply Voltage sum across CT1 and CT2 leads (typically in the 220-140 V range)
- `p1_kw` : Lead 1 average power in kW. Can be positive (excess back to utility) or negative (used from utility)
- `p2_kw` : Lead 2 average power in kW. Can be positive (excess back to utility) or negative (used from utility)
- `neg_ltea_3phsum_kwh` : Cumulative energy, in kWh, across all three phases, consumed

from utility
- `pos_ltea_3phsum_kwh` :Cumulative energy, in kWh, across all three phases, supplied to utility

Note that “net” utility consumption can be computed by subtracting the `ltea_3phsum_kwh` numbers.

The “production” meter only measures power produced by the PV array. At night, when there is no production, this number can go a tiny bit negative. This is the energy consumed by the PV electronics (inverters). It has been reported in the -15 W range, but that may be specific for a certain number of panels and model micro inverters. Since this power is always needed, that would mean that on a daily basis 24 x 15 Wh = 0.36 kWh from production is never available for supplying the home or back to the utility. If you insist you can try to factor this into any calculations, but I am going to suggest it is too low to worry about. In a 6 kW system this would represent 15 / 6000 = 0.25%.

Detail for micro inverter

The information for the inverter(s) has the usual UPPERCASE properties. It should be noted that when the panels are not powered, the returned `STATE` will be error. This presumes at some point the power was on (typically during initial install and commisioning) and the actual devices were discovered.

The following performance data seems to be available (but is not included where `STATE` is not “working”):

- `freq_hz` : Operating Frequency in Hz (typically around 60 Hz in the US)
- `i_3phsum_a` : AC Current (in A)
- `i_mppt1_a` : DC Current (in A)
- `ltea_3phsum_kwh` : Total energy (in kWh)
- `p_3phsum_kw` : AC Power (in kW)
- `p_mpptsum_kw` : DC Power (in kW)
- `t_htsnk_degc` : Heatsink temperature (degrees Celsius)
- `v_mppt1_v` : DC Voltage (in V)

- `vln_3phavg_v` : AC Voltage (in V)

GET /cgi-bin/dl_cgi?Command=Get_Comm&SerialNumber=ZT01234567890ABCDEF

This command requires the addition of the serial number. Possibly if there are multiple supervisors on the internal SunPower network it can select the correct one. The command returns information about the available network interfaces.

- Body


```
{
    "result":   "succeed",
    "networkstatus":    {
        "interfaces":   [{
                "interface":    "wan",
                "internet": "down",
                "ipaddr":   "",
                "link": "disconnected",
                "mode": "wan",
                "sms":  "unreachable",
                "state":    "down"
            }, {
                "interface":    "plc",
                "internet": "down",
                "ipaddr":   "",
                "link": "disconnected",
                "pairing":  "unpaired",
                "sms":  "unreachable",
                "speed":    0,
                "state":    "down"
            }, {
                "interface":    "sta0",
                "internet": "up",
```

```
                "ipaddr":   "192.168.10.239",
                "signal":   "-48",
                "sms":  "reachable",
                "ssid": "Starfield",
                "status":   "connected"
            }, {
                "interface":    "cell",
                "internet": "down",
                "ipaddr":   "",
                "is_alwayson":  false,
                "is_primary":   false,
                "link": "connected",
                "modem":    "MODEM_OK",
                "provider": "UNKNOWN",
                "signal":   0,
                "sim":  "SIM_READY",
                "sms":  "unreachable",
                "state":    "DOWN",
                "status":   "NOT_REGISTERED"
            }],
        "system":   {
            "interface":    "sta0",
            "internet": "up",
            "sms":  "reachable"
        },
        "ts":   "1635315583"
    }
}
```

GET /cgi-bin/dl_cgi?Command=Start

This command starts a configuration session. I am not sure what that means, but I am guessing it is related to the ability to use this as a REST API where any changes made using PUT, POST and DELETE cannot be done until a session is started version, and are not made permanent until (a) the result is a valid configuration, and (b) the session is “stopped” (see below).

- Body


```
{
    "result":   "succeed",
    "supervisor":   {
        "SWVER":    "2021.9, Build 41001",
        "SERIAL":   "ZT01234567890ABCDEF",
        "MODEL":    "PVS6",
        "BUILD":    41001,
        "FWVER":    "1.0.0",
        "SCVER":    16504,
        "EASICVER": 131329,
        "SCBUILD":  1185,
        "WNSERIAL": 16,
        "WNMODEL":  400,
        "WNVER":    3000
    }
}
```


GET /cgi-bin/dl_cgi?Command=Stop

This command stops the configuration session. Not (yet) clean what that is.

- Body

```
{
    "result":   "succeed"
}
```

GET /cgi-bin/dl_cgi?Command=CheckFW

This command checks if a firmware update is available. If not, it retirns “none”, otherwise a url for the firmware file.

- Body

```
{
    "url":  "none"
}
```

GET /cgi-bin/dl_cgi?Command=DeviceDetails&SerialNumber=ZT01234567890ABCDEF

I found this command mentioned somewhere, but I have not gotten it to work

- Body

```
{   "result":   "unknown command"}
```

GET /cgi-bin/dl_cgi?Command=GridProfileGet

This command exposes the currently selected/active grid profile (for a full list see the `GridProfileRefresh` command).

- Body

```
{
    "result":   "succeed",
    "active_name":  "IEEE-1547a-2014 + 2020 CA Rule21",
    "active_id":    "816bf3302d337a42680b996227ddbc46abf9cd05",
    "pending_name": "IEEE-1547a-2014 + 2020 CA Rule21",
    "pending_id":   "816bf3302d337a42680b996227ddbc46abf9cd05",
    "percent":  100,
    "supported_by": "ALL",
    "status":   "success"
}
```


GET /cgi-bin/dl_cgi?Command=GridProfileRefresh

This command seems to “refresh” in internal database of supply grid profiles. The profiles have a unique `id`, a descriptive `name`, refer to a file with meta data about the profile (`filename`) and have a list of zip codes (`zipcodes`) to which it (may) apply.

- Body


```
{
    "result":   "succeed",
    "success":  true,
    "creation": 1600704253,
    "profiles": [{
            "selfsupply":   true,
            "zipcodes": [{
                    "max":  96898,
                    "min":  96701
                }],
            "default":  false,
            "filename": "8c9c4170.meta",
            "id":   "8c9c4170457c88f6dcee7216357681d580a3b9bd",
            "name": "HECO OMH R14H (Legacy)"
```

```
        }, {
            "selfsupply":   true,
            "zipcodes": [{
                    "max":  96898,
                    "min":  96701
                }],
            "default":  false,
            "filename": "f169fe0d.meta",
            "id":   "f169fe0dc993987c5f105df9d651a284e8031c5a",
            "name": "HI GSUI OMH R14H 2018"
        }, {
            "selfsupply":   true,
            "zipcodes": [{
                    "max":  96898,
                    "min":  96701
                }],
            "default":  false,
            "filename": "2d8a06aa.meta",
            "id":   "2d8a06aa4812913510c7955cabaf114e5432fe67",
            "name": "HI GSUI OMH R14H 2018 + V/W"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  96162,
                    "min":  90001
                }],
            "default":  false,
            "filename": "816bf330.meta",
            "id":   "816bf3302d337a42680b996227ddbc46abf9cd05",
            "name": "IEEE-1547a-2014 + 2020 CA Rule21"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  -1,
```

```
                    "min":  -1
                }],
            "default":  false,
            "filename": "db54e057.meta",
            "id":   "db54e057f901db5d7907af561688f0af555c563d",
            "name": "CA CPUC R21 Reactive Power Priority"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  -1,
                    "min":  -1
                }],
            "default":  false,
            "filename": "43daefaa.meta",
            "id":   "43daefaa4404f4674edd2ad89c175a65f3cebc51",
            "name": "California CPUC Rule21"
        }, {
            "selfsupply":   false,
            "zipcodes": [81121, 81122, 81128, 81130, 81137, 81147,
81157, 81235, 81301, 81302, 81303, 81326, 81328, 81329],
            "default":  false,
            "filename": "954a2a24.meta",
            "id":   "954a2a240698d9c006afb72fc93febc622b296eb",
            "name": "La Plata Electric Assn"
        }, {
            "selfsupply":   false,
```

```
            "zipcodes": [17302, 17309, 17314, 17527, 18014, 18054,
    18073, 18074, 18076, 18084, 18901, 18902, 18910, 18912, 18913, 18914,
    18915, 18916, 18917, 18922, 18923, 18925, 18926, 18928, 18929, 18931,
    18932, 18933, 18934, 18936, 18938, 18940, 18942, 18943, 18944, 18946,
    18947, 18949, 18950, 18951, 18954, 18956, 18957, 18958, 18963, 18964,
    18966, 18969, 18971, 18974, 18976, 18977, 18979, 18980, 18991, 19001,
    19002, 19003, 19004, 19006, 19007, 19008, 19009, 19010, 19012, 19013,
    19014, 19015, 19016, 19017, 19018, 19019, 19020, 19021, 19022, 19023,
    19025, 19026, 19027, 19028, 19029, 19030, 19031, 19032, 19033, 19034,
    19035, 19036, 19037, 19038, 19039, 19040, 19041, 19043, 19044, 19046,
    19047, 19048, 19049, 19050, 19052, 19053, 19054, 19055, 19056, 19057,
    19058, 19060, 19061, 19063, 19064, 19065, 19066, 19067, 19070, 19072,
    19073, 19074, 19075, 19076, 19078, 19079, 19080, 19081, 19082, 19083,
    19085, 19086, 19087, 19088, 19089, 19090, 19091, 19092, 19093, 19094,
    19095, 19096, 19098, 19099, 19101, 19102, 19103, 19104, 19105, 19106,
    19107, 19108, 19109, 19110, 19111, 19112, 19113, 19114, 19115, 19116,
    19118, 19119, 19120, 19121, 19122, 19123, 19124, 19125, 19126, 19127,
    19128, 19129, 19130, 19131, 19132, 19133, 19134, 19135, 19136, 19137,
    19138, 19139, 19140, 19141, 19142, 19143, 19144, 19145, 19146, 19147,
    19148, 19149, 19150, 19151, 19152, 19153, 19154, 19155, 19160, 19161,
    19162, 19170, 19171, 19172, 19173, 19175, 19176, 19177, 19178, 19179,
    19181, 19182, 19183, 19184, 19185, 19187, 19188, 19190, 19191, 19192,
    19193, 19194, 19195, 19196, 19197, 19244, 19255, 19301, 19310, 19311,
    19312, 19316, 19317, 19318, 19319, 19320, 19330, 19331, 19333, 19335,
    19339, 19340, 19341, 19342, 19343, 19344, 19345, 19346, 19347, 19348,
    19350, 19351, 19352, 19353, 19354, 19355, 19357, 19358, 19360, 19362,
    19363, 19365, 19366, 19367, 19369, 19372, 19373, 19374, 19375, 19376,
    19380, 19381, 19382, 19383, 19388, 19390, 19395, 19397, 19398, 19399,
    19401, 19403, 19404, 19405, 19406, 19407, 19408, 19409, 19415, 19420,
    19421, 19422, 19423, 19424, 19425, 19426, 19428, 19429, 19430, 19432,
    19436, 19437, 19438, 19440, 19441, 19442, 19443, 19444, 19446, 19450,
    19451, 19453, 19454, 19455, 19456, 19457, 19460, 19462, 19464, 19465,
    19468, 19473, 19474, 19475, 19477, 19478, 19480, 19481, 19482, 19484,
    19485, 19486, 19490, 19492, 19493, 19494, 19495, 19496, 19520, 19525],
```

```
        "default":  false,
        "filename": "8b82ccd2.meta",
        "id":   "8b82ccd2cfe07c8cc49134fcdcd5edec8845cd5d",
        "name": "PECO Voltage Trip High (254V)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [17302, 17309, 17314, 17527, 18014, 18054,
    18073, 18074, 18076, 18084, 18901, 18902, 18910, 18912, 18913, 18914,
    18915, 18916, 18917, 18922, 18923, 18925, 18926, 18928, 18929, 18931,
    18932, 18933, 18934, 18936, 18938, 18940, 18942, 18943, 18944, 18946,
    18947, 18949, 18950, 18951, 18954, 18956, 18957, 18958, 18963, 18964,
    18966, 18969, 18971, 18974, 18976, 18977, 18979, 18980, 18991, 19001,
    19002, 19003, 19004, 19006, 19007, 19008, 19009, 19010, 19012, 19013,
    19014, 19015, 19016, 19017, 19018, 19019, 19020, 19021, 19022, 19023,
    19025, 19026, 19027, 19028, 19029, 19030, 19031, 19032, 19033, 19034,
    19035, 19036, 19037, 19038, 19039, 19040, 19041, 19043, 19044, 19046,
    19047, 19048, 19049, 19050, 19052, 19053, 19054, 19055, 19056, 19057,
    19058, 19060, 19061, 19063, 19064, 19065, 19066, 19067, 19070, 19072,
    19073, 19074, 19075, 19076, 19078, 19079, 19080, 19081, 19082, 19083,
    19085, 19086, 19087, 19088, 19089, 19090, 19091, 19092, 19093, 19094,
    19095, 19096, 19098, 19099, 19101, 19102, 19103, 19104, 19105, 19106,
    19107, 19108, 19109, 19110, 19111, 19112, 19113, 19114, 19115, 19116,
    19118, 19119, 19120, 19121, 19122, 19123, 19124, 19125, 19126, 19127,
    19128, 19129, 19130, 19131, 19132, 19133, 19134, 19135, 19136, 19137,
    19138, 19139, 19140, 19141, 19142, 19143, 19144, 19145, 19146, 19147,
    19148, 19149, 19150, 19151, 19152, 19153, 19154, 19155, 19160, 19161,
    19162, 19170, 19171, 19172, 19173, 19175, 19176, 19177, 19178, 19179,
    19181, 19182, 19183, 19184, 19185, 19187, 19188, 19190, 19191, 19192,
    19193, 19194, 19195, 19196, 19197, 19244, 19255, 19301, 19310, 19311,
    19312, 19316, 19317, 19318, 19319, 19320, 19330, 19331, 19333, 19335,
    19339, 19340, 19341, 19342, 19343, 19344, 19345, 19346, 19347, 19348,
    19350, 19351, 19352, 19353, 19354, 19355, 19357, 19358, 19360, 19362,
    19363, 19365, 19366, 19367, 19369, 19372, 19373, 19374, 19375, 19376,
    19380, 19381, 19382, 19383, 19388, 19390, 19395, 19397, 19398, 19399,
    19401, 19403, 19404, 19405, 19406, 19407, 19408, 19409, 19415, 19420,
    19421, 19422, 19423, 19424, 19425, 19426, 19428, 19429, 19430, 19432,
    19436, 19437, 19438, 19440, 19441, 19442, 19443, 19444, 19446, 19450,
    19451, 19453, 19454, 19455, 19456, 19457, 19460, 19462, 19464, 19465,
    19468, 19473, 19474, 19475, 19477, 19478, 19480, 19481, 19482, 19484,
    19485, 19486, 19490, 19492, 19493, 19494, 19495, 19496, 19520, 19525],
```

```
        "default":  false,
        "filename": "65edfb4c.meta",
        "id":   "65edfb4c22bc591b1556f910b6d5097dffb30308",
        "name": "PECO Voltage Trip High (255V)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [17302, 17309, 17314, 17527, 18014, 18054,
    18073, 18074, 18076, 18084, 18901, 18902, 18910, 18912, 18913, 18914,
    18915, 18916, 18917, 18922, 18923, 18925, 18926, 18928, 18929, 18931,
    18932, 18933, 18934, 18936, 18938, 18940, 18942, 18943, 18944, 18946,
    18947, 18949, 18950, 18951, 18954, 18956, 18957, 18958, 18963, 18964,
    18966, 18969, 18971, 18974, 18976, 18977, 18979, 18980, 18991, 19001,
    19002, 19003, 19004, 19006, 19007, 19008, 19009, 19010, 19012, 19013,
    19014, 19015, 19016, 19017, 19018, 19019, 19020, 19021, 19022, 19023,
    19025, 19026, 19027, 19028, 19029, 19030, 19031, 19032, 19033, 19034,
    19035, 19036, 19037, 19038, 19039, 19040, 19041, 19043, 19044, 19046,
    19047, 19048, 19049, 19050, 19052, 19053, 19054, 19055, 19056, 19057,
    19058, 19060, 19061, 19063, 19064, 19065, 19066, 19067, 19070, 19072,
    19073, 19074, 19075, 19076, 19078, 19079, 19080, 19081, 19082, 19083,
    19085, 19086, 19087, 19088, 19089, 19090, 19091, 19092, 19093, 19094,
    19095, 19096, 19098, 19099, 19101, 19102, 19103, 19104, 19105, 19106,
    19107, 19108, 19109, 19110, 19111, 19112, 19113, 19114, 19115, 19116,
    19118, 19119, 19120, 19121, 19122, 19123, 19124, 19125, 19126, 19127,
    19128, 19129, 19130, 19131, 19132, 19133, 19134, 19135, 19136, 19137,
    19138, 19139, 19140, 19141, 19142, 19143, 19144, 19145, 19146, 19147,
    19148, 19149, 19150, 19151, 19152, 19153, 19154, 19155, 19160, 19161,
    19162, 19170, 19171, 19172, 19173, 19175, 19176, 19177, 19178, 19179,
    19181, 19182, 19183, 19184, 19185, 19187, 19188, 19190, 19191, 19192,
    19193, 19194, 19195, 19196, 19197, 19244, 19255, 19301, 19310, 19311,
    19312, 19316, 19317, 19318, 19319, 19320, 19330, 19331, 19333, 19335,
    19339, 19340, 19341, 19342, 19343, 19344, 19345, 19346, 19347, 19348,
    19350, 19351, 19352, 19353, 19354, 19355, 19357, 19358, 19360, 19362,
    19363, 19365, 19366, 19367, 19369, 19372, 19373, 19374, 19375, 19376,
    19380, 19381, 19382, 19383, 19388, 19390, 19395, 19397, 19398, 19399,
    19401, 19403, 19404, 19405, 19406, 19407, 19408, 19409, 19415, 19420,
    19421, 19422, 19423, 19424, 19425, 19426, 19428, 19429, 19430, 19432,
    19436, 19437, 19438, 19440, 19441, 19442, 19443, 19444, 19446, 19450,
    19451, 19453, 19454, 19455, 19456, 19457, 19460, 19462, 19464, 19465,
    19468, 19473, 19474, 19475, 19477, 19478, 19480, 19481, 19482, 19484,
    19485, 19486, 19490, 19492, 19493, 19494, 19495, 19496, 19520, 19525],
```

```
        "default":  false,
        "filename": "bec6f3ba.meta",
        "id":   "bec6f3ba13036fea5b7a81b6344308bf40eeaca1",
        "name": "PECO Voltage Trip High (260V)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [17302, 17309, 17314, 17527, 18014, 18054,
    18073, 18074, 18076, 18084, 18901, 18902, 18910, 18912, 18913, 18914,
    18915, 18916, 18917, 18922, 18923, 18925, 18926, 18928, 18929, 18931,
    18932, 18933, 18934, 18936, 18938, 18940, 18942, 18943, 18944, 18946,
    18947, 18949, 18950, 18951, 18954, 18956, 18957, 18958, 18963, 18964,
    18966, 18969, 18971, 18974, 18976, 18977, 18979, 18980, 18991, 19001,
    19002, 19003, 19004, 19006, 19007, 19008, 19009, 19010, 19012, 19013,
    19014, 19015, 19016, 19017, 19018, 19019, 19020, 19021, 19022, 19023,
    19025, 19026, 19027, 19028, 19029, 19030, 19031, 19032, 19033, 19034,
    19035, 19036, 19037, 19038, 19039, 19040, 19041, 19043, 19044, 19046,
    19047, 19048, 19049, 19050, 19052, 19053, 19054, 19055, 19056, 19057,
    19058, 19060, 19061, 19063, 19064, 19065, 19066, 19067, 19070, 19072,
    19073, 19074, 19075, 19076, 19078, 19079, 19080, 19081, 19082, 19083,
    19085, 19086, 19087, 19088, 19089, 19090, 19091, 19092, 19093, 19094,
    19095, 19096, 19098, 19099, 19101, 19102, 19103, 19104, 19105, 19106,
    19107, 19108, 19109, 19110, 19111, 19112, 19113, 19114, 19115, 19116,
    19118, 19119, 19120, 19121, 19122, 19123, 19124, 19125, 19126, 19127,
    19128, 19129, 19130, 19131, 19132, 19133, 19134, 19135, 19136, 19137,
    19138, 19139, 19140, 19141, 19142, 19143, 19144, 19145, 19146, 19147,
    19148, 19149, 19150, 19151, 19152, 19153, 19154, 19155, 19160, 19161,
    19162, 19170, 19171, 19172, 19173, 19175, 19176, 19177, 19178, 19179,
    19181, 19182, 19183, 19184, 19185, 19187, 19188, 19190, 19191, 19192,
    19193, 19194, 19195, 19196, 19197, 19244, 19255, 19301, 19310, 19311,
    19312, 19316, 19317, 19318, 19319, 19320, 19330, 19331, 19333, 19335,
    19339, 19340, 19341, 19342, 19343, 19344, 19345, 19346, 19347, 19348,
    19350, 19351, 19352, 19353, 19354, 19355, 19357, 19358, 19360, 19362,
    19363, 19365, 19366, 19367, 19369, 19372, 19373, 19374, 19375, 19376,
    19380, 19381, 19382, 19383, 19388, 19390, 19395, 19397, 19398, 19399,
    19401, 19403, 19404, 19405, 19406, 19407, 19408, 19409, 19415, 19420,
    19421, 19422, 19423, 19424, 19425, 19426, 19428, 19429, 19430, 19432,
    19436, 19437, 19438, 19440, 19441, 19442, 19443, 19444, 19446, 19450,
    19451, 19453, 19454, 19455, 19456, 19457, 19460, 19462, 19464, 19465,
    19468, 19473, 19474, 19475, 19477, 19478, 19480, 19481, 19482, 19484,
    19485, 19486, 19490, 19492, 19493, 19494, 19495, 19496, 19520, 19525],
```

```
            "default":  false,
            "filename": "962c3f8e.meta",
            "id":   "962c3f8e614e312c5419dc324ae9593c76b4b46c",
            "name": "PECO PF (-0.95 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  14999,
                    "min":  10001
                }],
            "default":  false,
            "filename": "cc6a0dfe.meta",
            "id":   "cc6a0dfea4be46f083fca5989cea9340e7b51191",
            "name": "NYSEG Freq Trip Low (57Hz)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  14999,
                    "min":  10001
                }],
            "default":  false,
            "filename": "7fdb0b1d.meta",
            "id":   "7fdb0b1d2a1f9c483e75197d3b3dd53686d59a53",
            "name": "RG&E Freq Trip Low (57Hz)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  -1,
                    "min":  -1
                }],
            "default":  true,
            "filename": "b88febee.meta",
            "id":   "b88febee0ea30fd3a8ddd9ee840cfde633aed73d",
            "name": "IEEE-1547"
```

```
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  2799,
                    "min":  1001
                }],
            "default":  false,
            "filename": "cc09cb85.meta",
            "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
            "name": "ISO-NE Ride Through 2018"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "cc09cb85.meta",
            "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
            "name": "ISO-NE Ride Through 2018"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  4999,
                    "min":  4001
                }],
            "default":  false,
            "filename": "cc09cb85.meta",
            "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
            "name": "ISO-NE Ride Through 2018"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  3999,
```

```
                "min":  3001
            }],
        "default":  false,
        "filename": "cc09cb85.meta",
        "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
        "name": "ISO-NE Ride Through 2018"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  2999,
                "min":  2801
            }],
        "default":  false,
        "filename": "cc09cb85.meta",
        "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
        "name": "ISO-NE Ride Through 2018"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  5999,
                "min":  5001
            }],
        "default":  false,
        "filename": "cc09cb85.meta",
        "id":   "cc09cb85389dc96bfdd2557499e7b93f169fa377",
        "name": "ISO-NE Ride Through 2018"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "48813141.meta",
```

```
        "id":   "4881314181d307d82f5a73db8f05af3a1c1dc44e",
        "name": "United Illuminating PF (-0.90 absorb var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "04fa2851.meta",
        "id":   "04fa2851cd3cfc7cc218090b6eb3d8ac58005a9e",
        "name": "United Illuminating PF (-0.91 absorb var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "fc43e4fe.meta",
        "id":   "fc43e4fe8df5ffab5eb67d772b826014d1460090",
        "name": "United Illuminating PF (-0.92 absorb var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "a12c6e66.meta",
        "id":   "a12c6e66467f6bc94badab2cae2aafa13900437f",
        "name": "United Illuminating PF (-0.93 absorb var)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "e8b10eeb.meta",
            "id":   "e8b10eebf00326ee69b12994043ae5e34fd392d8",
            "name": "United Illuminating PF (-0.94 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "6219f2dd.meta",
            "id":   "6219f2ddc37474cb822f79eb40425da56d523fb8",
            "name": "United Illuminating PF (-0.95 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "4f6230b5.meta",
            "id":   "4f6230b52ec893ad631f69823bf60cec8a332361",
            "name": "United Illuminating PF (-0.96 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
```

```
            "default":  false,
            "filename": "ea8e4b4e.meta",
            "id":   "ea8e4b4e31ad9c748b54a7b5d3d6f6cd53787e6f",
            "name": "United Illuminating PF (-0.97 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "0c68e5e6.meta",
            "id":   "0c68e5e6c9d4c843fb5afa140c78132134387d7a",
            "name": "United Illuminating PF (-0.98 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "3f193060.meta",
            "id":   "3f1930609e3012ed502868223da845eb0f5637dd",
            "name": "United Illuminating PF (-0.99 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "78e96ebc.meta",
            "id":   "78e96ebc8d19584f3e6858515005d7a894577443",
            "name": "United Illuminating PF (1.0 Unity)"
```

```
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "7a72fb68.meta",
            "id":   "7a72fb68da48cc366c5d3cf493b0abf76e340d65",
            "name": "United Illuminating PF (0.90 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "2da3fdb7.meta",
            "id":   "2da3fdb7df5aba6aa56bd2914e9ed89670e28082",
            "name": "United Illuminating PF (0.91 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
                    "min":  6001
                }],
            "default":  false,
            "filename": "3f7f7d4d.meta",
            "id":   "3f7f7d4df2a574aa50171558fde4b4abbe8f8152",
            "name": "United Illuminating PF (0.92 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  6999,
```

```
                "min":  6001
            }],
        "default":  false,
        "filename": "67abafc1.meta",
        "id":   "67abafc148733b6bcf1c2d213299fa83ad123e85",
        "name": "United Illuminating PF (0.93 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "88706ef5.meta",
        "id":   "88706ef57c269e9d16bd445fe0bd0cac311ce676",
        "name": "United Illuminating PF (0.94 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "1949f47e.meta",
        "id":   "1949f47e8209187d9089f41ca91fafa572bbf63a",
        "name": "United Illuminating PF (0.95 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "bcc4a49f.meta",
```

```
        "id":   "bcc4a49f6357904b1f2aac738bd7fac44a4cdf67",
        "name": "United Illuminating PF (0.96 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "c27db8fa.meta",
        "id":   "c27db8fabae1cc344357bb4364a41c12103aacca",
        "name": "United Illuminating PF (0.97 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "b332fc6b.meta",
        "id":   "b332fc6b1746f3fde2ab23c971e39b79a3a34b69",
        "name": "United Illuminating PF (0.98 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  6999,
                "min":  6001
            }],
        "default":  false,
        "filename": "ba5d78f2.meta",
        "id":   "ba5d78f256d74413036c424da36666004bc72b9b",
        "name": "United Illuminating PF (0.99 provide var)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "99cd17b1.meta",
            "id":   "99cd17b1b41d7953dea20aab50d0b6302771bd8a",
            "name": "Xcel Energy PF (-0.90 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "d7e21775.meta",
            "id":   "d7e217757b64fa8b300cec77b3e095bc9c6c1269",
            "name": "Xcel Energy PF (-0.91 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "8e37cfa7.meta",
            "id":   "8e37cfa7b3d2a7214eba74a8eeabf671fb5ef0e7",
            "name": "Xcel Energy PF (-0.92 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
```

```
            "default":  false,
            "filename": "41958f46.meta",
            "id":   "41958f469b9721b2e68e47f779e0bf0065f9efb5",
            "name": "Xcel Energy PF (-0.93 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "fcd040c2.meta",
            "id":   "fcd040c2ab28790d88ce358174a3691da2c9db60",
            "name": "Xcel Energy PF (-0.94 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "d36a869e.meta",
            "id":   "d36a869eb424e73d36c8b4133592819ce89df94d",
            "name": "Xcel Energy PF (-0.95 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "4ae087e9.meta",
            "id":   "4ae087e915460658a614abc48c43d16c89e938fc",
            "name": "Xcel Energy PF (-0.96 absorb var)"
```

```
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "e361783c.meta",
            "id":   "e361783cb19dc76046f81de9be5614b959e3e217",
            "name": "Xcel Energy PF (-0.97 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "80aeebb4.meta",
            "id":   "80aeebb4b426a3a2520f67a1040fc1b483e3f089",
            "name": "Xcel Energy PF (-0.98 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "2a5becc6.meta",
            "id":   "2a5becc64606014fc7bfa132ce9cbf00a8102289",
            "name": "Xcel Energy PF (-0.99 absorb var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
```

```
                "min":  80001
            }],
        "default":  false,
        "filename": "7c23f824.meta",
        "id":   "7c23f824b69a5864f851010597e6d7407020db2a",
        "name": "Xcel Energy PF (1.0 Unity)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "a908f0dc.meta",
        "id":   "a908f0dc7ceb5d70dedef20a37ace7a8f1d02a02",
        "name": "Xcel Energy PF (0.90 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "e3e54c42.meta",
        "id":   "e3e54c42084e68c326a774df3f55e093cd917c91",
        "name": "Xcel Energy PF (0.91 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "90e93fd2.meta",
```

```
        "id":   "90e93fd2926de5a464f591bfa433659cbdca24dc",
        "name": "Xcel Energy PF (0.92 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "e114f6db.meta",
        "id":   "e114f6dbded8129202ff352d1adde03b5365b041",
        "name": "Xcel Energy PF (0.93 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "aac6b45c.meta",
        "id":   "aac6b45cf9cfe41812afe6d8f90ca99f14d54b7a",
        "name": "Xcel Energy PF (0.94 provide var)"
    }, {
        "selfsupply":   false,
        "zipcodes": [{
                "max":  81699,
                "min":  80001
            }],
        "default":  false,
        "filename": "eb88ba39.meta",
        "id":   "eb88ba3987f10536bce7eff548953fe1746f5a08",
        "name": "Xcel Energy PF (0.95 provide var)"
    }, {
        "selfsupply":   false,
```

```
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "3e2404d4.meta",
            "id":   "3e2404d455101fec65b656d4014485722772ef02",
            "name": "Xcel Energy PF (0.96 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "4557857a.meta",
            "id":   "4557857a86d79ccdeff01bace9108f434f58f930",
            "name": "Xcel Energy PF (0.97 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
            "default":  false,
            "filename": "241bcb43.meta",
            "id":   "241bcb432a5cac09a6be33a8fded6f67597a6737",
            "name": "Xcel Energy PF (0.98 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  81699,
                    "min":  80001
                }],
```

```
            "default":  false,
            "filename": "f2963273.meta",
            "id":   "f2963273ffd35ee64ec2177dc8b6afa86cd66396",
            "name": "Xcel Energy PF (0.99 provide var)"
        }, {
            "selfsupply":   false,
            "zipcodes": [{
                    "max":  999999,
                    "min":  0
                }],
            "default":  false,
            "filename": "471080f6.meta",
            "id":   "471080f62a24d8be88f58864c398717c11bb876b",
            "name": "IEEE-1547a-2014"
        }]
}
```


GET /cgi-bin/dl_cgi?Command=GetCellPurchased

This command determines (it looks like) whether the customer has purchased cellular based data access to this monitoring system.

- Body


```
{   "cell_purchased":   "null"}
```


GET /cgi-bin/dl_cgi?Command=GetDiscoveryProgress

This command shows the progress status of the device discovery process. In the example below it shows that progress for discovering micro inverters is 100% done and none (new) were found. Since the `progress` is an array, I imagine other kinds of discovery may be reported here as well (probably only during commisioning or changing the system setup).

- Body

```
{
    "progress": [{
            "TYPE": "MicroInverters",
            "PROGR":    "100",
            "NFOUND":   "0"
        }],
    "complete": true,
    "result":   "succeed"
}
```


GET /cgi-bin/dl_cgi?Command=SetCellPurchased&SerialNumber=ZT01234567890ABCDEF

It seems this command can be used to set the purchase status of cellular data support. If done using a GET you get the result below. It may be necessary to issue a POST command with a body to actually make a change, and that probably needs to happen inside an active session.

- Body


```
{
    "cell_purchased":   "null",
    "result":   "succeed"
}
```


Alternative API

Reportedly it is also possible to retrieve information from the Cloud-based SunPower servers. The reported URL is `` `https://elhapi.edp.sunpower.com/v1/elh/address/<address id>/components` `` where `< address id>` is a system specific 6-digit code for your system. You can discover the correct code by using the regular portal, setting your browser in developer mode, and looking in the “network” tab. You will be able to see the code and data that flows.

I was not able to replicate the above, but in my situation, using the same developer tools approach I saw a call to `https://edp-api-graphql.edp.sunpower.com/graphql`. It appears that with the request, a body needs to be sent. It looks like this:

```
[
  {
    "operationName": "FetchPartyData",
    "variables": {
      "partyId": "<a UUID>"
    },
    "query": "query FetchPartyData($partyId: String!) {\n  party(partyId:
$partyId) {\n    partyId\n    displayName\n    email\n    phone\n    sites
{\n      siteKey\n      hasWifi\n      hasLivedata\n      siteName\n
siteType\n      address1\n      city\n      state\n      postalCode\n
systemSize\n      commissioningDate\n      timezone\n      currentWeather
{\n        sunrise\n        sunset\n        dateTime\n        __typename\n
     }\n      battery {\n        operationMode\n
backUpReserveSocLevel\n        backupTimeLeft {\n          formatted {\n
         days\n            hours\n            minutes\n
__typename\n          }\n          __typename\n        }\n
socAndChargeCapacity {\n          customerStateOfCharge\n
stateOfChargePercentage\n          __typename\n        }\n
__typename\n      }\n      assignments(assignmentType: COMMISSION) {\n
   deviceSerialNumber\n        deviceType\n        deviceKey\n
assignmentEffectiveTimestamp\n        devices(deviceType: \"logger\",
deviceStatus: true) {\n          dvcKey\n          comProto\n
__typename\n        }\n        __typename\n      }\n      __typename\n
}\n    __typename\n  }\n}\n"
  }
]
```

Notice that it basically send a query for data, identifying the “party” for which data is needed using a “UUID”. I have not investigated how to find this number, but of course you can look at this data once and write it down. It is unique for your installation and won’t change.

The response looks like this (lots of inverter entries removed):


```
[{
    "data": {
        "party": {
            "partyId": "<UUID>",
            "displayName": "<your name here>",
            "email": "user@gmail.com",
            "phone": null,
            "sites": [{
                "siteKey": "E_1234",
                "hasWifi": true,
                "hasLivedata": true,
                "siteName": "Dolf Starreveld Residence",
                "siteType": "production",
                "address1": "<real address here>",
                "city": "<city here>",
                "state": "CA",
                "postalCode": "<zip here>",
                "systemSize": 8640,
                "commissioningDate": "2021-10-14T17:10:50",
                "timezone": "America/Los_Angeles",
                "currentWeather": {
                    "sunrise": 1635431362,
                    "sunset": 1635470052,
                    "dateTime": 1635445811,
                    "__typename": "CurrentWeather"
                },
                "battery": {
```

```
                    "operationMode": null,
                    "backUpReserveSocLevel": null,
                    "backupTimeLeft": null,
                    "socAndChargeCapacity": null,
                    "__typename": "Battery"
                },
                "assignments": [{
                    "deviceSerialNumber": "ZT213585000549A0748",
                    "deviceType": "DATALOGGER",
                    "deviceKey":
"ZT01234567890ABCDEF_ZT01234567890ABCDEF_PV SUPERVISOR PVS6",
                    "assignmentEffectiveTimestamp": "2021-10-
15T00:10:50.120Z",
                    "devices": [{
                        "dvcKey":
"ZT01234567890ABCDEF_ZT01234567890ABCDEF_PV Supervisor PVS6",
                        "comProto": "MQTT",
                        "__typename": "Device"
                    }],
                    "__typename": "Assignment"
                }],
                "__typename": "Site"
            }],
            "__typename": "Party"
        }
    }
}]
```


Without explaining everything here, this is basic site/party data and identifies the supervisor device only. This device, apparently communicates back to SunPower using the MQTT protocol.

Another request fetches alerts (`FetchAlerts`) and produces:

```
[{
    "data": {
        "site": {
            "siteKey": "E_1234",
            "alerts": [{
                "alertType": "CommunicationOutageInverterICE",
                "alertStatus": "Open",
                "eventTimestamp": "2021-10-21T02:27:58-07:00",
                "__typename": "Alert"
            }, {
                "alertType": "CommunicationOutageInverterICE",
                "alertStatus": "Open",
                "eventTimestamp": "2021-10-21T02:27:59-07:00",
                "__typename": "Alert"
            }],
            "assignments": [{
                "deviceSerialNumber": "ZT01234567890ABCDEF",
                "devices": [{
                    "dvcKey": "ZT01234567890ABCDEF_ZT01234567890ABCDEF_PV
Supervisor PVS6",
                    "prodMdlNm": "PV Supervisor PVS6",
                    "lastRcvdEps": 1635447600,
                    "meterType": null,
                    "__typename": "Device"
                }, {
                    "dvcKey":
"ZT01234567890ABCDEF_E00122125120472_AC_Module_Type_E",
                    "prodMdlNm": "AC_Module_Type_E",
                    "lastRcvdEps": 1634679600,
                    "meterType": null,
                    "__typename": "Device"
                }, {
```

```
                    "dvcKey":
"ZT01234567890ABCDEF_PVS6M12345678p_PVS6M0400p",
                    "prodMdlNm": "PVS6M0400p",
                    "lastRcvdEps": 1635446700,
                    "meterType": {
                        "type": "GROSS_PRODUCTION",
                        "level": "1",
                        "__typename": "MeterType"
                    },
                    "__typename": "Device"
                }, {
                    "dvcKey":
"ZT01234567890ABCDEF_PVS6M12345678c_PVS6M0400c",
                    "prodMdlNm": "PVS6M0400c",
                    "lastRcvdEps": 1635446700,
                    "meterType": {
                        "type": "NOT_USED",
                        "level": "0",
                        "__typename": "MeterType"
                    },
                    "__typename": "Device"
                }],
                "__typename": "Assignment"
            }],
            "__typename": "Site"
        }
    }
}, {
    "data": {
        "site": {
            "hasWifi": true,
            "siteKey": "E_1234",
            "assignments": [{
                "assignmentType": "COMMISSION",
```

```
                "deviceSerialNumber": "ZT01234567890ABCDEF",
                "assignmentEffectiveTimestamp": "2021-10-15T00:10:50.120Z",
                "devices": [{
                    "dvcKey": "ZT01234567890ABCDEF_ZT01234567890ABCDEF_PV
Supervisor PVS6",
                    "dvcTy": "logger",
                    "prodMdlNm": "PV Supervisor PVS6",
                    "lastRcvdEps": 1635447000,
                    "deviceStatus": {
                        "pvStatus": true,
                        "essStatus": false,
                        "netIntfRptCtnt": {
                            "currNetIntfEnum": "STA0",
                            "__typename": "NetworkInterfaceReport"
                        },
                        "__typename": "DeviceStatus"
                    },
                    "__typename": "Device"
                }],
                "__typename": "Assignment"
            }],
            "__typename": "Site"
        }
    }
}, {
    "data": {
        "environmentalSavings": {
            "costSaving": null,
            "co2": {
                "value": 0,
                "unit": "Pounds",
                "__typename": "EnvironmentalSavingsItem"
            },
            "tree": {
```

```
                "value": 0,
                "unit": "Trees",
                "__typename": "EnvironmentalSavingsItem"
            },
            "mile": {
                "value": 0,
                "unit": "Miles",
                "__typename": "EnvironmentalSavingsItem"
            },
            "gas": {
                "value": 0,
                "unit": "Gallons",
                "__typename": "EnvironmentalSavingsItem"
            },
            "__typename": "EnvironmentalSavings"
        }
    }
}, {
    "data": {
        "energy": {
            "energyDataSeries": {
                "production": [
                    ["2021-10-28T00:00:00", "0", "0"],
                    ["2021-10-28T01:00:00", "0", "8"],
                    ["2021-10-28T02:00:00", "0", "0"],
                    // more omitted
                ],
                "consumption": [
                    ["2021-10-28T00:00:00", "0", "100"],
                    ["2021-10-28T01:00:00", "0", "100"],
                    ["2021-10-28T02:00:00", "0", "100"],
                    // more omitted
                ],
                "storage": [
```

```
                    ["2021-10-28T00:00:00", "0", "100"],
                    ["2021-10-28T01:00:00", "0", "100"],
                    ["2021-10-28T02:00:00", "0", "100"]
                    // more omitted
                ],
                "grid": [
                    ["2021-10-28T00:00:00", "0", "0"],
                    ["2021-10-28T01:00:00", "0", "0"],
                    ["2021-10-28T02:00:00", "0", "0"],
                    // more omitted
                ],
                "__typename": "DataSeries"
            },
            "totalProduction": 0,
            "totalConsumption": 0,
            "energyMixPercentage": 0,
            "totalGridImport": 0,
            "totalGridExport": 0,
            "netGridImportExport": 0,
            "totalStorageCharged": 0,
            "totalStorageDischarged": 0,
            "netStorageChargedDischarged": 0,
            "__typename": "Energy"
        }
    }
}, {
    "data": {
        "power": {
            "powerDataSeries": {
                // Formatted as above
                "__typename": "DataSeries"
            },
            "__typename": "Power"
        }
```

```
      }
  }]
```

You will note that a lot of information is available here as well. It may, however, be subject to data aggregation in an interval longer than the frequency with which data changes when you directly query the PVS6 system.

You can see that data in the `site` section (may) contains “Alerts” (containing a type, status and time stamp in ISO format), “Assignments” seem to list all known devices that belong to the “site”.

Another command is `FetchPowerData`. It appears to return time series data for plotting power and energy graphs.

Since I am not using this approach, this is as far as I will take the documentation.