If you just want to use the OVMS module to access your car's diagnostics or do some tuning, you can do so without a SIM card over the module's serial port.

You'll need...

1. OVMS module
2. OVMS OBD2 connector cable
3. USB to serial port (9 pin) adapter (or serial cable if your Laptop still has a serial port)
4. Laptop running some terminal program (i.e. Minicom or HyperTerminal)

If the module is new you probably also need to flash the latest firmware version first to enable all functions, so don't forget the PICkit on your shopping list.

Setup

1. Connect the module's DIAG port to your Laptop and start the terminal program.
2. Set the terminal parameters to **9600 baud, 8N1, no flow control, automatic line feeds**. Check if the backspace key is transmitted as backspace (not del).
3. Activate the terminal log output to a file for later processing.
4. Switch the car on and connect the module's VEHICLE port to the car.
5. Since framework version 2.8.8, this step is only necessary if you've inserted a SIM card (if not, the module will automatically enter DIAG mode): Wait for the first output to appear („RDY“), then type „**SETUP**“ (or "setup") + enter (you might consider assigning this to a macro key; minicom interprets „^M“ as the enter key).
    - **If you do not see a "RDY" line** on powering up the module, your modem needs a **one time** configuration change to set the terminal speed to 9600 baud on boot:
        - let the module connected, wait until output appears
        - type "AT+IPR=9600" + enter => the modem should respond with "OK"
        - type "AT&W" + enter => the modem should respond with "OK"
        - reboot the module (disconnect + reconnect) & try again.
6. The module should now respond with „# OVMS DIAGNOSTICS MODE“ and the green LED should stay continuously on.

```
RDY

GPS Ready

SETUP

# OVMS DIAGNOSTICS MODE
```

You are now in „DIAG mode“, which means the module will not try to access any GSM / GPRS network, instead it will read commands from the serial port and send command output there as well.

The GSM modem is listening to your input as well, you can send „AT“ commands to it (see SIM908 / SIM808 manual for these). OVMS commands are not understood by the GSM modem, so it will output „ERROR“ (sometimes „OK“) for these – that's normal and can be ignored. (Hint: you can turn off the modem responses by "ATQ1" aka "quiet mode", but don't forget to reset to default at the end of your diag mode session.)

There is minimal line editing support: beginning with firmware framework version 2.8.2, you may use the backspace key to correct wrong input characters, and Ctrl-A / Ctrl-C to abort an input. Before 2.8.2 there was no editing, a workaround is to use some text editor to create the commands and copy them into the terminal. Another option is to use the perl client "diagcmd.pl", see below.

OVMS commands can be „SMS“ or „MSG“ commands. SMS commands tend to be more readable, MSG commands are optimized for data processing. All OVMS commands are terminated by the enter key (CR) and need to be sent with exact syntax, i.e. no additional or left out space characters. Command keywords may be given in lower case, but arguments may need to be exact.

Configure car type and CAN bus write access

This only needs to be done once for a new or erased module, or if connecting to another car type. To configure the module for the Renault Twizy, send:

```
m 4,14,RT
```

Note the space character after the „m“. This issues the MSG command #4 to set parameter #14 (car type) to „RT“ (Renault Twizy). The module responds with:

```
# MP-0 c4,0
#.
```

„MP-0“ is the standard result intro for MSG commands, „c4“ means command #4 and „0“ means „OK“. The "#." line tags the end of the command output, I'll omit this from the remaining examples.

Test the car type by e.g. sending the „STAT“ SMS command:

```
s stat

# Not charging
#  Full charge: 20 min.
#  Range: 48 - 48 km
#  SOC: 96.98% (96.98%..97.02%)
#  ODO: 32526.1 km
```

...or by checking the battery pack & cell voltages:

```
s batt v

# P:57.50V ?1:4.115V 2:4.105V 3:4.105V 4:4.105V 5:4.105V
6:4.105V 7:4.105V 8:4.105V 9:4.105V 10:4.105V 11:4.105V
12:4.105V 13:4.105V ?14:4.115V
```

If you want to access the SEVCON, set feature #15 to 1 to allow CAN write access:

```
m 2,15,1
# MP-0 c2,0
```

(MSG command 2 = set feature, you could also use „s feature 15 1“)

SEVCON access & tuning

Straight forward, as shown before: SMS commands are sent with the „s“ prefix. Same applies to the „CFG“ command. Example: set the smoothing to 0...

```
s cfg smooth 0
# CFG SMOOTH: OK scrv=1 srmp=0
```

...and recuperation to 20% neutral and 40% on brake:

```
s cfg recup 20 40
# CFG RECUP: OK ntr=202 brk=404
```

All CFG command options work just the same way as via SMS, so you can also do low level accesses like this:

```
s cfg read 3814 1
# CFG READ:  SDO 0x3814.01: 0x004C = 76
```

Download diagnostics

The SEVCON diagnostic data cannot be accessed via SMS, so you need to use the MSG commands for this via DIAG mode as well.

We do not have a server connection in DIAG mode, but we don't need one: the MSG command #210 telling the module to read and upload the SEVCON diagnostics to the server will just output the records on the terminal instead of sending them to the server.

Example: read the SEVCON event counters:

```
m 210,4
# MP-0 HRT-ENG-LogKeyTime,0,86400,445,200
# MP-0 HRT-ENG-LogCounts,0,86400,488E,Mom dir,441,2,0,112,210
# MP-0 HRT-ENG-LogCounts,1,86400,4582,Safety Case 1,257,122,153,7,4
# MP-0 HRT-ENG-LogCounts,2,86400,5143,VERLOG,363,23,297,130,2
# MP-0 HRT-ENG-LogCounts,3,86400,4F01,Bad State,412,98,410,47,8
# MP-0 HRT-ENG-LogCounts,4,86400,4701,CAN Warn,410,97,410,47,2
# MP-0 HRT-ENG-LogCounts,5,86400,4681,Preop,442,207,410,47,99
# MP-0 HRT-ENG-LogCounts,6,86400,2401,Login,445,197,410,115,174
# MP-0 HRT-ENG-LogCounts,7,86400,4F55,DSP param,438,171,420,65,3
# MP-0 HRT-ENG-LogCounts,8,86400,5043,Param fixed range error,434,109,434,109,1
# MP-0 HRT-ENG-LogCounts,9,86400,0000,,0,0,0,0,0
# MP-0 c210,0,4,10
```

The CSV record format can easily be extracted from the terminal log and read into a spread sheet. Field order is close to the server records, except the primary key (0-9 here) and the expire time (86400 = 24h here) instead of the timestamp.

DIAG mode Perl clients

The Perl client "**diagcmd.pl**" can be used as an alternative to a terminal emulator, for example to script a diag session. It can establish the diag mode automatically and send an arbitrary number of commands to the module.

Example: to initialize diag mode, configure the module for the Renault Twizy and enable CAN write access, you can do:

```
perl diagcmd.pl --init "m 4,14,RT" "m 2,15,1"
```

Due to "--init" the command will first establish the OVMS diag mode:

```
*** INIT: PLEASE POWER UP OR RESET OVMS MODULE NOW
```

Some seconds after powering up / resetting the module you should see:

```
*** INIT: DIAG MODE ESTABLISHED
MP-0 c4,0
MP-0 c2,0
```

You can issue further commands without "--init" once the module is in diagnostics mode.

If diag mode cannot be established, please read section "Setup" (fix terminal speed).

The Perl client "**dcf_download.pl**" can be used to download the complete SEVCON device configuration (register dump with dictionary meta data). This may be useful as a reference/backup if you're about to change registers outside the macro command scope. Btw: the client contains some useful Perl wrapper functions to make reading from & writing to SEVCON registers easy.

Note: you may need to specify your serial device / com port. For more details read client/README, for command options use "--help" on both clients.