OVMS

Open Vehicle Monitoring System

www.openvehicles.com

Renault Twizy v2.5
OVMS Hardware Module v2
User Guide v1.0.1 (2013/01/11)

History

v1.0	2013/01/05	Initial version (RT2.5)
v1.0.1	2013/01/11	Minor fixes & optimization

Table of contents

Welcome!

The OVMS (Open Vehicle Monitoring System) team is a group of enthusiasts who are developing a means to remotely communicate with our cars, and are having fun while doing it.

The OVMS module is a low-cost hardware device that you install in your car simply by installing a SIM card, connecting the module to your car's Diagnostic port connector, and positioning a cellular antenna. Once connected, the OVMS module enables remote control and monitoring of your car.

There are two ways for you to communicate with the OVMS module:

1. Send text messages from a cell phone to the OVMS module's phone number. The module will respond back via text messaging. If you want, the OVMS module can also send text messages to you when the car reaches certain states, such as if charging is interrupted.

2. Use a smartphone App. Both the OVMS module and the App communicate with an OVMS server via UDP/IP or TCP/IP over the Internet. The smartphone Apps provide a richer experience and more functionality, but they do require a data plan on the SIM car you purchase and install in the OVMS module.

This Guide will help you setup and configure your OVMS module. Initial configuration of the OVMS module is done via SMS. Once configured, you can use either SMS and/or the cellphone Apps to communicate with the OVMS module.

Warning!

OVMS is a hobbyist project, not a commercial product. It was designed by enthusiasts for enthusiasts. Installation and use of this module requires some technical knowledge, and if you don't have that we recommend you contact other users in your area to ask for assistance.

Legal disclaimer: by using the OVMS you agree to do so completely on your own risk. Being a hobbyist project, the OVMS has neither CE approval nor undergone any official EMC tests. It has no ECE approval, so depending on your country may not be legal on public roads.

The OVMS module is continuously powered by the car, even when the car is off, but it uses very little power (about **34 Wh per day**).

Warning!

While the OVMS module uses extremely low power, it does continuously draw power from the car's battery, so it will contribute to 'vampire' power drains. Do not allow your car battery to reach 0% SOC, and if it does, plug in and charge the car immediately. Failure to do this can result in unrecoverable failure of the car's battery.

Parts needed

You can buy all parts at the OVMS hardware partner Fasttech:

http://www.fasttech.com/link/ovms

Note: Fasttech is shipping from China, import customs and tax will apply for EU.

For each Twizy you'll need:

- Universal GPS Antenna (SMA Connector)
  GPS Antenna: SMA: $4.89
- OpenVehicles OVMS GSM Antenna
  GSM Antenna: $2.50
- ODB-II to DB9 Data Cable for OVMS
  ODB-II OVMS Cable: $9.50
- OVMS Car Module v2
  OVMS Module: $99.00

If you want to avoid having two antennas: There are combi antennas integrating GSM & GPS available (e.g. "shark fin antenna"). If you test one of these, please report your results :-). Both antenna connectors are SMA, an active GPS antenna needs to run on 3 V.

To update the OVMS firmware, you'll also need one of these:

- PICKIT 2 Compatible Programmer/Debugger
  PICKIT 2 Compatible Programmer: $16.99
- PICKIT 3 Compatible Programmer/Debugger
  PICKIT 3 Compatible Programmer: $26.28

If you want to do OVMS development and/or debugging, a serial interface or serial to USB adapter will be needed.

SIM card

You'll need a **standard size** SIM card with a data plan. The Twizy module will normally need about **1-5 MB of data per month**, depending on your driving and GPS logging. Data though will be sent in small and infrequent packets, so be aware of providers rounding up prepaid data transfers (like Congstar.de) or book some minimal flat rate. Also, mobile **GPRS coverage and stability** depends on the network; for Germany, D1 (T-Mobile) offers best coverage and stability, followed by D2 (Vodafone). E-Plus is usable in urban areas, O2 is not recommended.

Some pointers for Germany (conditions may have changed!):

- T-Mobile "Xtra Call" + "Xtra Flat Daten": D1, best coverage, flat rate required, limit 50 MB (should be sufficient in most cases)

- Congstar "Prepaid Starter" + "Surf Flat Option 200": D1, best coverage, flat rate required, limit 200 MB
- Allmobil: D2, good coverage, fair rates, no flat rate required
- Blau "9 cent Tarif": E-Plus, usable urban coverage, fair rates, no flat rate required

Follow your provider's guidelines on activating the SIM card (full activation can take up to 24 hours). The card needs to be unlocked to start without PIN entry. This can easily be done using your mobile phones SIM card management App. Using your mobile phone you can also test the card activation status and GPRS access.

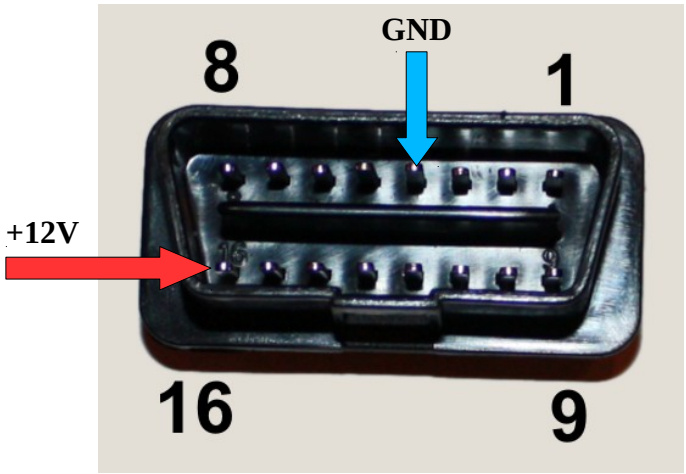
Firmware update / SIM installation

See "Firmware Easy-Install" on the OVMS home page.

Before closing the OVMS, insert the SIM card and make sure the switch besides the LEDs is set inwards (outwards deactivates GPRS).

Dry run

For advanced users: if you want to "dry run" the module on your desk before installing into your car, you just need to provide 12 V =. The module needs about 110 mA. Connect GND to the OBD-II plug at pin 4 and +12V at pin 16:



If you connect the antennas, you can fully interact with the module on your desk using SMS commands and the OVMS Apps.

You can also connect to the DIAG port serial interface (9600 baud, 8N1, no handshake) to see the

module working. If you want to preconfigure modules for other users, you can also use the DIAG mode to enter configuration commands without need of using the GSM network. See document "Firmware-Development" for details.

Car installation

The OBD-II port (diagnostic port) is located at the bottom of the left front glove box. Remove the connector cover and plug in the data cable (only possible in one orientation). Attach the antennas and connect the module to the data cable. The module will power on as soon as the data cable is connected.

For a first installation, it's sufficient to simply put the complete OVMS module and both antennas into the left front case. This will normally provide enough signal quality for some first tests, may even be sufficient for productive use – try out.

There are many options for a better placement of both the module and the antennas. For example you may want to put the module in the right (lockable) glove box – for a first simple installation, the data cable will fit under the glove box cover, you may later decide to hide it under the dashboard.

Antenna position should be best at the center top back of the Twizy (there's no steel frame). Cables can be put under the plastic body covers, see forums for instructions on how to unmount those. You may want to leave some space between the GSM and GPS antenna, as the GSM antenna can emit short high power bursts.

Configuration

Basic registration and configuration of the Twizy OVMS is basically the same as for the Tesla Roadster, except the **vehicle type** needs to be "RT" (for "Renault Twizy").

Follow the Tesla Roadster configuration guide. After "REGISTER" and "PASS", you should first send the "MODULE" command with the fourth argument set to "RT", for example:

```
MODULE TWIZY42 K SMSIP RT
```

This will activate the Twizy specific vehicle data processing and command extensions.

The module should now respond to the "VERSION" command like this:

```
OVMS Firmware version: 2.2.2/RT2.5/V2
```

This is the framework version, the vehicle type + version and the hardware version.

If you haven't done already, turn the Twizy on, so the module can read the diagnostic data from the CAN bus. **The Twizy CAN bus is available only while the car is on or charging.**

Now, send a "STAT?" SMS to the module. It should reply with the Twizy specific status message looking like this:

```
Not charging
Full charge: 18 min.
Range: 59 - 78 km
SOC: 97.38% (50.00%..97.38%)
ODO: 4437 km
```

Next, send a "GPS?" SMS to the module to query the current coordinates. Please note the module may need about a minute after setup as "**RT**" to get the first GPS fix.

Twizy specific commands & capabilities

The OVMS framework defines a common set of commands and capabilities originating at the Tesla Roadster implementation (see Tesla guide). As the framework currently evolves to cover any kind of vehicle, the commands and capabilities now become dynamic properties of the vehicle type.

Some common commands will have no effect on the Twizy, for example locking and unlocking the car. Some common diagnostic capabilities are "virtual" on the Twizy (derived), and some capabilities are unavailable.

The Twizy also changes some standard command behaviour and adds some new commands.

The Apps currently only provide a user interface for the standard capabilities but will support the vehicle specific configuration in the future. At the time writing, most Twizy specific functions need to be adressed by SMS, nevertheless the Apps still provide a nice UI for the standard functions.

(Todo: Feature support matrix)

Overview of Twizy SMS commands

Commands	Function
STAT?	Status output (extended)
RANGE ...	Ideal range configuration
CA ...	Charge alerts (SOC/range)
BATT ...	Battery monitoring
POWER ...	Power usage statistics

Extended Status

The Twizy includes the following information in the SMS command "STAT?" response as well as

battery status text messages sent to the App:

- Charge state: one of
  - "Not charging"
  - "Charging"
  - "Charging Stopped" – if charge was interrupted
  - "Charging, Topping off" – if SOC is above charge alert or 94%
  - "Charging Done" – if fully charged
- Charge power sum (Wh)
- Time estimation for full charge (minutes)
- Range (unit as configured)
- SOC (State of charge in %)
- Odometer (unit as configured)

Being able to rely on a charge working as planned is crucial. If charging is interrupted, the OVMS will automatically send the extended status message by SMS and/or IP (as configured). So you'll be informed immediately if some fuse blows or someone pulls the plug.

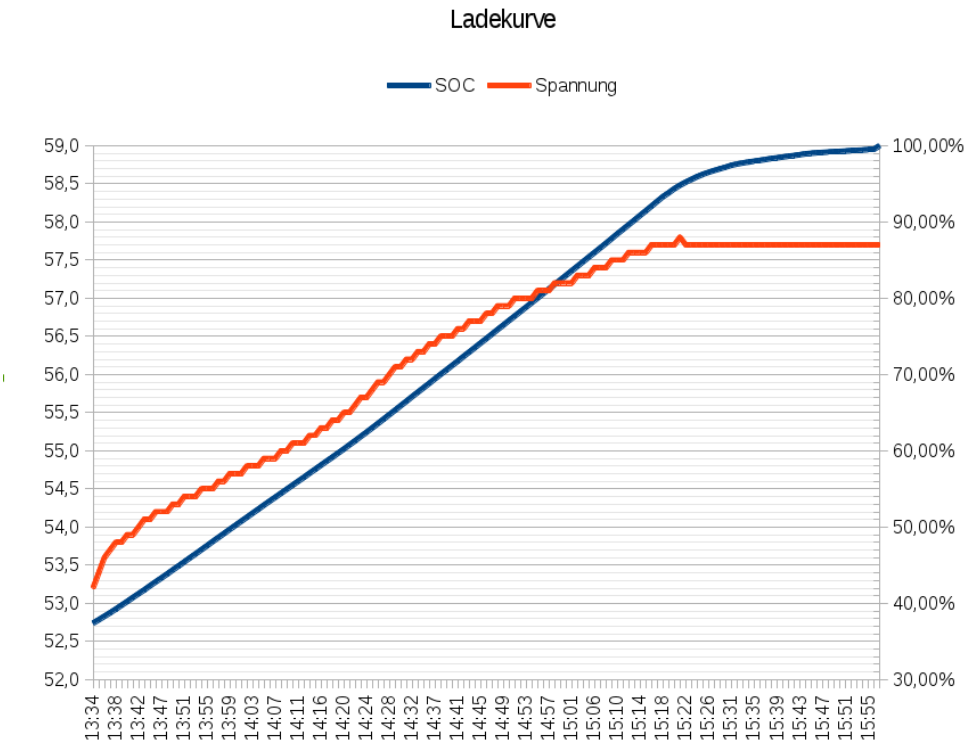
The status message will also be sent if charging starts after an interrupt, and when fully charged. You may also configure a minimum SOC alert (using standard `FEATURE` #9), and a charge alert for a sufficient range and/or SOC (see "Charge alerts").

Charge time estimation

The SOC value of the Twizy (any Lithium SOC value to be precise) is a battery model based estimation and will be quite wrong sometimes (too low normally, especially on low temperatures). The Twizy recalibrates it's SOC to 100% when full charge is reached.

As the charge time estimation is designed to be on the safe side, times normally will be better by 5-30 Minutes depending on the actual cell state and the amount of error in the Twizy estimation.

Due to the Lithium chemistry and battery pack structure (cell balancing), the charge speed is nearly linear up to ~94% SOC and much slower on the last ~6%. A typical charge curve looks like this:



(BTW: you can draw curves like that by using the "`RT-PWR-BattPack`" records, see below!)

Time estimation takes this non-linearity into account by dividing the charge process in two sections (0-94% = 180 minutes / 94-100% = 40 minutes).

Keep in mind: if you need the quickest possible intermediate charge, start with a warm battery and stop at or below 95%.

Range

Range will include two values, an estimated and an ideal range. The ideal range is based on your max range configuration (see "Ideal range configuration"). Estimated range is based on the Twizy's own range estimation, which in turn is based on the last 3 km driving. This will be scaled up to the current SOC during charge, so will differ from the value, the Twizy will tell you on the next key-on (as that is a different Twizy estimation based on the last 150 km). As soon as you turn on the Twizy, the module will update it's estimation as well.

Please note: The OVMS firmware internally stores odometer and range values in miles. Until that can be changed, you will occasionally see a 1 km difference between the range displayed in the dashboard and the OVMS outputs. This is due to the fact integer mile values do not provide enough precision to cover all possible km values. Conversion math has been tuned to minimize this error.

SOC

The SOC output includes the current SOC and the current SOC usage range (lowest & highest SOC). The SOC range reflects your last battery usage or charging cycle.

As mentioned above, the SOC is a model based estimation of the Twizy. So it's quite normal it will sometimes be wrong. To protect the battery from over-discharging, it will normally tend to show you a SOC lower than real. There's currently nothing the OVMS can do about that.

Ideal range configuration

The Twizy's range estimation is based on the energy use of the last 3 km driving, and it will drop faster on high usage than it will raise on low. So if you happen to have a hill climb at the beginning of your tour, the estimation will be completely wrong for a long time.

Most people don't change their driving style and routes frequently and will get about the same range from a full charge most of the time. So, using the OVMS, you can configure your personal maximum range to let the OVMS do a second "ideal" range estimation just on base of the current SOC.

Enter ranges in the unit configured (assumed "K" = km here).

Command	Function
`RANGE 80`	Set ideal maximum range to 80 km
`RANGE?`	Query current ideal maximum range setting
`RANGE`	Disable ideal range estimation

The range setting is stored in feature slot #12, so can also be queried by the OVMS "`FEATURES?`" command.

Charge alerts

If you need to wait for a charge to get somewhere (home for example), you can let the OVMS notify you as soon as a sufficient charge level has been reached.

Sufficient charge alerts can be triggered by a SOC value and/or by a range. If both triggers are set, it will use the first one reached.

Command	Function
`CA 30`	Enable charge alert on 30 km range
`CA 60%`	Enable charge alert on 60% SOC
`CA 30 60%`	Enable charge alert on either range or SOC
`CA?`	Query current charge alerts

Command	Function
CA	Disable charge alerts

Range alerts will be triggered on ideal range reached. If you want to use the Twizy range estimation, disable the ideal range estimation for that charge.

All "CA" command responses includes a charge time estimation for the nearest alert threshold set as well as for a full charge.

Charge alert thresholds are stored in feature slots #10 (SOC) and #11 (range), so can also be queried by the OVMS "FEATURES?" command.

Battery monitoring

Although the battery is leased, it's health status still is of vital importance for us users. If you want to or need to rely on your car, you must be able to detect all signs of potential problems, before they become hazardous.

Fortunately, the Twizy BMS (battery management system) outputs some values on the CAN bus, so the OVMS can provide a little more insight about the battery status than just the SOC. Values currently identified on the CAN bus include cell voltages, pack voltages and cell module temperatures.

Some tech background

The battery pack of the Twizy consists of 42 cells, arranged in 7 modules of 6 cells each. Within a module, cells are layed out 2S3P, that means there's a total series of 14 packs of 3 parallel cells each. Cell chemistry is supposed to be LiMn based, so each cell has a theoretical voltage range of about 2,5 - 4,3 V, nominal voltage around 3,6-3,7 V. Renault restricts the usable SOC window to maximize battery cycle life. Charging top voltage is 57,7 V for the pack = 4,12 V per cell.

Important for good performance of a battery pack is that all cells in a serial configuration are at about the same voltage level and temperature. If a single cell drifts out of line in either direction, that's a sign of some possible defect.

Now there are 14 cell voltages and 7 module temperatures on the CAN bus, plus two pack voltages. So, what the OVMS can do to detect potential problems, is to monitor all cell voltages and temperatures, and to produce an alert if it detects a critical deviation. It also collects the maximum deviations from the mean values over a usage cycle and sends these statistics on SMS request as well as to the OVMS server for further analysis and evaluation.

The limits are, the 14 voltages are collective for 3 parallel cells each, and the 7 temperatures each cover a module of 6 cells. So we're not quite on the single cell level yet, but it's sufficient to detect critical situations.

Using the battery monitor

The battery monitoring needs no configuration. It's enabled by default and works automatically in the background and normally will only alert you about critical conditions.

Critical conditions are:

- Cell voltage offset from mean value is higher than 100 mV
- Module temperature offset from mean value is higher than 3 °C

In addition to critical conditions, the monitoring will also set a "watch" flag if the voltage or temperate has an offset from the mean value that is higher than the current standard deviation of all cells / modules. You may want to keep an eye on these cells, if the deviations tend to raise over time, these may be the source of performance loss or even become the part that breaks.

All battery statistics and alert flags are **automatically reset each time you start the Twizy**. So if an alert occurs, you can easily see if it was a temporary stress issue or if it persists with the next drives.

Remember, the CAN bus is offline while the car is switched off, so temperatures and voltages are not updated while the car is off. The last values read from the bus will be kept by the OVMS.

Command	Function
`BATT`	Output current alert and watch status. This will be sent automatically if any alert condition changes, but not on watch conditions. Example: `Volts: ?C1:+10mV Temps: OK` Read: Cell 1 had a max deviation of 10 mV above average, thus has been tagged "suspicious". An alert would be tagged by a "!". Temperatures have been nominal.
`BATT V`	Output current voltage levels. Example: `P:57.50V ?1:4.120V 2:4.110V 3:4.110V 4:4.110V 5:4.110V 6:4.110V 7:4.110V 8:4.110V 9:4.110V 10:4.110V 11:4.110V 12:4.105V 13:4.105V 14:4.120V` Read: Pack voltage is 57.5 V, followed by the 14 cell voltages. Note the "?" tagging the suspicious cell. Precision of cell voltages is 0.005 V.
`BATT VD`	Output collected max voltage deviations. Example: `P:57.50V ?1:+10mV 2:+0mV 3:+0mV 4:+0mV 5:+5mV 6:+0mV 7:+0mV 8:+0mV 9:+0mV 10:+0mV 11:+0mV 12:-5mV`

Command	Function
	`13:-5mV 14:+10mV` Read: Cell 1 (suspicious) was at most 10 mV above average, cell 12+13 had a max deviation of 5 mV below average. Note cell 14 also once had +10 mV deviation, but that did not trigger the "watch" flag, so was inside standard deviation at that time.
`BATT T`	Output current cell module temperatures. Example: `P:10C (10C..10C) 1:10C 2:10C 3:10C 4:10C 5:10C 6:10C 7:10C` Read: Current pack temperature is 10 °C, temperature range of last usage was from 10..10 °C. Following are the 7 module temperatures, no watches or alerts.
`BATT TD`	Output collected max temperature deviations. Example: `P:10C (10C..10C) 1:+0C 2:+0C 3:+0C 4:+0C 5:+0C 6:+0C 7:+0C` Read: again, pack temperature and range, followed by the 7 max deviations during the last use.
`BATT R`	Reset battery monitor. Normally automatically done on each switch-on, the command enables resets during charge or without turning off the car.

More details and log records may be queried from the OVMS server, using historical message records of types "`RT-PWR-BattPack`" and "`RT-PWR-BattCell`". See "Perl client" on how to retrieve these.

Battery pack record format:

```
RT-PWR-BattPack,<timestamp>,<packnr>,
      ,<nr_of_cells>,<cell_startnr>
      ,<volt_alertstatus>,<temp_alertstatus>
      ,<soc>,<soc_min>,<soc_max>
      ,<volt_act>,<volt_act_cellnom>
      ,<volt_min>,<volt_min_cellnom>
      ,<volt_max>,<volt_max_cellnom>
      ,<temp_act>,<temp_min>,<temp_max>
```

- "alertstatus" can be one of
  - 0 = unknown

  - 1 = nominal
  - 2 = watch
  - 3 = alert
  - 4 = failure
- SOC values are in 1/100 %
- Voltages are in 1/100 V; "cellnom" = nominal single cell voltage
- Temperatures are in °C
- Min & max values are from the current use cycle (since last switch-on)

Example:

```
RT-PWR-BattPack,2013-01-01 19:06:45,1,14,1
      ,2,1
      ,8612,8612,9698
      ,5610,400,5390,380,5740,410
      ,10,8,10
```

(line breaks added for readability)

Battery cell record format:

```
      RT-PWR-BattCell,<timestamp>,<cellnr>,<packnr>
              ,<volt_alertstatus>,<temp_alertstatus>,
              ,<volt_act>,<volt_min>,<volt_max>,<volt_maxdev>
              ,<temp_act>,<temp_min>,<temp_max>,<temp_maxdev>
```

- "alertstatus" is encoded as above (battery pack)
- Voltages are in mV (but Twizy sensor resolution is 5 mV)
- Temperatures are in °C
- "maxdev" = maximum deviations, these are signed to denote the direction

Example:

```
RT-PWR-BattCell,2013-01-01 19:06:45,1,1
      ,2,1
      ,4015,3865,4115,15
      ,10,8,10,1
```

(line breaks added for readability)

Power usage statistics

One of the things the Twizy lacks is a usable ecometer, to give you feedback on your driving style. The builtin ecometer (the four circles) is not very helpful in optimizing power usages.

The power statistics of the OVMS is a first attempt (read: work in progress) to provide some more

detailed information about your power usage profile. It currently is limited to text message and server records, a graphical user interface needs to be implemented in the App.

Power statistics currently are collected in two categories, speed and level. The speed category is divided into three sections, constant speed, acceleration and deceleration. The level category is divided into two sections, up and down. In each section, used and gained (by recuperation) power will be collected separately.

Be aware, that level change detection depends on the GPS altitude. So, if GPS altitude is unavailable or inaccurate, so are level power usages.

Power sums are currently totals of Wh for the sections, collecting section track lengths is on the todo list. Until then you might consider correlating the GPS log with the power usage log.

Power statistics are currently reset when switching the Twizy on, and the report is sent after switching the Twizy off. No configuration is currently needed, statistics are enabled by default and will run in the background.

Command	**Function**

`POWER`	Output current power statistics report. This is also sent automatically when turning the Twizy off (and at least 1 Wh was used).
	
	Example:
	`Power -2460 +77 Wh`
	`Const 48% -1221 +21 Wh`
	`Accel 26% -769 +9 Wh`
	`Decel 26% -470 +48 Wh`
	`Up 191m -765 +22 Wh`
	`Down 149m -412 +14 Wh`
	
	Read: total power used on that trip was 2460 Wh, gained 77 Wh. Following three lines give the speed sections with their percentage of the whole tour. Up & down show the level sections, the meters are the totals of height differences collected.

Besides text notification, the function will send power usage records to the OVMS server once per minute. The history message type is "`RT-PWR-UsageStats`", see "Perl client" on how to retrieve these from the server. Hold time is 24 hours.

Power usage record format:

```
RT-PWR-UsageStats,<timestamp>,0
     ,<speed_CONST_cnt>,<speed_CONST_use>,<speed_CONST_rec>
     ,<speed_ACCEL_cnt>,<speed_ACCEL_use>,<speed_ACCEL_rec>
     ,<speed_DECEL_cnt>,<speed_DECEL_use>,<speed_DECEL_rec>
```

```
      ,<level_UP_hsum>,<level_UP_use>,<level_UP_rec>
      ,<level_DOWN_hsum>,<level_DOWN_use>,<level_DOWN_rec>
```

- All "use" and "rec" values are in internal integer representation, to convert to Wh divide by 22500.
- "cnt" values represent time amounts, unit 1/100 second (10 ms = power stats resolution)
- "hsum" values are totals of height changes in meters

Example:

```
RT-PWR-UsageStats,2013-01-01 18:53:59,0
      ,26397,956842,1624
      ,3132,684299,4037
      ,3599,319977,19455
      ,38,920895,730
      ,5,834823,3751
```

(line breaks added for readability)

GPS track logging

The Twizy OVMS will automatically send GPS records to the OVMS server. This is an extension to the standard location message of the framework: the records are history entries that will be kept for 24 hours, so you can retrieve a complete GPS track after your trip.

You'll currently need to use the perl client to retrieve these records. Record type is "`RT-GPS-Log`". Remember, records will be deleted after 24 hours.

Normal checkpoint frequency is one per minute. If you enable location streaming mode via App or by setting "`FEATURE 8 1`", log entries will be sent every five seconds.

GPS log record format:

```
RT-GPS-Log,<timestamp>,<odometer_mi_10th>
      ,<latitude>,<longitude>,<altitude>
      ,<direction>,<speed>
      ,<gps_fix>,<gps_stale_cnt>,<gsm_signal>
```

- Odometer is in 1/10th miles
- Speed will be output in the unit configured (mph/kph)
- GPS fix is 0/1
- Stale counter begins at 120 on reception of valid coordinates, counts down about once per second to 0
- GSM signal quality is in the range 0 .. 31, higher means better

Example:

```
RT-GPS-Log,2013-01-01 19:03:40,27461,51.257704,7.160899,139,242,48,1,119,20
```

Antenna optimization

The GPS log entries include the current vehicle speed, a GPS stale counter (counting down from 120 to 0, the lower it is the staler the coordinates) and the GSM signal quality (value range 0..31, the higher the better).

Analyzing these values + connection drops, you can use the GPS log to optimize your antenna positioning:

- Define a fixed route for a test drive.
- Take the tour, retrieve the GPS log.
- Change the antenna position.
- Take another tour, retrieve the new GPS log.
- Compare GPS and GSM signal qualities along the track, keep the antenna position that delivers better values.

Perl client

The perl client is a simple text client for the OVMS server. It communicates with the server just the same way an App does, but it's not limited by a graphical user interface. Think of it as an OVMS shell.

To use the perl client, you need a working perl installation and some additional perl packages (all available by CPAN). Follow the instructions in the "HOWTO-Server" document.

The command client is "cmd.pl" located in the "server" directory. It needs a vehicle and password configuration, edit "ovms_client.conf" accordingly.

Cd to the server directory. Some usage examples:

```
./cmd.pl 31
```

=> lists the directory of currently stored history records

```
./cmd.pl 32 "RT-GPS-Log" >gpslog.csv
```

=> save all GPS log records to "gpslog.csv"

```
./cmd.pl 41 "*100#"
```

=> query SIM card account balance (change USSD code if needed)

Thanks to...

- the whole OVMS team for developing the OVMS platform
- Mark Webb-Johnson for excellent support and constructive feedback
- Bruce McMillan for the Twizy App artwork
- everyone at the german Twizy forum http://www.twizy-online.de/ who participated in decoding the CAN messages
- Renault for the Twizy
- Tesla Motors for kicking fossilized butts just in time

Contact / Feature requests

If you need help, want to give some feedback, find bugs, have an idea on improving or miss some feature, please don't hesitate to post on the OVMS forum:

http://www.openvehicles.com/forum

If you want to take part in the development in any way, please subscribe to the OVMS developers mailing list:

http://lists.teslaclub.hk/mailman/listinfo/ovmsdev

Remember, this is a community project, any help is appreciated :-)

Thank you!