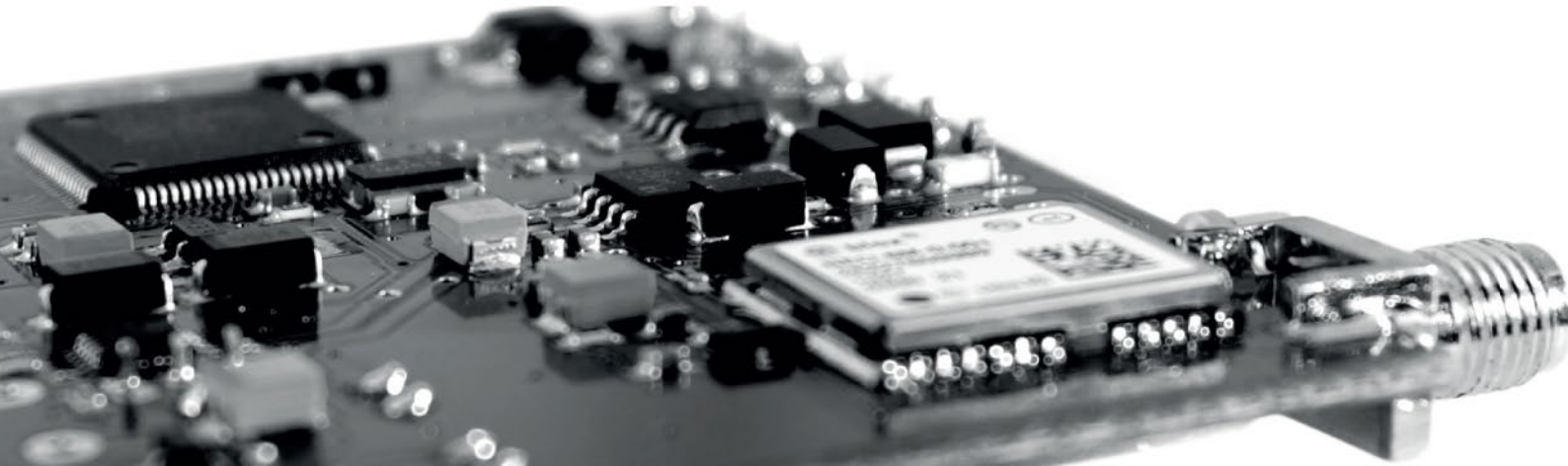
Geotik Sp. z o.o.
ul. Kolejowa 5/7
01-217 Warszawa

NIP: 527 279 37 21
REGON: 366238011
KRS: 0000659463

GPS LOGISTIC PRO

GPS LOGISTIC PRO MANUAL



Contents

1. Release history 3
2. Hardware 6
2.1. Technical details GLP 2.0 6
2.2. Technical details GLP 1.5 8
2.3. LED indicator GLP 1.5 9
2.4. LED indicator GLP 2.0 9
2.4 Installation guidance GLP 1.5 10
2.5 Installation guidance GLP 2.0 11
2.6 EcoDriving 13
2.7 Search for CAN data 13
2.8 Bluetooth interface for technical support (optional) 15
3 Commands and variables 16
3.4 Telephone lock level 16
3.5 Executable commands 17
3.6 Read-only variables 22
3.7 Configuration variables 22
4 Configuration of basic functions 34
4.4 Unlock the device 34
4.5 Communication with server 34
4.6 Update procedure 34
4.7 Update procedure over USB connection for GLP 2.0 35
5 Firmware changelog GLP 2.0 35
6 Firmware changelog GLP 1.5 37
7 Error codes 42
7.4 From GSM modem 42
7.5 From HTTP server 43
7.6 Update Errors 43
8 Data Acquisition Triggers 44
9 Jamming / interference monitor reported states 45
10 Indicators from CANbus 46

1. Release history

Document version	Description
12/12/20	• Please check http://bit.ly/GLP_-_changelog
10/09/20	• Added GSM module update in GLP 2.0 firmware (.dfotaupdate). Requires firmware 1.02.39 or later. • Added new Data Acquisition Trigger #246
22/08/20	• Updated GSM_CFG description. APN and GSM module mode works from firmware version 1.02.37 (GLP 2.0 only)
06/08/20	• Corrected SafeCAN connection in GLP 2.0
05/08/20	• Updated IGNITION_CFG to reflect current options
23/07/20	• Corrected reason of acquisition number from 65 to 68 - Immobilizer ON (active) • Added DroneKey integration
06/07/20	• Updated GSM module details
29/06/20	• Updated GLP 1.5 schematic
12/06/20	• Added firmware update over USB connection for GPL 2.0 • GLP 2.0 changelog link added • .bootupdate added to firmware 1.02.19
07/05/20	• GLP 2.0 PCB layout updated to version 2.0.3; tamper detection hardware foundation added • Installation guidance GLP 2.0 updated
27/04/20	• Updated description of CAN_ID – Bank2 added and FLAG description edited
1.02.19 / 4.02.19 08/04/20	• Added .accauto – autocalibration process.
14/02/20	• Added jamming reported states for older firmware • Restored release history from GLP 1.5 and older
07/02/20	• Added indicators from CANbus
05/02/20	• Corrected wiring diagram on Installation Guidance page
04/02/20	• Acquisition trigger – added description; available from fw 2.71.50 / 3.10.50 / 4.1.50 • Jamming/interference monitor reported states
1.02.11 20/09/19	• New firmware
1.02.10 06/09/19	• New firmware
1.02.09 20/08/19	• New firmware
1.02.08 15/08/19	• New firmware
1.02.07 18/07/19	• New firmware • Added new functions: .updcmd & .gprscmd
1.02.06 11/07/19	• New firmware
1.02.05 19/06/19	• New firmware
1.02.03 07/06/19	• New firmware
1.02.02 30/05/19	• New firmware
1.02.1 29/05/19	• New firmware
1.01.63 02/08/2019	• New firmware

1.01.62 01/08/2019	• Added GLP 2.0
V4.01.11 18/12/17	• New firmware
V3.10.09/V4.01.09 **V5.01.09/V7.01.09** 16/11/17	• New firmware
V4.01.08 29/10/17	• New firmware
V4.01.07 22/10/17	• New firmware
V2.72.06/V3.10.06 **V4.01.06/V5.01.06** **V7.01.06** 11/10/17	• New firmware
V2.72.02/V3.10.02 **V4.01.02/V5.01.02** **V7.01.02** 27/08/17	• New firmware
V2.72.01/V3.10.01 **V4.01.01/V5.01.01** **V7.01.01** 20/08/17	• New firmware
V2.71.09/V3.09.09 **V4.00.09/V5.00.09** **V7.00.09** 08/06/17	• New firmware
V2.71.01/V3.09.01/01 08/05/17	• New firmware V2.71.01/V3.09.01
V2.71.00/V3.09.00/01 20/02/17	• New firmware V2.71.00/V3.09.00 • New chapter "Bluetooth interface for technical support"
V2.70.02/V3.08.02/01 08/11/16	• New firmware V2.70.02/V3.08.02
V2.70.01/V3.08.01/01 13/09/16	• New firmware V2.70.01/V3.08.01
V2.69.06/V3.07.06/01 11/08/16	• New firmware V2.69.06/V3.07.06
V2.69.05/V3.07.05/01 27/07/16	• New firmware V2.69.05/V3.07.05
V2.69.02/V3.07.02/01 07/07/16	• New firmware V2.69.02/V3.07.02
V2.69.01/V3.07.01/01 28/06/16	• New firmware V2.69.01/V3.07.01
V2.68.06/V3.06.06/01 22/06/16	• New firmware V2.68.06/V3.06.06
V2.68.04/V3.06.04/01 16/06/16	• New firmware V2.68.04/V3.06.04
V2.68.02/V3.06.02/01 17/05/16	• New firmware V2.68.02/V3.06.02
V2.67.08/V3.05.08/01 29/02/16	• New firmware V2.67.08/V3.05.08
V2.67.05/V3.05.05/01 17/02/16	• New firmware V2.67.05/V3.05.05
V2.67.03/V3.05.03/01 10/02/16	• New firmware V2.67.03/V3.05.03 • Added "Camera Data" in protocol
V2.66.03/V3.04.03/01 02/02/16	• New firmware V2.66.03/V3.04.03
V2.66.01/V3.04.01/01 20/01/16	• New firmware V2.66.01/V3.04.01
V2.65.01/V3.03.01/01 14/12/15	• New firmware V2.65.01/V3.03.01

V2.64.06/V3.02.06/01 09/12/15	• New firmware V2.64.06/V3.02.06 • New chapter "GLP extension hack – mounting instruction"
V2.64.03/V3.02.03/01 26/10/15	• New firmware V2.64.03/V3.02.03
V2.63.06/V3.01.09/01 08/10/15	• New firmware V2.63.06/V3.01.09
V2.63.04/V3.01.07/01 25/09/15	• New firmware V2.63.04/V3.01.07 • Added "Info 32" (terminal confirmation) in protocol
V2.63.03/V3.01.06/01 15/09/15	• New firmware V2.63.03/V3.01.06 (3G version) • In protocol added new function to parameter 11
V2.62.08.01 11/09/15	• New firmware V2.62.08
V2.62.06.01 28/08/15	• New firmware V2.62.06 • Added beta version package
V2.62.04.01 18/08/15	• New firmware V2.62.04
V2.62.03.01 17/08/15	• New firmware V2.62.03
V2.61.05.01 28/07/15	• New firmware V2.61.05 • In protocol added string parameter 244 and all string parameters are shortened to 64 bytes
V2.60.05.01 15/07/15	• New firmware V2.60.05
V2.60.03.01 13/07/15	• New firmware V2.60.03
V2.60.02.01 07/07/15	• New firmware V2.60.02
V2.59.15.01 10/06/15	• New firmware V2.59.15 • New chapter "Search for CAN data"
V2.59.14.01 08/06/15	• New firmware V2.59.14 • Changes in protocol – CSQ and High Resolution Speed
V2.59.10.01 22/05/15	• New firmware V2.59.10 • New chapter "LED indicator" • New chapter "EcoDriving"
V2.58.11.02 21/04/15	• New chapter "Technical details" • Added eco driving parameters to protocol
V2.58.11.01 08/04/15	• New firmware V2.58.11 • Added Information about supported parameters by FW
V2.58.06.01 26/03/15	• New firmware V2.58.06 • Added a few error codes • Fix description about EVENTS and ACTIONS • New subsection "Update Errors"
V2.57.02.01 05/02/15	• New firmware V2.57.02 • Added information in "Installation guidance"
V2.56.01.02 21/01/15	• Added information about SafeCan
V2.56.01.01 16/01/15	• New firmware V2.56.01
V2.55.02.01 05/01/15	• New firmware V2.55.02
V2.55.01.01 15/12/14	• New firmware V2.55.01 • Changes in protocol – param 1
V2.54.12.01 28/11/14	• New firmware V2.54.12
V2.54.05.01 25/11/14	• New firmware V2.54.05
V2.54.03.01 30/10/14	• New firmware V2.54.03 • New variables • Editorial corrections
V2.53.02.02 02/09/14	• Fix in update chapter and installation guidance

V2.53.02.01 26/06/14	• New firmware V2.53.02
V2.53.01.01 16/06/14	• New firmware V2.53.01 • New commands and variables • New chapter "Error codes"
V2.52.05.02 27/05/14	• QR codes added
V2.52.05.01 05/05/14	• New firmware V2.52.05 and V2.52.04 • New commands and variables • New chapter "Telephone lock level"
V2.52.02.02 30/04/14	• Protocol chapter reorganization • Fix - SERVER_CODING default is 0, not 1 • Add date to release history and firmware changelog
V2.52.02.01 24/03/14	• New firmware V2.52.02
V2.52.01.02 03/03/14	• Protocol added
V2.52.01.01 24/02/14	• Document created

2. Hardware

2.1. Technical details GLP 2.0

Element	Specification
Interface	• Power supply 7 – 36 V • 3x Analog input, 0 – 36V, 12bit, 1600Hz, averaging of 1024 samples • 7x Digital inputs • RS485 A/B interface • K-Line interface • 2x CAN-H, CAN-L (SAE J1939, SAE J2411) • SafeCAN connector (CAN-H, CAN-L) SAE J1939 • 4x Digital output, open collector, max 2A • 1x OneWire • 5V output for SafeCan or OneWire devices • Optional encrypted (AES256) wireless connection on 869.525Mhz (Sub Ghz RF module) • Optional Bluetooth • 2xLED status • MiniSIM card slot • Antenna connectors SMA female outer shell, female inner pin

GSM module (2G, LTE Cat.M1/NB-IoT)	• LTE Cat M1/Cat NB1/EGPRS module
	maximum data rate of 300Kbps downlink and 375Kbps uplink
	  • Cat M1/Cat NB1:
	  • LTE FDD: B1/B2/B3/B4/B5/B8/B12/B13/B18/ B19/B20/B26/B28
	  • LTE TDD: B39 (For Cat M1 Only)
	  • 2G (EGPRS): 850/900/1800/1900MHz
	  • The transmitter power: max. power: 23dBm

GNSS module	• System support: GPS, GLONASS, BeiDou/Compass, Galileo, QZSS
	• Sensitivity:
	  ○ Tracking: -157 dBm
	  ○ Reacquisition: -157 dBm
	  ○ Cold start: -146 dBm
	• Hot start < 2.7sec
	• TTFF
	  ○ Cold start (open sky): 31s (autonomous); 11.54s (XTRA enabled)
	  ○ Warm staty (open sky): 21s (autonomous); 2.52s (XTRA enabled)
	  ○ Hot start (open sky): 2.7s (autonomous); 1.82s (XTRA enabled)
	• Horizontal position accuracy: < 2.5m (CEP 50%; autonomous, open sky)

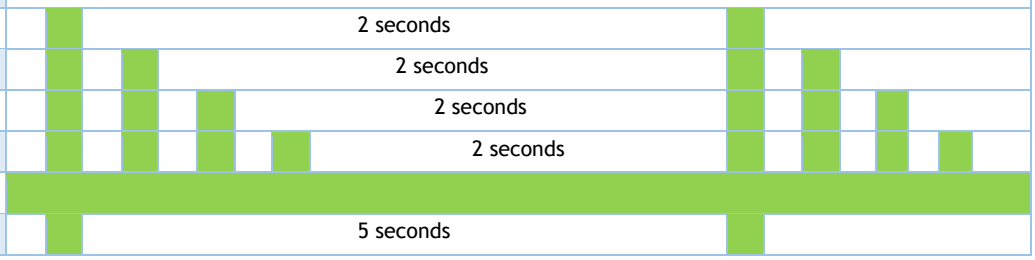
Power consumption	• Sleep mode – 7.3mA @ 12V, 3mA @ battery • Normal mode - 70mA @ 12V, 90mA @ battery • Sending data - 250mA @ 12V, 350mA @ battery • MAX (Battery charging) – 350mA @ 12 V
Environmental conditions	• Operating temperature –20˚C to 70˚C (battery charging from 0˚C to 45˚C) • Storage temperature –40˚C to 80˚C (excluding battery – prolonged storage –20˚C to 25˚C) • Operating humidity up to 95% (non-condensing) • Increased Impact resistance • High resistance to vibration • Size (LDW): 77 x 70 x 27.25 mm
Additional information	• Memory can store 10000-40000 data sets, it depends on how many parameters it collects.

2.2. Technical details GLP 1.5

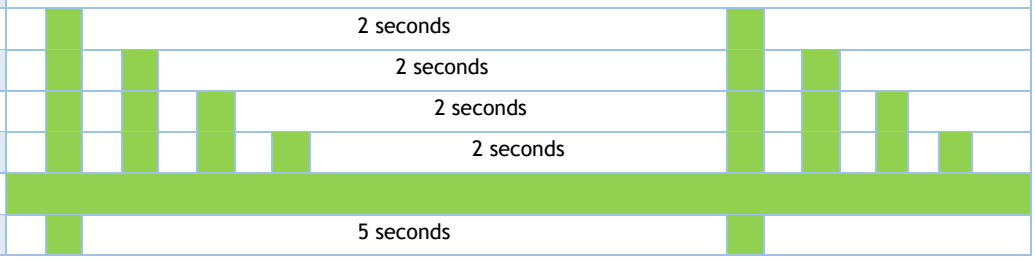
Element	Specification
Interface	• Power supply 9 – 36 V • 1x Analog input, 0 – 36V, 12bit, 1600Hz, averaging of 1024 samples • 2x Analog inputs (optional) 0 - 36V, 12bit, 1600Hz, averaging of 1024 samples shared with RS interface* • 5x Digital inputs • RS232 (standard) or RS485 A/B interface (optional) • CAN-H, CAN-L (SAE J1939) with OBD2 connector • 1x Digital output, open collector, max 2A • 1x Digital output (optional) , open collector, max 2A** • 1x OneWire • 1x pulse input, tested up to 20kH, shared with DIN1 • Dedicated SafeCAN connector (CAN-H, CAN-L) SAE J1939 • 5V output for SafeCan or OneWire devices • Encrypted (AES256) wireless connection on 869.525Mhz (Sub Ghz RF module) • LED status • SIM card slot • Antenna connectors SMA female outer shell, female inner pin * Either two additional inputs or RS interface ** Either one additional output or oneWire
GSM module (2G)	• Quad-band 850/900/1800/1900 Mhz • GPRS class 8 + 10 • The transmitter power: - 2W @ 850/900 Mhz - 1W @ 1800/1900MHZ • Built-in omnidirectional antenna or external antenna
GSM module (3G) optional	• Dual-Band UMTS/HSDPA 850/1900MHz or UMTS/HSDPA 900/2100MHz • Quad-band 850/900/1800/1900 Mhz • GPRS and EDGE class 12 • The transmitter power: - 0.25W @ UMTS 850/900/1900/2100 Mhz - 2W @ GSM 850/GSM900 Mhz - 1W @ DSC 1800/PCS1800 Mhz • External antenna
GSM module (LTE) optional	• LTE Cat M1/Cat NB1/EGPRS module maximum data rate of 300Kbps downlink and 375Kbps uplink • Cat M1/Cat NB1: • LTE FDD: B1/B2/B3/B4/B5/B8/B12/B13/B18/ • B19/B20/B26/B28 • LTE TDD: B39 (For Cat M1 Only) • EGPRS: • 850/900/1800/1900MHz • The transmitter power: max. power: 23dBm Built-in omnidirectional antenna or external antenna
GPS module	• 50 channels • Hot start < 1sec • Cold start < 27sec • Horizontal position accuracy < 2.5m (CEP, 50%) • Receiver sensitivity -161dBm • Built-in active antenna or external active antenna
Bluetooth module optional	• 4800 to 1382400bps communication speed • Current: 40mA max. • Output Power: -4 to +6dbm Class 2 • Range: up to 20m • PIN-secured communication • Ability to set PIN • Ability to set signature
Power consumption	• Sleep mode – 7.3mA @ 12V, 3mA @ battery • Normal mode - 70mA @ 12V, 90mA @ battery • Sending data - 250mA @ 12V, 350mA @ battery • MAX (Battery charging) – 350mA @ 12 V

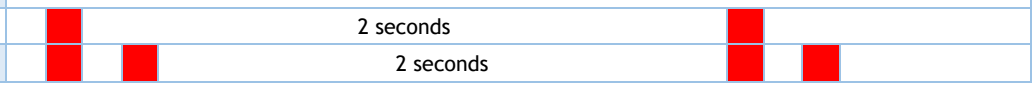
Environmental conditions	• Operating temperature -20˚C to 55˚C • Storage temperature -40˚C to 60˚C • Operating humidity up to 95% (non-condensing) • Enclosure protection rating IP54 • Increased Impact resistance • High resistance to vibration • Size (LDW): 136 x 80.15 x 18 mm, SMA GPS or GSM antenna connector +6.8 mm in length.
Additional information	• Memory can store 10000-20000 positions, it depends on how many parameters it collect

2.3. LED indicator GLP 1.5

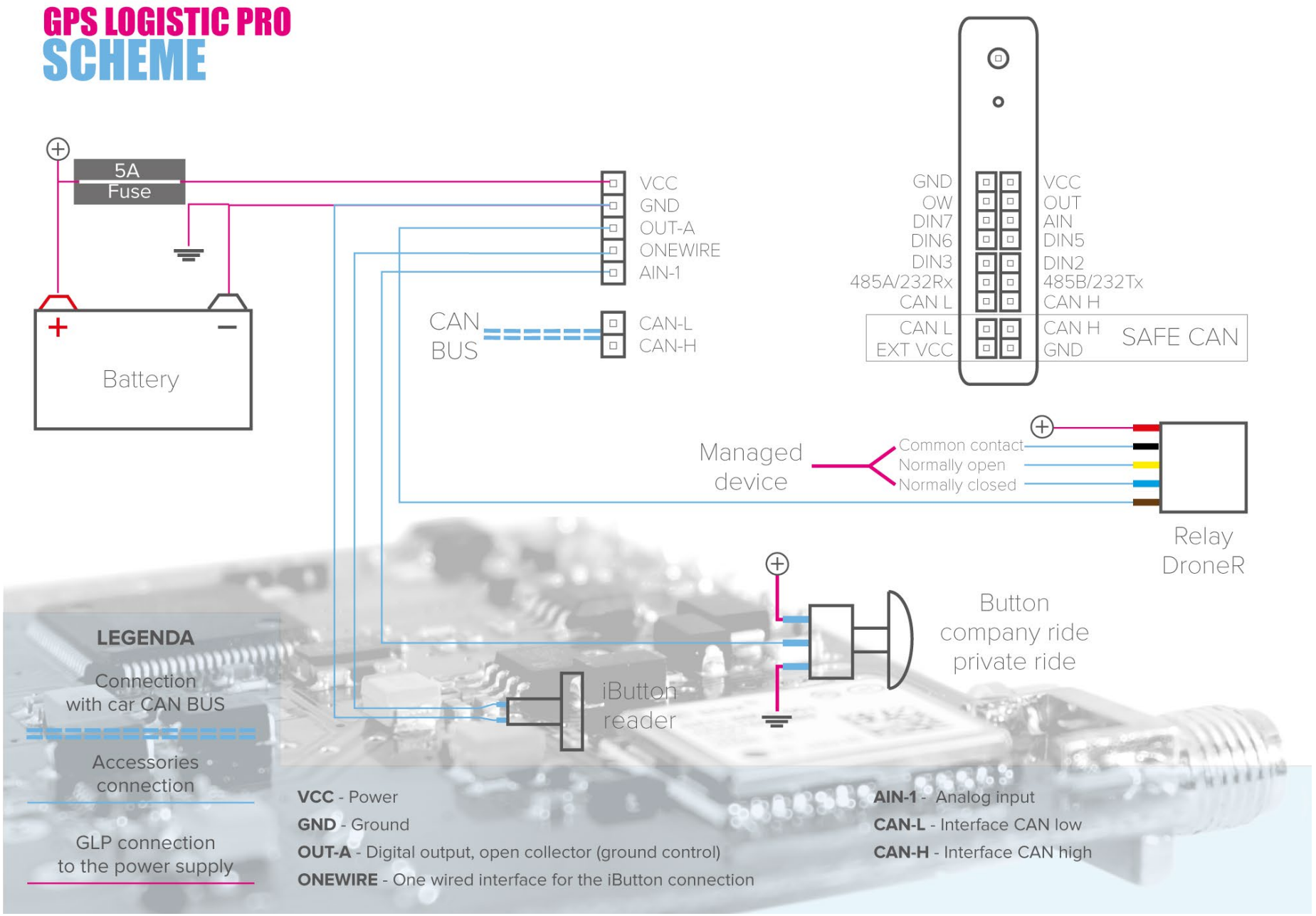
Mode	Blinking
OFF	
Normal	
GPS problem	
GSM problem	
GPS & GSM problem	
Sending data	
Sleep	

2.4. LED indicator GLP 2.0

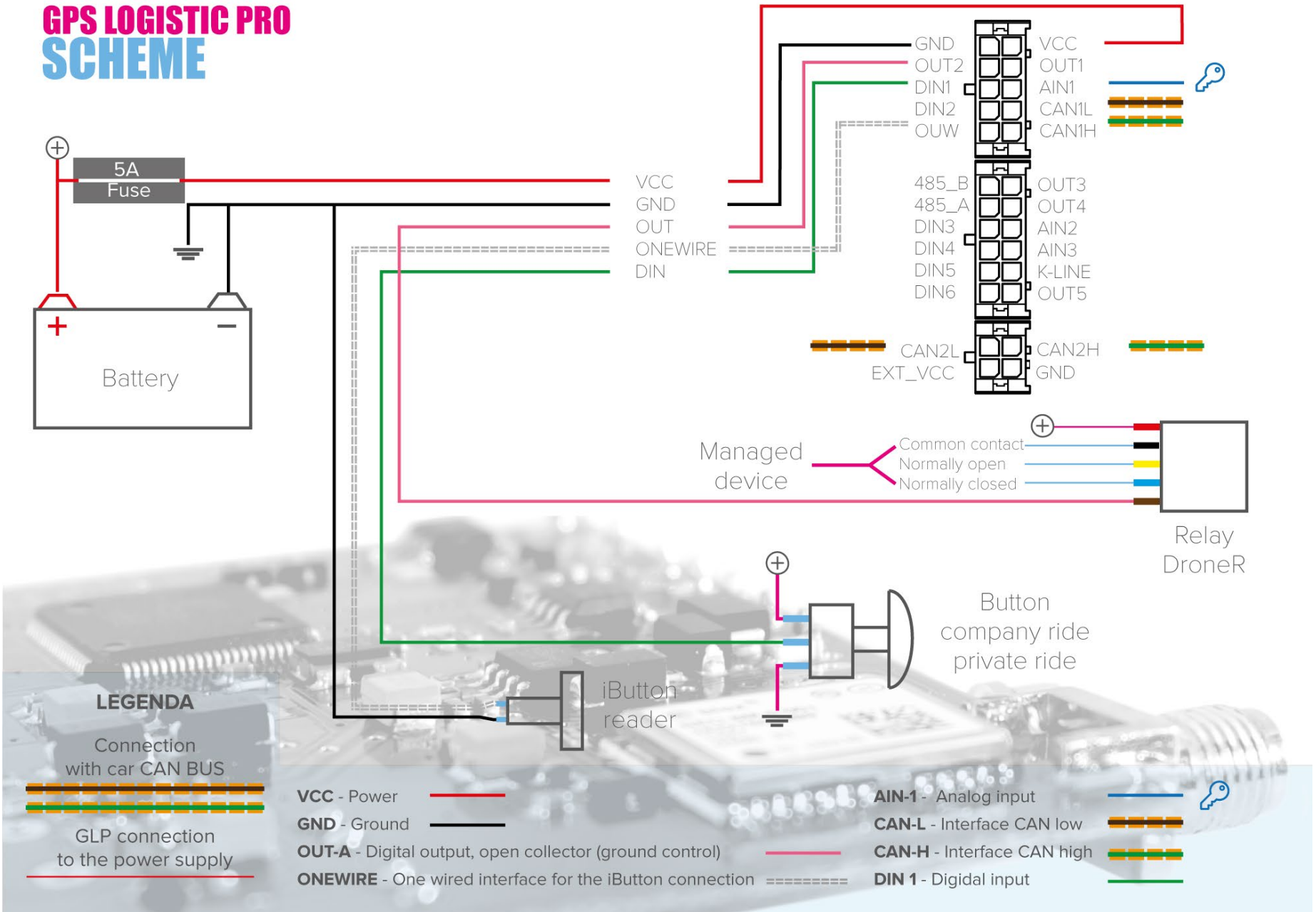
Mode	Blinking
OFF	
Normal	
GPS problem	
GSM problem	
GPS & GSM problem	
Sending data	
Sleep	

Mode	Blinking
OFF	
Normal	
CAN problem	

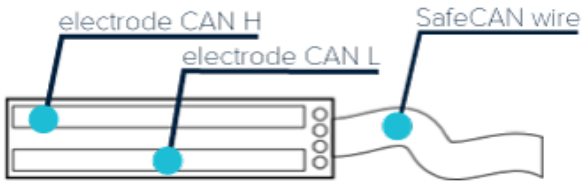
2.4 Installation guidance GLP 1.5



2.5 Installation guidance GLP 2.0



SafeCAN



SafeCANs were modified. There is one additional wire which need to be connected to GND.

IMPORTANT:

Firstly always connect the VCC and GND main beam (power), then SafeCAN - never the other way around. Connection contactless converter (SafeCAN) without main beam will leads to device damage during the first or re-connection of power. One of symptoms of injury is quickly overheating.

During using these SafeCANs please do not enable SafeCANs (3 wire) options in configurator. GLP configuration have to be the same like without it

2.6 EcoDriving

EcoDriving is a functionality that allows measurement of driver behaviours (harsh acceleration, braking and turning). GLP has implemented it in two different ways of measurement:

- based on GPS,
- based on accelerometer.
- based on accelerometer; counting events only

Accelerometer is standard feature of GLP. In all modes, when measurement reaches the specified threshold value, GLP saves position with true flag on special harsh parameter built in protocol. Additionally it saves reached value in specified by user parameter (or its counter).

On GLP we recommend to use EcoDriving based on accelerometer. Measurement are taken at 25Hz and they are not affected to error caused by poor GPS signal. Cornering works above 30km/h to limit false alarms during parking etc.

To use EcoDriving based on accelerometer data you need to calibrate accelerometer in two steps:

1. Calibration in a car after installation. This calibration is necessary to determine direction of device (installation do not require mounting device on any special position, it requires device to be **fixed permanently in vehicle to a sturdy structure**). Vehicle should stay on a flat surface and should not move. You can start calibration procedure using sms/tcp command ".acccalcar"/"@acccalcar". During the first 10 seconds device determines up/down direction. After the answer: "ACC Calibration in car READY – slowly GO and rapidly STOP on flat surface" you have 60 second to **slowly move** and **rapidly stop** vehicle. It is necessary to **move straight ahead** and then the braking with force much **greater than acceleration.** Value of the force is not important so you can move just 2-3 meters. This determines forward/backward direction. After you stop you have to wait for the answer: "ACC Calibration in car SUCCESS" before you move vehicle again.
2. Autocalibration after installation is done is possible. Please refer to .accauto command.

Only precisely and carefully calibrated device will guarantee the correct measurements. Bold expression are the most important parts of the process!

You can use also EcoDriving based on GPS. There is no need to do any special calibration. That EcoDriving is limited by technology to 1Hz measurment. To prevent false alarms (caused by GPS drift) harsh acceleration and braking work when speed is above 10km/h, cornering works above 30km/h.

Warning! Accelerometer does not work when you are connected to GLP 1.5 using wireless dongle.

2.7 Search for CAN data

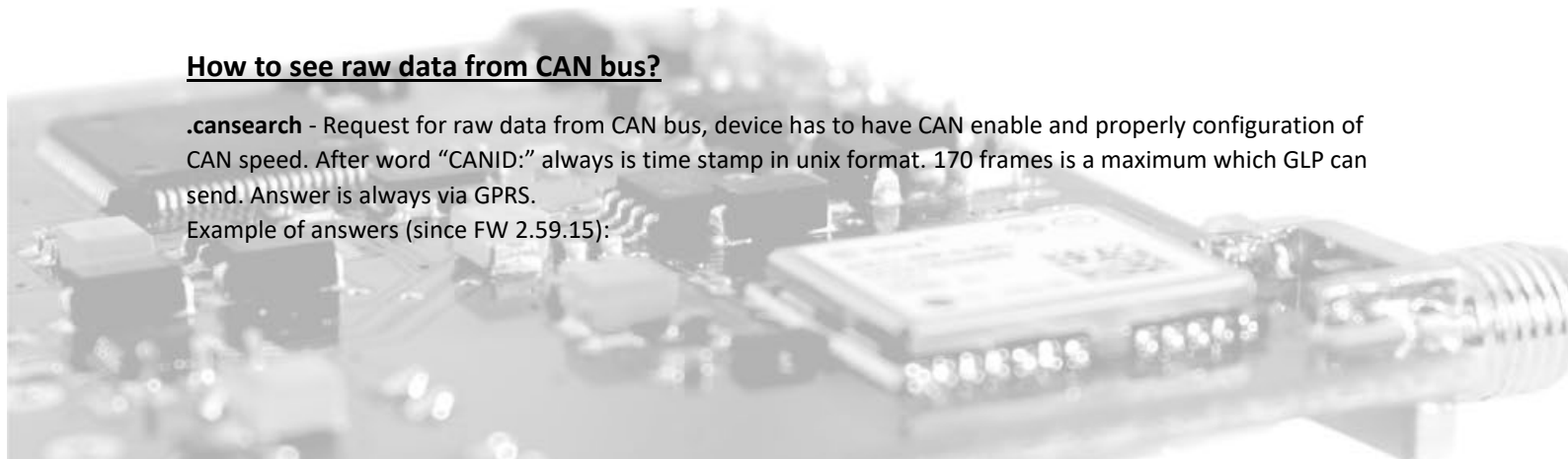
GLP has the ability to scan CAN bus to find interesting data. It provides a few functions to do that.

How to find proper connection?

.testcan/@testcan – Test of CAN connection. Device search data on 8 different speed (46, 50, 62, 83, 100, 125, 250, 500), in the case of success it sends answer with proper speed and number of frames. In case of failure it sends "Not found". 170 frames is a maximum which GLP can count. Waiting for an answer about 2min. Answer via SMS or TCP (./@).

How to see raw data from CAN bus?

.cansearch - Request for raw data from CAN bus, device has to have CAN enable and properly configuration of CAN speed. After word "CANID:" always is time stamp in unix format. 170 frames is a maximum which GLP can send. Answer is always via GPRS.
Example of answers (since FW 2.59.15):



Without data:
CANID:1433831715;

With proper data:
CANID:1433831715;00000101=0100000000000000;00000119=0700000000000000;00000113=4400300000673300;00000123=4400300000673300;

How to find long term information such distance or fuel level?

.fcansearch X - Activates periodic cansearch when ignition is on. X is a time in minutes, if set to 0 periodic cansearch will be disable. After device reset this functionality will be automatically disabled.

It is necessary to collect data from few days. To do that you have to set device to send data to our proxy server and enable periodic cansearch:

.set SERVER_HOST proxy.gpsguardian.pl
.set SERVER_PORT 8889
.fcansearch 1

Next contact with us. Give us information about GLP IMEI, address and port of your server (to forward data), vehicle information (model, year of production, engine) and all information what do you want to get: vehicle distance, motohours, tank capacity etc. We will add device to our server. Proxy will forward data to your server and write it to file that we will analyse.

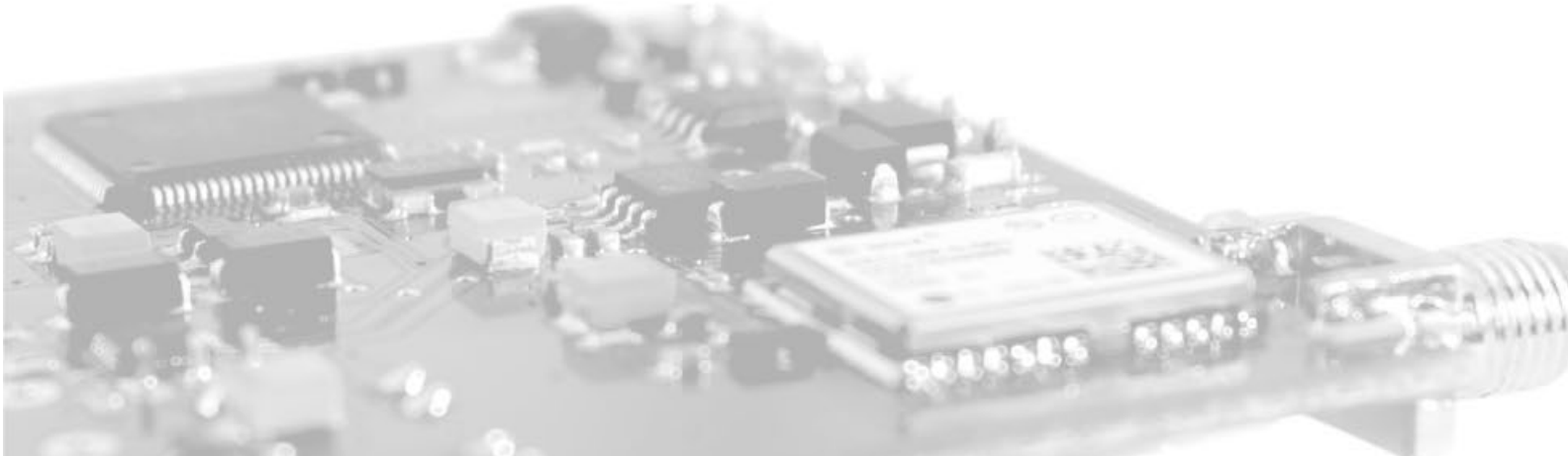
How to find short term information like RPM or position of pedals?

.startcansearch - Enable continuously cansearch. When it is working GLP does not send positions. After device reset this functionality will be automatically disabled. You can disable it using **.stopcansearch** function too.

To do that you have to enable continuously cansearch. Next you have to do some states in vehicle, for example, push acceleration pedal in 25%, next in 50%, 75% and 100% or open doors, close it, etc. First and last state have to be without any event (only ignition on, engine off). **All the time we need only one change** in the vehicle, for example only clutch pedal or only accelerator pedal, not both in the same time. Each state have to be constant, take at least a 60 seconds and each state have to be write down with precise time. **You have to write what you do and when you do it!** Next you have to copy all data from device terminal and send to us with your notes about states. We will analyse it.

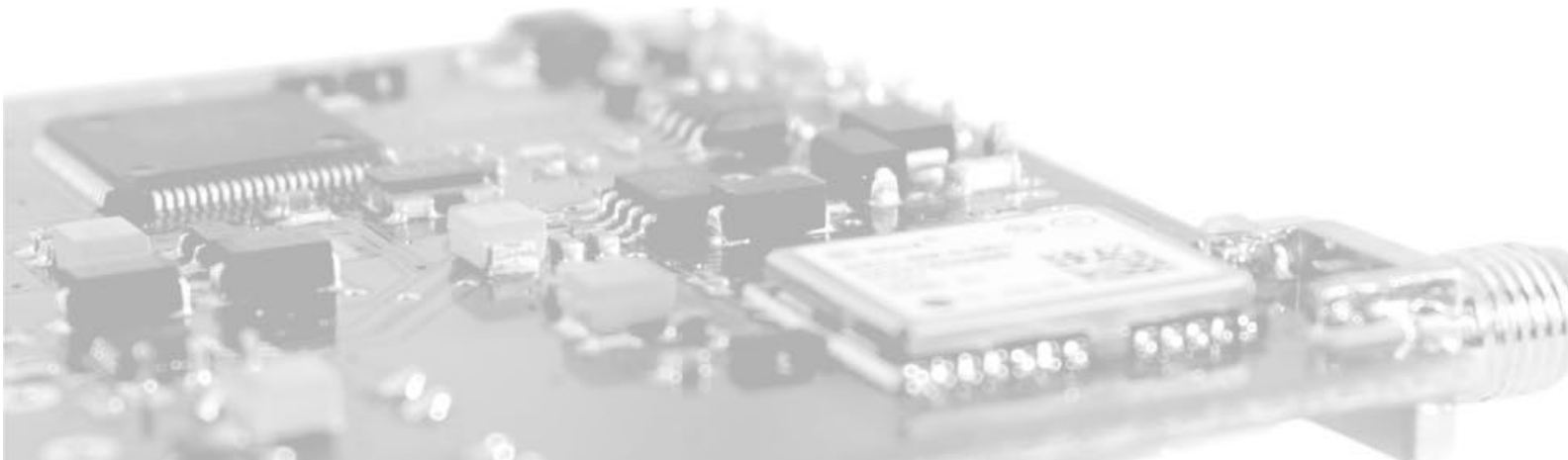
Example of searching state of clutch and position of acceleration pedal (not RPM):

- .startcansearch
- State 1: engine off, ignition on, write down time, wait a 60 seconds.
- State 2: engine off, ignition on, clutch pedal pressed, write down time, wait a 60 seconds.
- State 3: engine off, ignition on, accelerator pedal pressed about 30%, write down time, wait a 60 seconds.
- State 4: engine off, ignition on, accelerator pedal pressed about 60%, write down time, wait a 60 seconds.
- State 5: engine off, ignition on, accelerator pedal pressed 100%, write down time, wait a 60 seconds.
- State 6: engine off, ignition on, write down time, wait a 60 seconds.
- Copy all data from terminal
- .stopcansearch
- Send email to us with all information described above



2.8 Bluetooth interface for technical support (optional)

This functionality enables wireless communication between the device GLP, and the external device (support) without the need for additional service cable. Wireless Bluetooth connection gives you the same functionality as connection with using a service cable. Wireless connection is protected by a PIN whose use is required when establishing a connection. Default PIN is **"0000"**. User may change the PIN after establishing a wireless connection. If you have forgotten your PIN and is impossible to establish a wireless connection, PIN can be reset using the appropriate commands sent by SMS (SMS: .rstbtpin). This command will be handled properly only if it has been sent from the service number.



3 Commands and variables

Commands (in text form) may be sent via SMS or TCP channel. It is possible to send a command as an answer for data delivered to server by using PUT method sent from server side.

Each command must be ended with CRLF "[0x0D][0x0A]" (does not apply for SMS commands), and started with dot or at sign, for example:

TCP: *.systemreset[0x0D][0x0A]*

SMS: *.systemreset*

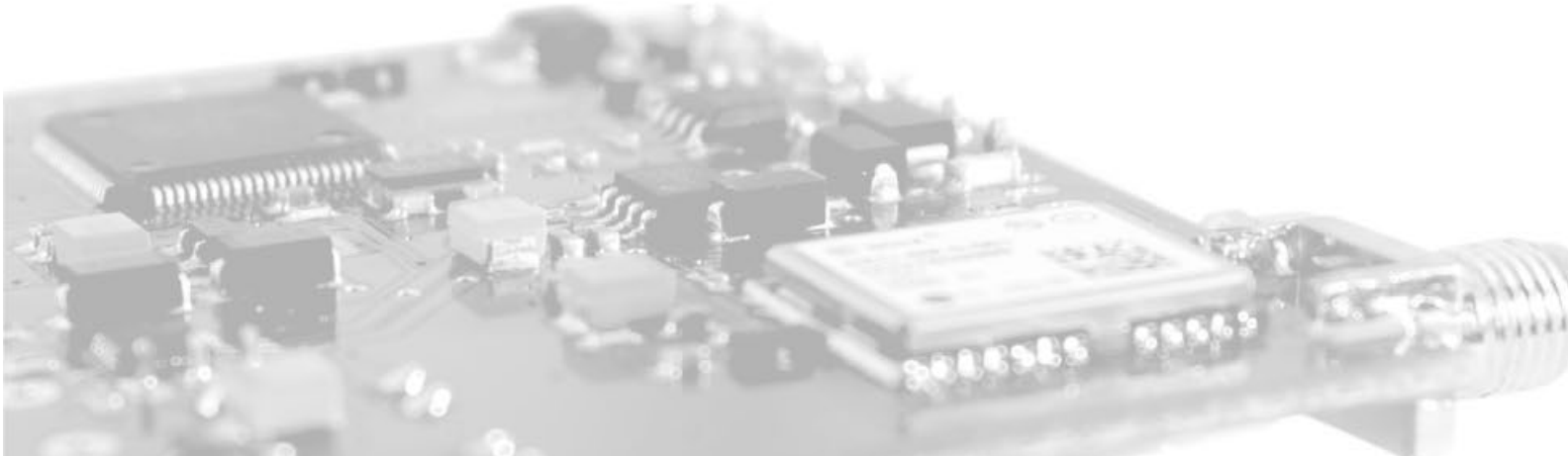
3.4 Telephone lock level

There are 3 levels of telephone lock. These levels are controlled by FLAG_TEL variable and described in first table. There are also 3 types of commands, listed in last table.

Tel lock level	0			1			2		
SMS from tel	service	user	other	service	user	other	service	user	other
Non lockable commands	OK	OK	OK	OK	OK	OK	OK	OK	OK
User commands	OK	OK	OK	OK	OK	OK	OK	OK	XL
All commands	OK	OK	OK	OK	X	X	OK	X	X

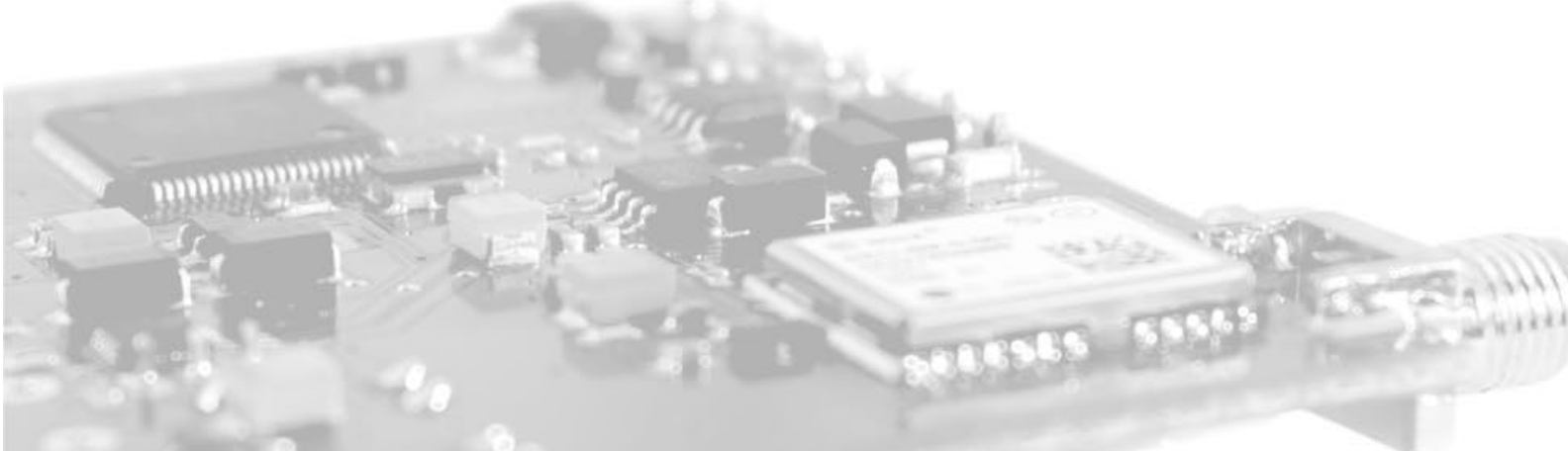
OK	available
XL	available when LIMIT_TEL more than 0
X	unavailable

All Commands	User commands	Non lockable commands
...	.FLAG_TEL_RESET	.FLAG_TEL_RESET
	.output on	
	.output off	
	.testcan	
	.test	
	.getrap	
	.acccaltab	
	.acccalcar	



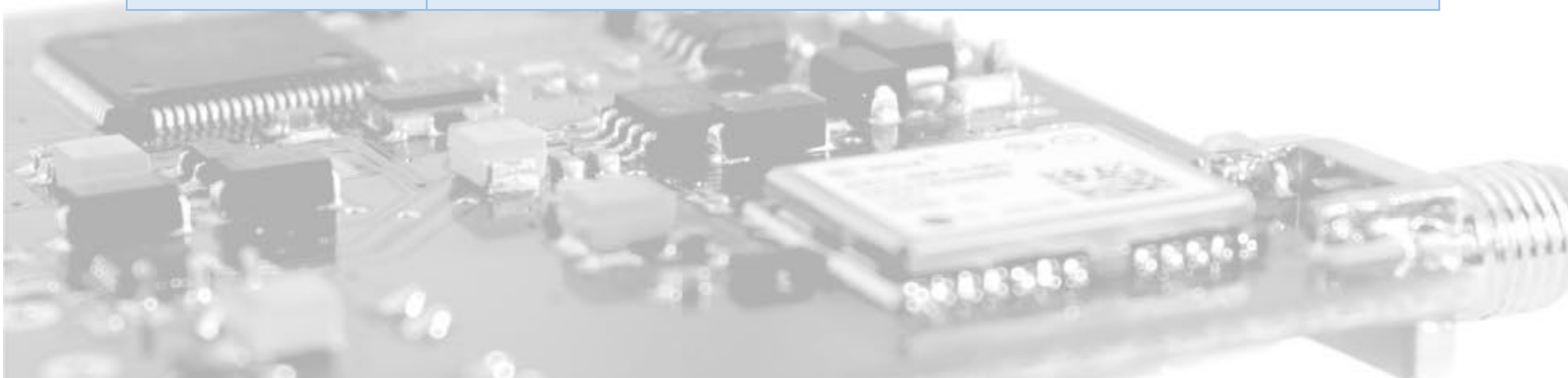
3.5 Executable commands

Syntax	Description
`.FLAG_TEL_RESET XX`	Reset FLAG_TEL to 0 where XX is GLP password. After reset GLP will send confirmation SMS. **Example:** *.FLAG_TEL_RESET OPBEYWLS*
`.get X`	Read variable (answer will be sent via SMS to number from which was the request) where X is a variable. **Example:** *.get version*
`@get X`	Read variable (answer will be sent via GPRS) where X is a variable. **Example:** *@get version*
`.set X Y`	Set variable where X is a variable and Y new value. **Example:** *.set SERVICE_TEL +48123456789*
`.setc X Y`	Set variable with SMS confirmation (answer will be sent via SMS to number from which was the request) where X is a variable and Y new value. **Example:** *.setc SERVICE_TEL +48123456789*
`.savepos`	Forced position log. **Example:** *.savepos*
`.sendpos`	Forced update of all logs held in buffer. **Example:** *.sendpos*
`.getrap`	Request SMS report. Content of the report is defined by REPORT_MESSAGE variable. Report will be sent to a number from which was the request or to SERVICE_TEL when TCP request. **Example:** *.getrap*
`.getops`	Get current and available operator codes. Waiting for an answer about 1min. Answer via SMS. **Example:** *.getops*
`@getops`	Functions like .getops, answer via GPRS. **Example:** *@test*
`.changeops X`	Command to force a change of operator where X is an operator. This is one-time function, GLP do not save this change. **Example:** *.changeops 26001*
`.systemreset`	Reset CPU. **Example:** *.systemreset*
`.reset`	Reset GSM module. **Example:** *.reset*
`.hardreset`	Reset CPU. No data saving **Example:** *.hardreset*
`.gpsreset`	Reset GPS module. **Example:** *.gpsreset*
`.pincfg X[,Y,Z]`	Enable/disable/change PIN on SIM card, where X - action select: 0 - disable 1 - enable 2 - change Y - when X is 0 or 1 you can omit it or set current PIN when X is 2, it is current or new PIN Z - when X is 0 or 1 always omit it when X is 2 and you omit Z, Y is new PIN when X is 2 and you set Z, Y is current PIN and Z is new PIN Answer is always via SMS. **Example:** *.pincfg 1* **Example:** *.pincfg 1,1234* **Example:** *.pincfg 2,1111* **Example:** *.pincfg 2,1234,1111*

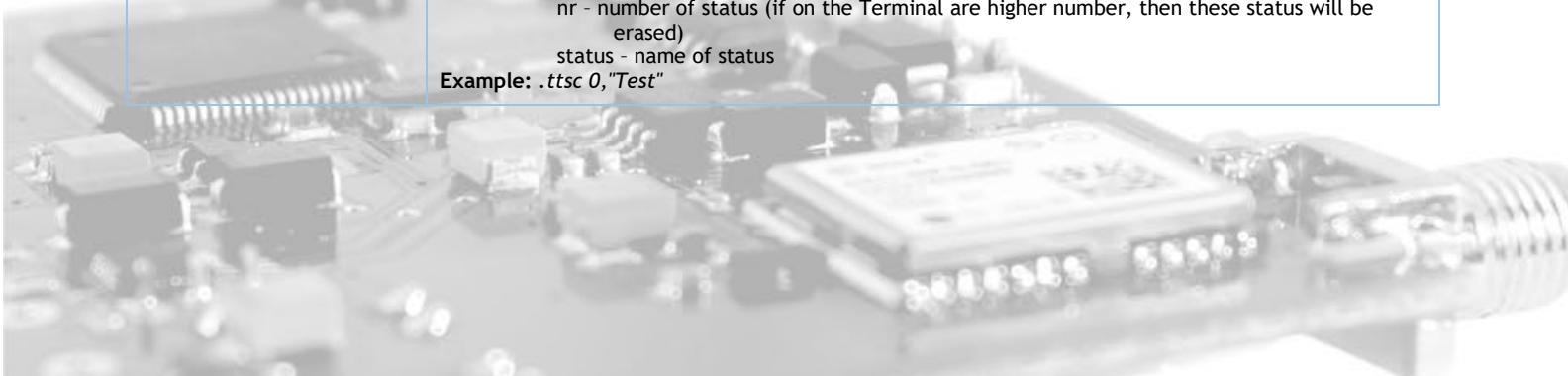


.test	Test of basic functions of device. Answer via SMS. **Example answer:** *GPS=5; GSM=23; GPRS=1; BAT=98%; IGN=0; INPUTS=0001; VCC=12661mV; ADC=26mV; FUEL=0; DIS=0; RPM=0* ***Where:*** GPS – signal strength (0-5) depends on the HDOP: 0 – no fix 1 – more or equal 5 HDOP 2 – more or equal 3 HDOP 3 – more or equal 2 HDOP 4 – more or equal 1 HDOP 5 – less than 1 HDOP GSM – signal strength (CSQ) GPRS – 0 – no TCP connection with server, 1 – TCP connection with server BAT – battery level in % IGN – status of ignition, 0 - ignition off, 1- ignition on INPUTS – status of inputs, 0 – low, 1 - high, input1, inpu2, input3, input4 VCC – supply voltage ADC – analog input voltage FUEL – value from parameter nr 10, default fuel level from CAN DIS - value from parameter nr 11, default distance from CAN RPM - value from parameter nr 12, default RPM from CAN **Example:** *.test* **Warning**: this commands takes up to 60 seconds to execute. Do not send several times if you don't intend specify to execute it several time.
@test	Function like .test, answer via GPRS. **Example:** *@test*
.testcan	Test of CAN connection. Device search data on 8 different speed (46, 50, 62, 83, 100, 125, 250, 500), in the case of success it sends answer with proper speed and number of frames. In case of failure it sends "Not found". 170 frames is a maximum which GLP can count. Waiting for an answer about 2min. Answer via SMS. **Example:** *.testcan*
@testcan	Function like .testcan, answer via GPRS. **Example:** *@testcan*
.testgprs	Test connection to server. The device sends a test query to the server. In the case of success or failure of connecting process, there is an SMS message sent with proper information about the connection. **Example:** *.testgprs*
.testflash	Test whole memory. It take about 3 minutes and answer is send via SMS **Example:** *.testflash*
@testflash	Test whole memory. It take about 3 minutes and answer is send via TCP **Example:** *@testflash*
.accauto / @accauto	Automatically calibrate accelerometer. GLP waits until ignition is OFF and detects full stop. Then first part of calibration process is executed. Next step is started after ignition is ON and speed is above 50 km/h. Based on acceleration change second part of calibration process is executed. Successful calibration requires at least one rapid acceleration and deacceleration (speed up and then brake rapidly). When process finishes with success you will receive "ACC Calibration in car SUCCESS" (SMS or GPRS). If you detect ignition OFF and no message then resend command.
.acccaltab	Calibrate accelerometer on table (step 1). Answer via SMS. More information in "EcoDriving" chapter. **Example:** *.acccaltab*
@acccaltab	Function like .acccaltab, answer via GPRS. **Example:** *@acccaltab*
.acccalcar	Calibrate accelerometer in car (step 2). Answer via SMS. More information in "EcoDriving" chapter. **Example:** *.acccalcar*
@acccalcar	Function like .acccalcar, answer via GPRS. **Example:** *@acccalcar*
.shippingmode	Function forces sleeping independently of configuration. GLP will wake up when voltage rise up to 10V. GLP disable outputs before sleeping. **Example:** *.shippingmode*
.update	Performs download and update of firmware file (SERVICE_UPDATE_FILE) from SERVICE_HOST. After successful download firmware will be installed. **Example:** *.update*

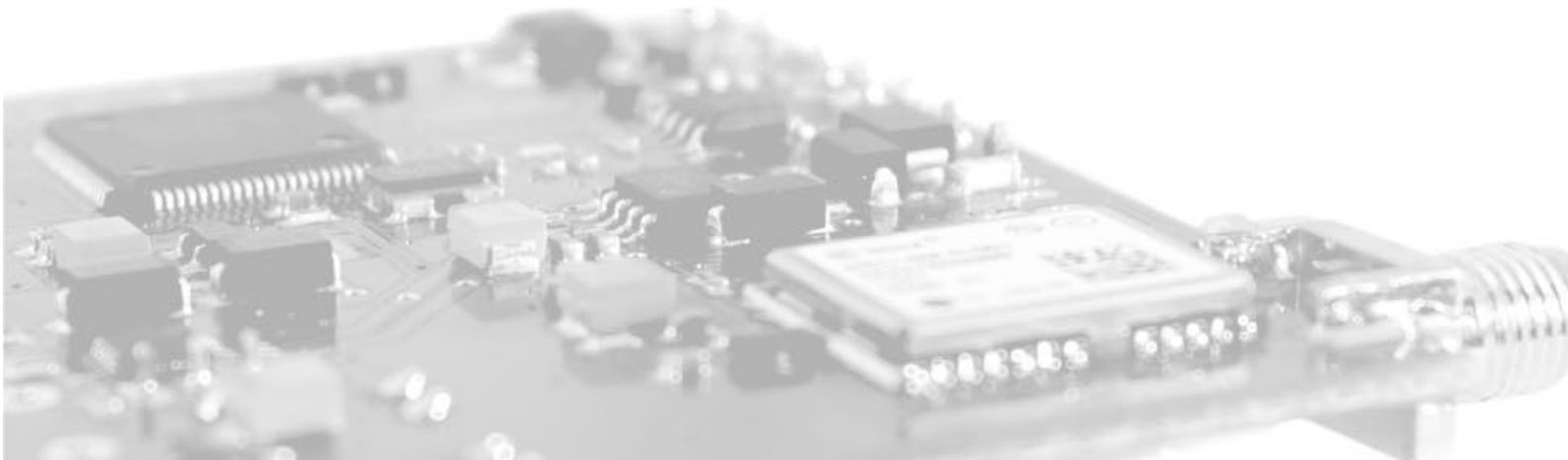
.bootupdate	Performs download and update of bootloader file file (SERVICE_UPDATE_FILE) from SERVICE_HOST. After successful download new bootloader will be installed. Rarely used. Do it with extra care. No confirmation received. Works from firmware 1.2.19 (GLP 2.0 only) **Example:** *.bootupdate*
.dfotaupdate	Performs download and update of GSM module firmware in GLP 2.0. May take up to 5 minutes. Please perform it when device is stationary and CSQ value is above 15. Download consumes about 4MB of data transfer.
.updcmd A,B,C,D	Function uses to set all necessary variables to perform an update, where: A - SERVICE_HOST B - SERVICE_PORT C - SERVICE_UPDATE_FILE D – value 0 to set variables above, 1 to set and perform update **Example:** *.updcmd support.gpsguardian.pl,80,/update/v3update40210.glu,1*
.gprscmd A,B,C,D,E,F,G	Function uses to set all necessary variables to perform an update, where: A - GPRS_APN B – user name for APN C – password for APN D - SERVER_HOST E - SERVER_PORT F - SERVER_TYPE G – value 0 to set variables above, 1 to set, save and reset GSM module **Example:** *.gprscmd "internet",,,track2.gpsguardian.pl,50092,0,1*
.output on *X*	Activates output X where X is number of output (close to ground). **Example:** *.output on 1*
.output off *X*	Deactivates output X where X is number of output (high impedance). **Example:** *.output off 1*
.setodm XXX	Set odometer value where XXX are distance in km. **Example:** *.setodm 1234*
.cansearch	Request for raw data from CAN bus, device has to have CAN enable and properly configuration of CAN speed. 170 frames is a maximum which GLP can send. Answer is always via GPRS. More information in "Search for CAN data" chapter. **Example:** *.cansearch*
@cansearch	The same like .cansearch, answer via GPRS too. More information in "Search for CAN data" chapter. **Example:** *@cansearch*
.fcansearch X	Activates periodic cansearch when ignition is on. X is a time in minutes, if set to 0 periodic cansearch will be disable. After device reset this functionality will be automatically disabled. More information in "Search for CAN data" chapter. **Example:** *.fcansearch 5*
.fcanfast X	The same like .fcansearch, but X is a time in seconds **Example:** *.fcanfast 5*
.startcansearch	Enable continuously cansearch. When it is working GLP does not send positions. After device reset this functionality will be automatically disabled. More information in "Search for CAN data" chapter. **Example:** *.startcansearch*
.stopcansearch	Disable continuously cansearch. **Example:** *.stopcansearch*
.fcanreset X	Resets CAN module periodic, X is a time in minutes. If set to 0 periodic canreset will be disable. **Example:** *.fcanreset 1*
.takepic [x]	Takes picture from camera. In default it takes picture from camera with ID 0. There is possibility to take picture from camera with other ID (necessary when few cameras connected to RS485). **Example:** *.takepic* **Example:** *.takepic 1*
.didget	Request for all saved driver IDs. Answer is always via GPRS. **Example:** *.didget*



.didadd XXX [a0, a1,a2,a3,a4,a5, a6,a7]	Add driver ID where XXX is driver ID and a0…a7 are actions (optional). If ID is present only actions will be replaced. *a0…a7* - actions to do ACTIONS: 0 - none 1 - save data 2 - save data + send via GPRS 3 - save data + send via SMS 4 - save data + send via GPRS + SMS 5 - save data + Call Alarm 6 - save data + send via GPRS + Call Alarm 7 - save data + send via SMS + Call Alarm 8 - save data + send via GPRS + SMS + Call Alarm 9 - enable movement sensor A - disable accelerometer 101…108 - enable output 1…8 201…208 - disable output 1…8 301…308 - toggle output 1…8 400…407 - select filter 0…7 500…503 - do action group **Example:** *.didadd 1234* **Example:** *.didadd 1234 1,101,401*
.diddel XXX	Delete saved driver ID where XXX is driver ID. **Example:** *.diddel 1234*
.defaultcfg	Sets firmware default settings. **Example:** *.defaultcfg*
.getcfg	Request download of the configuration file from the service web server. **Example:** *.getcfg*
.getdidfile	Request download of the Driver IDs file from the service web server. **Example:** *.getdidfile*
.getopsfile	Request download of the Operator Codes file from the service web server. **Example:** *.getopsfile*
.clrbuf	Deletes all buffered position logs. **Example:** *.clrbuf*
.makecall +48xxxxxxxxx	Request to make a call to a given number. **Example:** *.makecall +48123456789*
.tcpclose	Close current GPRS connection temporarily. **Example:** *.tcpclose*
.ignition on	Bypass internal ignition detection and set it as active. **Example:** *.ignition on*
.ignition off	Disable ignition bypass. **Example:** *.ignition off*
.ttc	Request to get terminal status. **Example:** *.ttc*
.tt id,"message" [,"answer1", "answer2",..., "answerN"]	Send Garmin® FMI Message, where: id - message ID (max 64bit) message - message body, max 200 Bytes answer1 - optional, if defined they'll be displayed as possible answers (max 50 chars) Full command can have up to 255 bytes. **Example:** *.tt 59499,"Test","YES","NO"*
.ttp id,latitude, longitude[,"message"]	Send Garmin® FMI localization of dispatch, where: id - Message ID(max 64bit) latitude - position in format +-dd.ddddd longitude - position in format +-ddd.ddddd message - optional message body, max 200 Bytes Full command can have up to 255 bytes. **Example:** *.ttp 113,52.2297,21.0122,"Test"*
.ttsc nr,"status"	Send Garmin® FMI configuration of driver status, where: nr - number of status (if on the Terminal are higher number, then these status will be erased) status - name of status **Example:** *.ttsc 0,"Test"*



.rstbtpin	Reset Bluetooth PIN (for devices with Bluetooth module only!) to default value – "0000". To reset Bluetooth PIN, you must send this command from the service number. Return: 1. *[WARNING!] PIN is not reset. Your phone number is not a service number!*; 2. *[SUCCESS!] PIN has been reset. Default PIN is: "0000"*; 3. [ERROR!] PIN is not reset. Err no.: 0x<EN>", where <EN> is number of Bluetooth module error. Information: If this command returns "[ERROR!] PIN is not reset. Err no.: 0xFE" is possible that you device version hasn't Bluetooth module. **Example:** *.rstbtpin*
.uploadconf	Send all variables to server via GPRS **Example:** *. uploadconf*
.memclr	Full erase of external flash memory. WARNING! Device will lost all configurations. Use on own risk. **Example:** *.memclr*
.idsupdate	Updating IMEI and IMSI in case of error **Example:** *.idsupdate*
.setpar par,val,X	Manualy set parameter value. Par – parameter number to be used Val – value to be saved in DEC format X – sending behaviour 0 or NULL – send based on FILTER settings 1 – sends immediately after saving. **Example:** *.setpar 194,1234* **Example:** *.setpar 194,1234,0* **Example:** *.setpar 194,1234,1*
.backupcfg	Manually saves configuration to memory.

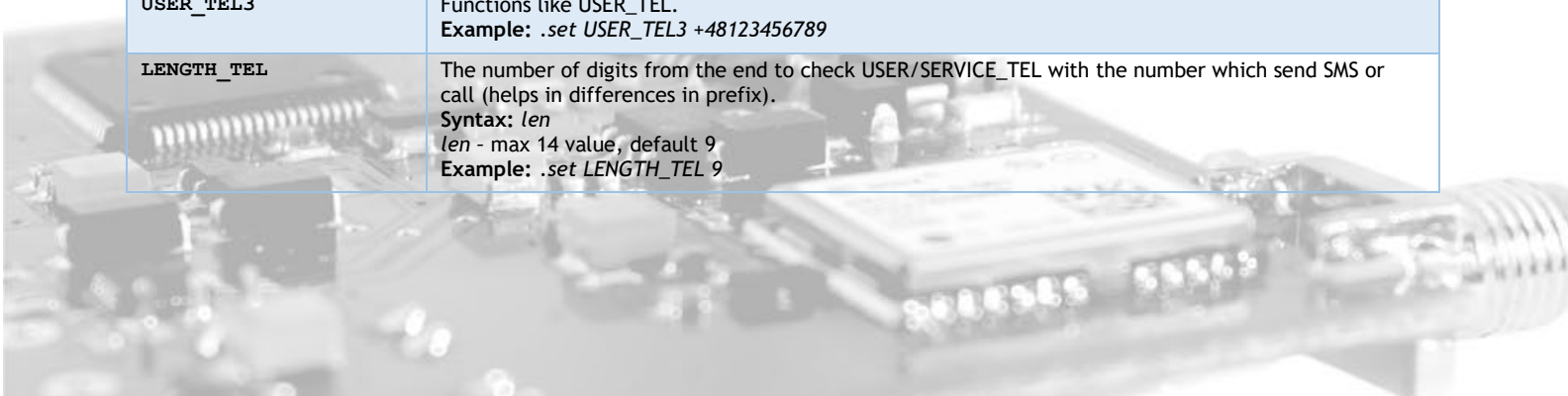


3.6 Read-only variables

Name	Description
version	Firmware version
ismqtt	firmware version type: YES - mqtt enabled NO - mqtt disabled GLP2.0 - standard GLP 2.0
imei	IMEI number
imsi	IMSI number
cbc	Battery information: *state [0-normal/1-charging], level[%], voltage[mV]*
csq	GSM signal strength
creg	Information about the registration status and access technology of the serving cell
su	Quantity of visible GPS satellites
pos	Current geo-position
date	GPS Date
time	GPS Time
pos_count	Quantity of buffered position logs
gsm_model	Gsm module model (for example 900 means SimCom 900)

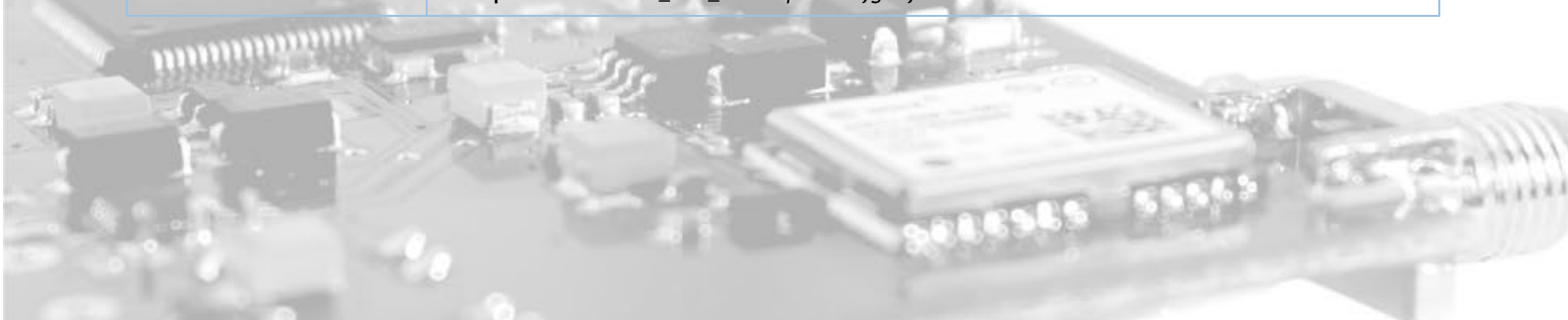
3.7 Configuration variables

Name	Description
DEV_NAME	Device name. May be used to store vehicle registration number. **Syntax:** *name* *Name* - max 49 characters **Example:** *.set DEV_NAME car1234*
SERVICE_TEL	From this number device can get commands and answer the phone after three signals independently of FLAG_TEL. **Syntax:** *tel* *tel* - max 15 characters **Example:** *.set SERVICE_TEL +48123456789*
SERVICE_TEL2	Functions like SERVICE_TEL **Example:** *.set SERVICE_TEL2 +48123456789*
SERVICE_TEL3	Functions like SERVICE_TEL **Example:** *.set SERVICE_TEL3 +48123456789*
USER_TEL	This number will be destination of SMS from event rules. Second function is that device can answer the phone after three signals and can get user commands independently of FLAG_TEL. **Syntax:** *tel* *tel* - max 15 characters **Example:** *.set USER_TEL +48123456789*
USER_TEL2	Functions like USER_TEL. **Example:** *.set USER_TEL2 +48123456789*
USER_TEL3	Functions like USER_TEL. **Example:** *.set USER_TEL3 +48123456789*
LENGTH_TEL	The number of digits from the end to check USER/SERVICE_TEL with the number which send SMS or call (helps in differences in prefix). **Syntax:** *len* *len* - max 14 value, default 9 **Example:** *.set LENGTH_TEL 9*

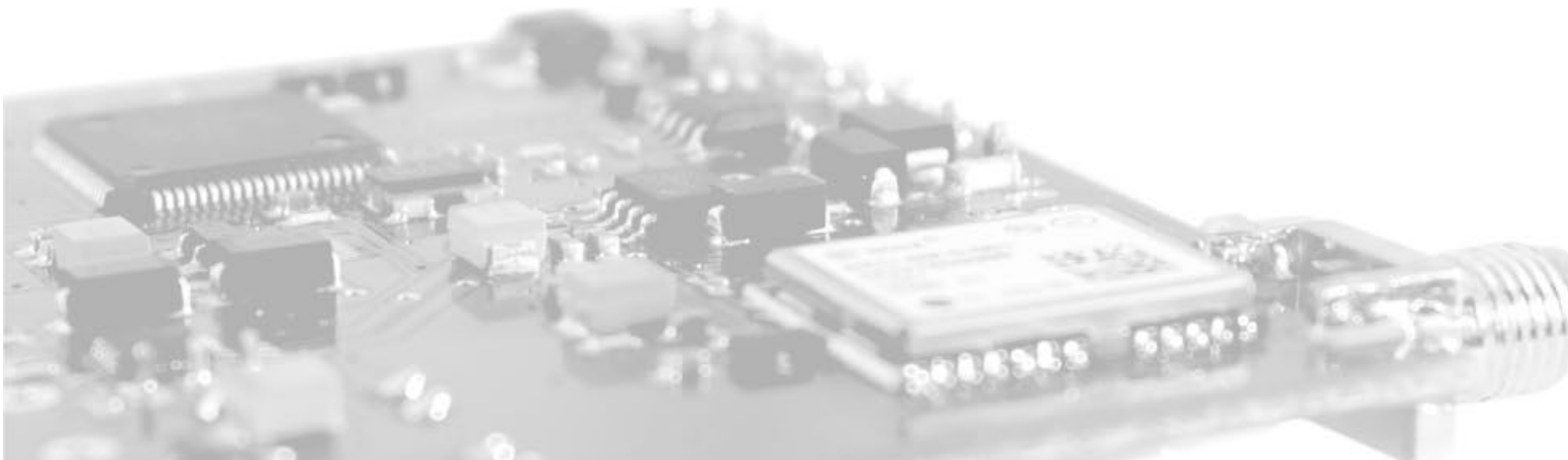


FLAG_TEL	Flag to control telephone lock level. **Syntax:** *ftel* *ftel* - 3 options, default 2, description in chapter "Telephone lock level" **Example:** *.set FLAG_TEL 2*
LIMIT_TEL	This parameter allows to set device from "other" tel number. **Syntax:** lt lt - after each sms with specific function from "other" number, parameter will be decremented by 1; when value is 0 SMS are ignored; max 100 value, default 10 **Example:** .set LIMIT_TEL 10
EVENT_TEL	This parameter control who has to be informed in a case of event occurring. **Syntax:** *etel* *etel* - 4 options, default 1 0 - none 1 - send sms to USER_TEL 1 2 - send sms to USER_TEL 1 and 2 3 - send sms to USER_TEL 1 and 2 and 3 **Example:** *.set EVENT_TEL 1*
FLAG_ISBAR (Deprecated)	Since FW 2.54 FLAG_ISBAR was removed. Lock flag of communication with configurator. **Syntax:** *fisbar* *fisbar* - 2 options, default 1 0 - unloced 1 - locked **Example:** *.set FLAG_ISBAR 1*
UPDATE_SMS	Parameter to disable sending SMS after update. **Syntax:** *usms* *usms* - 2 options, default 1 0 - disabled 1 - enabled **Example:** *.set UPDATE_SMS 0*
RSMBP_CFG	Self RSMBP mac address. **Syntax:** *addr* *addr* - 2 options, default AUTO AUTO - automatic mode MANUAL - 16bit address in hexagonal form (example: 0x01AE) **Example:** *.set RSMBP_CFG AUTO*
REPORT_MESSAGE	SMS report template. **Syntax:** *rmsg* *rmsg* - max 99 characters, possible to set any variable (need to put "%" before variable name) **Example:** *.set REPORT_MESSAGE %DEV_NAME; IMEI:%imei; Bat:%cbc; Csq:%csq; Pc:%pos_count; SU:%su; Pos:%pos*
ACTION_MESSAGE	SMS action template. **Syntax:** *amsg* *amsg* - max 127 characters, possible to set any variable (need to put "%" before variable name) **Example:** *.set ACTION_MESSAGE %DEV_NAME; IMEI:%imei; Action start*
PIN_NUMBER	The PIN number of used SIM card (must be set if used card uses PIN). To enter the pin for the target card, use the card without the PIN number, and after configuration replace it with target card. Other possibility to set PIN is to use configurator. **Syntax:** *pin* *pin* - max 4 characters, default 1234 **Example:** *.set PIN_NUBER 1234*
GPRS_APN	APN GPRS configuration. **Syntax:** *"apn","user","pass"* *apn* - access point name *user* - access point user name *pass* - *access point password* max 99 characters in total **Example:** *.set GPRS_APN "internet","internet","internet"* If there is no user name and password it is possible to send only APN **Example:** *.set GPRS_APN "internet"*
GPRS_DNS	IP addresses of DNS servers. **Syntax:** *"dns1","dns2"* *dns1* - ip address of 1'st dns server (standard format) *dns2* - ip address of 2'nd dns server max 39 characters in total **Example:** *.set GPRS_DNS "153.19.0.50","8.8.8.8"*

GSM_OPS_LIST	Preferred operator code, if this network is in range GSM module connect to it. **Syntax:** *opsc* *opsc* - max 12 characters, operator code consists of two parts: Mobile Country Code (MCC) and Mobile Network Code (MNC) without space between them, blank or 0 set it to full automatic mode. **Example:** *.set GSM_OPS_LIST 26001*
SERVER_HOST	Host name or IP address. **Syntax:** *haddr* *haddr* - max 49 characters **Example:** *.set SERVER_HOST track.gpsguardian.pl*
SERVER_PORT	Host GPRS Port number. **Syntax:** *hport* *hport* - max 6 characters **Example:** *.set SERVER_PORT 50092*
SERVER_TYPE	Type of connection with the server. **Syntax:** *ctype* *ctype* - 5 options, default 0 0 - TCP 1 - UDP 2 - HTTP 3 - TCP + ACK 4 - UDP + ACK **Example:** *.set SERVER_TYPE 0*
SERVER_CLOSE	Defines if connection shall be closed immediately after each upload session. **Syntax:** *cclose* *cclose* - 2 options, default 0 0 - do not close connection 1 - close connection after each update. **Example:** *.set SERVER_CLOSE 0*
SERVER_CODING	Data coding format. **Syntax:** *ccoding* *ccoding* - 2 options, default 0 0 - binary data 1 - ASCII (HEX) coding **Example:** *.set SERVER_CODING 0*
SERVER_PUT_FILE	Path to the put-file. Must be set in case of http protocol usage. **Syntax:** *spf* *spf* - max 49 characters **Example:** *.set SERVER_PUT_FILE /put/put.php*
SERVICE_HOST	Address of service server which hosts firmware updates and configuration files. **Syntax:** *saddr* *saddr* - max 49 characters **Example:** *.set SERVICE_HOST track.gpsguardian.pl*
SERVICE_PORT	GPRS Port number of service server. **Syntax:** *hport* *hport* - max 6 characters **Example:** *.set SERVICE_PORT 80*
SERVICE_UPDATE_FILE	Path to the file with firmware. **Syntax:** *suf* *suf* - max 49 characters **Example:** *.set SERVICE_UPDATE_FILE /update/update.glu*
SERVICE_DID_FILE	Path to the file with driver id. **Syntax:** *sdf* *sdf* - max 49 characters **Example:** *.set SERVICE_DID_FILE /update/did.glu*
SERVICE_OPS_FILE	Path to the file with operator list. **Syntax:** *sof* *sof* - max 49 characters **Example:** *.set SERVICE_OPS_FILE /update/ops.glu*
SERVICE_CFG_FILE	Path to the file with configuration. **Syntax:** *scf* *scf* - max 49 characters **Example:** *.set SERVICE_CFG_FILE /update/cfg.tcf*



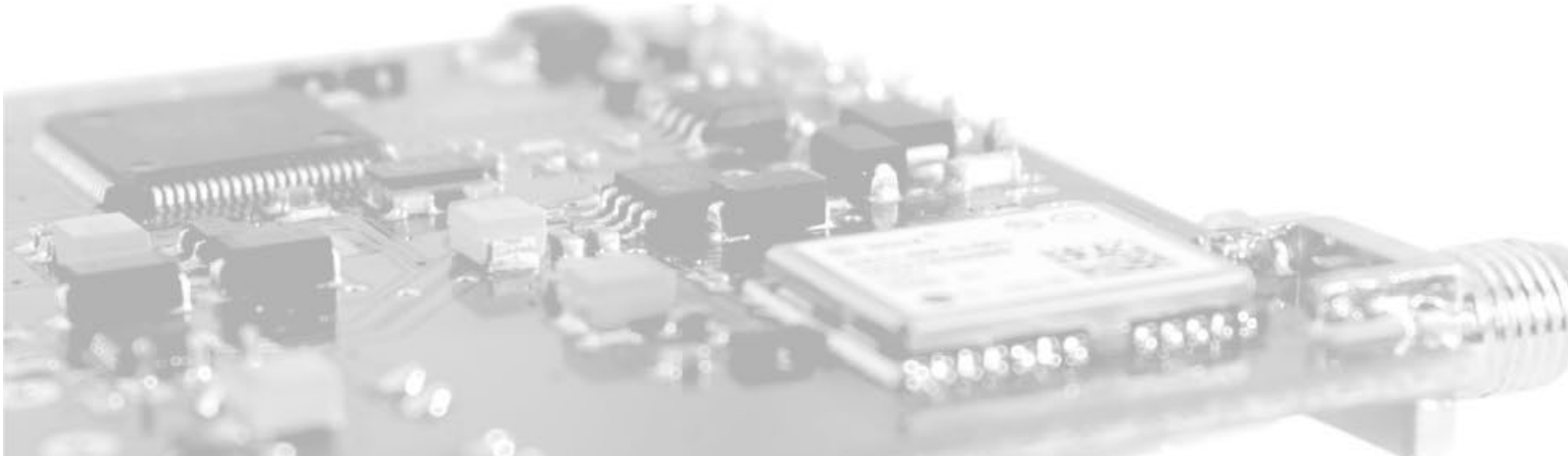
GSM_CFG	Retransmit timeout, interval between clearing errors flag which block gprs sending. **Syntax:** *rt,mq,apn,net* *rt* - max 65534 second, 0 – disabled, default 900 mq – restart timeout related to mqtt error counter flag, default 0 apn – APN mode: 0 (APN used from configuration); 1 – auto APN based on mother network MCC/MNC net – network configuration: 0 (2G only); 1 (auto mode); 2 (LTE CAT.M1 only); 3 (NB-IoT); 4 (CAT.M1 or NB-IoT); default 0 **Example:** *.set GSM_CFG 900,0,0,0*
GCI_CFG	Configuration of parameter numbers for Global Cell ID **Syntax:** *mc,mn,la,ce* *mc* - parameter number for MCC, 0 – not send *mn* - parameter number for MNC, 0 – not send *la* - parameter number for LAC, 0 – not send *ce* - parameter number for Cell Identifier, 0 – not send **Example:** *.set GCI_CFG 128,129,28,29*
PLANE_MODE	Plane mode sets GSM BAND to 1800Mh – low noise but low coverage. **Syntax:** *pm* *pm* - 2 options, default 0 0 – disabled, auto mode 1 – enabled, 1800Mhz only **Example:** *.set PLANE_MODE 1*
GSMMODE	Configuration of parameter number for sending information about 2G/3G mode, 0 – 2G, 1 - 3G (3G version of GLP only). **Syntax:** *gm* *gm* – parameter number **Example:** *.set GSMMODE 27*
DIS_CFG	Configuration of meaning of third parameter in FILTER0…7. Normally third parameter is to save position when speed change, if we set DIS_CFG to 1 position will be saved when distance is reached. **Syntax:** *dis* - 2 options, default 0 0 – disabled (speed) 1 – enabled (distance in meters) **Example:** *.set DIS_CFG 0*



FILTER0 ... FILTER7	Configuration of GPS filters to control the data logging, sending and actions of event rules. **Syntax:** *tp,dg,sp,tm,rp,hd,ts,ig,sl,a0,a1,a,2,a3,id,fm* *tp* - sending mode, 3 options, default 2 0 – do not send 1 – send last position from packed data 2 – normal send *dg* - the threshold of course changes to log position, 0 – disable, default 10° *sp* - the threshold of speed change to log position, 0 – disable, default 20 km/h, if DIS_CFG is 1 this is threshold of distance in meters to log position *tm* - time (in seconds) since last update, beyond which GLP log position when ignition is ON, 0 – disable, default 60 *rp* - quantity of positioning logs which shall be transmitted within single update, 0 – disable sending, default 5 *hd* - limit of GPS signal quality, below this value device will not log data, default 100 *ts* - time (in seconds) since last update, beyond which GLP log position when ignition is OFF, 0 – disable, default 240 *ig* - to control action of ignition changes, default 2 *sl* - to control action of GLP go to sleep, default 2 *a0* - to control action of alarm 0, default 2 *a1* - to control action of alarm 1, default 0 *a2* - to control action of alarm 2, default 0 *a3* - to control action of alarm 3, default 0 *id* - to control action of driver id change, default 2 *fm* - to control action of driver id or driver status from FMI changed, default 2 ACTIONS: 0 – none 1 – save data 2 – save data + send via GPRS 3 – save data + send via SMS 4 – save data + send via GPRS + SMS 5 – save data + Call Alarm 6 – save data + send via GPRS + Call Alarm 7 – save data + send via SMS + Call Alarm 8 – save data + send via GPRS + SMS + Call Alarm 9 – enable movement sensor A – disable accelerometer B - take picture 101…108 – enable output 1…8 201…208 – disable output 1…8 301…308 – toggle output 1…8 400…407 – select filter 0…7 500…503 – do action group max 63 characters in total **Example:** *.set FILTER0 2,10,20,60,5,100,240,2,2,2,0,0,0,2,2*
FILTER_CFG	Current active filter; actions or user can change it. **Syntax:** *fc* *fc* – 8 options from 0 to 7, default 0 **Example:** *.set FILTER_CFG 0*
ACT_GROUP0 ... ACT_GROUP3	Action groups for use like other actions. Do not use “do action group” in this parameter. **Syntax:** *a0,a1,a2,a3,a4,a5,a6,a7* *a0…a7* – actions to do ACTIONS: 0 – none 1 – save data 2 – save data + send via GPRS 3 – save data + send via SMS 4 – save data + send via GPRS + SMS 5 – save data + Call Alarm 6 – save data + send via GPRS + Call Alarm 7 – save data + send via SMS + Call Alarm 8 – save data + send via GPRS + SMS + Call Alarm 9 – enable movement sensor A – disable accelerometer B - take picture 101…108 – enable output 1…8 201…208 – disable output 1…8 301…308 – toggle output 1…8 400…407 – select filter 0…7 **Example:** *.set ACT_GROUP0 1,101,401*

GPS_CFG	Configuration of parameter number for odometer value, High Resolution Speed Flag, State of A-GPS and GPS Jamming detection (parameter number). **Syntax:** *op,hr,ag,ja* *op* - parameter number for odometer, 0 – not send; odometer updated only when speed above 5 km/h *hr* - high resolution speed, 2 options, default 0 0 – disabled, speed in km/h 1 – enabled, speed in 0.01km/h *ag* - A-GPS, 2 options, default 0 0 – disabled 1 – enabled, GLP download data when there is no visible satellites and ignition change to ON *ja* - parameter number for GPS Jamming, 0 – not send, description of reported values **Example:** *.set GPS_CFG 30,1,1,64*
ODOMETER	Odometer value in meters. **Syntax:** *od* *od* - max 2147483647 meters **Example:** *.set ODOMETER 1234000*
IGNITION_CFG	Configuration of ignition signal conditioning. **Syntax:** *it,th1,th2* *it* - type of ignition detection, 2 options, default 0 0 - signal determined by analysis of the supply voltage, then: *th1* - defines voltage in mV, below which the ignition is treated as disabled *th2* - defines voltage in mV, above which the ignition is treated as active 2 - signal determined by the state of declared input, then: *th1* - select the input that determinates ignition *th2* - not used 3 - signal determined by analysis of the battery voltage, then: *th1* - defines voltage in mV, below which the ignition is treated as disabled *th2* - defines voltage in mV, above which the ignition is treated as active 4 - signal determined by analysis of built in accelerometer; TILT_CFG configuration required: example .set TILT_CFG 200,6,2,10 will set up ignition ON when 2 impulses within 10 seconds are higher than 0.2 G (whatever direction); ignition off when no impulses higher than 0.2 G detected within 6 seconds. *th1* - not used *th2* - not used 5 – CANbus data used; proper CAN_IDx configuration required with flag 17; example: .set CAN_ID1 1,KXYZXYZXYZXYZXYZXYZXYZ,17,YY,0 *th1* - not used *th2* - not used **Example:** *.set IGNITION_CFG 0,13300,13700* **Example:** *.set IGNITION_CFG 2,1*
ADC_CFG	Configuration of parameters numbers for analog channels, thresholds and averaging. **Syntax:** *bp,sp,ap,sv,cv,av,hv,av* *bp* - parameter number for battery voltage, 0 – not send *sp* - parameter number for supply voltage, 0 – not send *ap* - parameter number for analog input voltage, 0 – not send *sv* - sleep voltage, threshold for supply voltage (in mV) for event rule below which the GLP will sleep, 0 – not sleep *cv* - threshold for supply voltage for alarms in mV *av* - threshold for analog input for alarms in mV *hv* - hysteresis in mV for threshold voltages av – time in seconds for averaging analog input, 0 - disabled max 31 characters in total **Example:** *.set ADC_CFG 2,3,4,11500,2000,0,200,0*
ADC2_CFG	Configuration of parameters numbers for analog channels, thresholds, averaging. These two analog inputs needs additional elements on PCB. **Syntax:** *p2,p3,t2,t3,a2,a3,v1,in1* *p2* - parameter number for analog 2 input, 0 – not send *p3* - parameter number for analog 3 input, 0 – not send *t2* - threshold for analog 2 input for alarms in mV *t3* - threshold for analog 3 input for alarms in mV *a2* - time in seconds for averaging analog 2 input, 0 - disabled *a3* - time in seconds for averaging analog 3 input, 0 – disabled *t4* - threshold for battery voltage for sleep mode in mV; below device enters sleep mode to protect battery overcharge in1 – inputs configuration; 0 – activated by GND; 1 – activated by VCC signal; convert decimal to binary for example 5 = 0b000101 what means 1st and 3rd input is activated by VCC; default is 0 **Example:** *.set ADC2_CFG 5,6,4000,5000,0,10,3700,0*

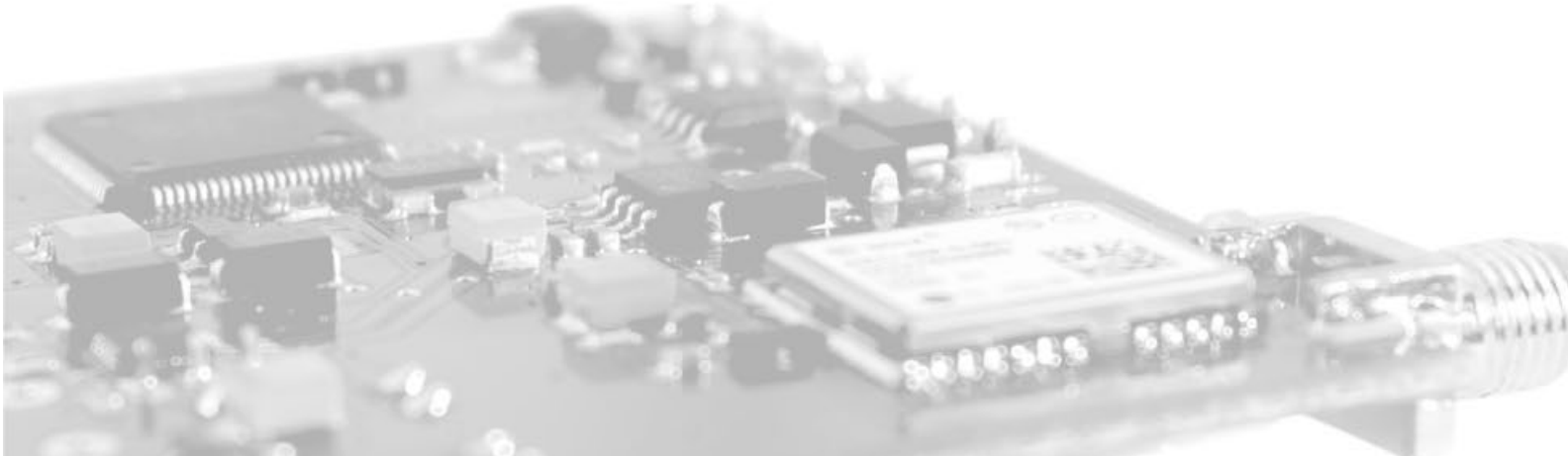
PULSE_CFG	Configuration of pulse measurement. **Syntax:** *mo,pn* *mo* - working mode, 3 options: 0 – disabled 1 – frequency measurement, result in Hz 2 – pulsometer measurement, summing pulses 3 – pulse duty cycle measurement, result in % *pn* - parameter number for pulse measurement, 0 – not send **Example:** *.set PULSE_CFG 1,20*
PULSE_THRESH	Threshold for pulse measurement for alarms. **Syntax:** *th,hy* *th* - threshold for pulse measurement for alarms *hy* - hysteresis for pulse threshold **Example:** *.set PULSE_CFG 1,20*
PULSOMETER	Pulsometer value **Syntax:** *pv* *pv* - max 9223372036854775807 **Example:** *.set PUSLOMETER 1234000*
SLEEP_CFG	Configuration of sleep rules. **Syntax:** *ts,is,ig,sl,a0,tw* *ts* - time to go sleep after sleep conditions are fulfilled, 0 to 86400 second, 0 – disable sleep mode, minium time is 60 second *is* - intelligent sleep, immediately sleep (independently of ts) after done sleep actions from active FILTER, 0/1 - disable/enable, not working when sleep action is a group action *ig* - ignition is sleep condition, 0/1 - disable/enable *sl* - sleep voltage is sleep condition, 0/1 - disable/enable *a0* - alarm 0 is sleep condition, 0/1 - disable/enable *tw* - time to wake up, 1 to 324000 in seconds, 0 – disable **Example:** *.set SLEEP_CFG 300,0,0,1,1,43200*
AGF_CFG	Parameter to configuration of conditions and radius of AutoGeofencing. **Syntax:** *mo,rd* *mo* - condition for start/stop AutoGeofencing , few options: 0 – disabled 100 – ignition low/high 101 – ignition high/low 200 – analog input 0 (supply voltage) low/high 201 – analog input 0 (supply voltage) high/low 210…250 – analog input 1…5 low/high 211…251 - analog input 1…5 high/low 310…340 – digital input 1…4 low/high 311…341 - digital input 1…4 high/low 500 – driver id logout/login all 510 - driver id logout/login from list 700 – frequency low/high 701 – frequency high/low *rd* - radius in meters **Example:** *.set AGF_CFG 100,20*



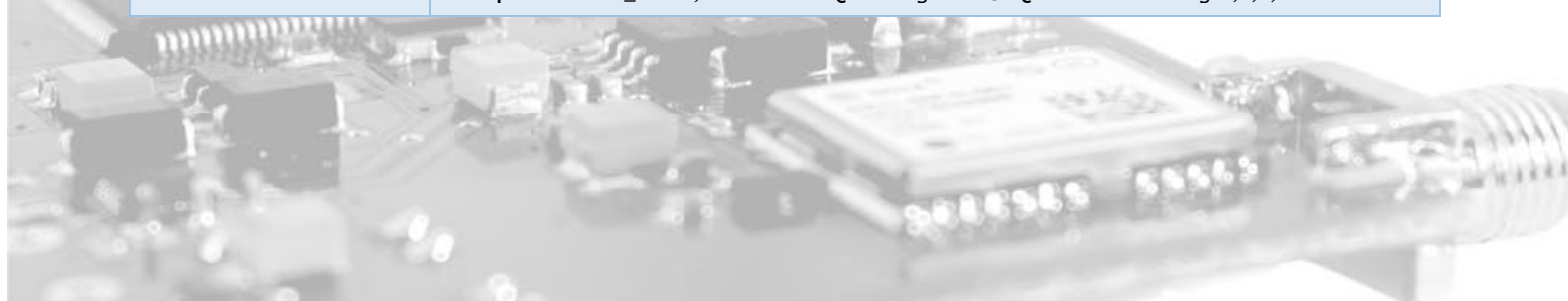
ALARM0 ... ALARM7	Parameters to control event rules. **Syntax:** *a0,a1,a2,a3,a4,a5,a6,a7* *a0* – or/and condition for alarm events, 0/1 *a1…a7* – events to check EVENTS: 0 – none 100 – ignition low state 101 – ignition high state 200 – analog input 0 (supply voltage) low state 201 – analog input 0 (supply voltage) high state 210…230 – analog input 1…3 low state 211…231 - analog input 1…3 high state 240…250 – analog input 1…2 from drone low state 241…251 - analog input 1…2 from drone high state 310…340 – digital input 1…4 low state 311…341 - digital input 1…4 high state 320 - digital input 2 low state 400 - movement 500 – driver id logout 501 - driver id login 511 - driver id login from list 600 – exit from AutoGeofencing 700 – frequency low state 701 – frequency high state 800…870 – CAN data 0…7 low state 801…871 – CAN data 0…7 high state 900 – drone is not available **Example:** *.set ALARM0 0,200*
DEVICE0 ... DEVICE7	Configuration of RSMBP Devices. **Syntax:** *type,addr,d0,d1,d2,d3,d4,d5,d6,d7* *type* **-** device type, 5 options: 0x12 – iGLink, then: *d0*..d7 – not used 0x20 – Drone IO, then: *d0* – analog input 1 *d1* – analog input 2 *d2* – temperature sensor input *d3* – supply voltage *d4* – digital input *d5* – threshold for analog input 1 for alarms *d6* – threshold for analog input 2 for alarms d7 – not used 0x21 – Drone OW, then: *d0* – analog input 1 *d1* – analog input 2 *d2* – temperature sensor input *d3* – supply voltage *d4* – not used *d5* – threshold for analog input 1 for alarms *d6* – threshold for analog input 2 for alarms d7 – not used 0x22 – Drone RS485 (Vepamon), then: *d0* – fuel level *d1* – fuel level 2 *d2* – temperature sensor input *d3* – supply voltage *d4*…d7 – not used 0x30 – iPKey, then: *d0*..d7 – not used *addr* **-** remote device address *d0…d7* – parameter numbers for corresponding measurements max 31 characters in total the variable DEVICE0 must be used at first place. if parameter is set 0(zero) none of its data will be transmitted **Example:** *.set DEVICE0 0x21,0xBE21,5,6,7,8,9,0,0,0*
THERM_CFG	Configuration of parameter number for temperature readings from 1-wire sensor. **Syntax:** *tp* *tp* **-** parameter number, 0 – not send **Example:** *.set THERM_CFG 20*

EXTEN_CFG	Configuration to control working of extension devices. **Syntax:** *type,par* *type* - mode of working/type of device, 4 options: 0 – none/Additional Analog inputs 1 – RS232 Camera 2 – RS485 Camera 3 – TMR Par – additional parameter, actually used only with cameras to set size of picture (depends on camera) **Example:** *.set EXTEN_CFG 0,0* **Example:** *.set EXTEN_CFG 2,4*
ECO_CFG	Configuration of EcoDriving. EcoDriving starts working when it is enabled and ignition changes to ON. When it is on accelerometer mode accelerometer have to be calibrated on table first (step 1) and after installation in car (step 2). More information in "EcoDriving" chapter. **Syntax:** *mo,ac,br,cr,acp,brp,crp,ta,tb,tc* *mo* - working mode, 3 options: 0 – disabled 1 – based on GPS 2 – based on accelerometer; local maximum is reported 3 – based on accelerometer, counters only *ac* - acceleration force sensitivity in 0.001g, 0 to 2000, 2000 = 2g *br* - braking force sensitivity in 0.001g, 0 to 2000, 2000 = 2g *cr* - cornering force sensitivity in 0.001g, 0 to 2000, 2000 = 2g; in GPS mode is in deg/s, 30=30°/s *acp* - parameter number for acceleration value, 0 – not send *brp* - parameter number for braking value, 0 – not send *crp* - parameter number for cornering value, 0 – not send *ta* – acceleration timer (1=100ms); event is reported when acceleration is continuously above threshold for at least ta time; default 0; *tb* - braking timer (1=100ms); event is reported when braking is continuously above threshold for at least tb time; default 0; *tc* - cornering timer (1=100ms); event is reported when cornering is continuously above threshold for at least tc time; default 0 **Example:** *.set ECO_CFG 2,250,300,250,25,26,27,0,0,0*
TILT_CFG	Configuration of movement sensor (based on accelerometer). Movement sensor starts working when its sensitivity is not 0 and when one of things occur: ignition changes to OFF, GLP goes sleep, special event occurs. Last two sub parameters are to set number of movements in period to generate movement event. Sampling are 1 second (max 1 movement per second) and max value are 30. When number is 1 and period is greater than 1 then movement event will be held during this time. **Syntax:** *ts,td,tn,tp* *ts* - movement force sensitivity in 0.001g, 0 to 2000, 0 – disable, 2000 = 2g, default 0 *td* - delay to calibrate it after ignition is OFF, 0 to 9999 second, default 0 *tn* - number of movements in Period to generate Movement Event, 1 to 30, default 1 *tp* - period to count movements, 1 to 30, default 1 **Example:** *.set TILT_CFG 250,15,1,1*
SPEED_CFG	Configuration to control threshold of high speed detection **Syntax:** *sp,ts* *sp - speed threshold, 0 – disable, 120 = 120km/h, default 0* *ts – duration to trigger alarm after exceed sp, default 30s* **Example:** *.set SPEED_CFG 120,30*
ACCIDENT_CFG	Configuration of accident mode (based on accelerometer). Accident mode starts working when its sensitivity is not 0, accelerometer was calibrated on table (step 2) and ignition changes to ON. **Syntax:** *as,aa* *as* - accident force sensitivity in 0.001g, 0 to 2000, 0 – disable, 2000 = 2g, default 0 *aa* - action to do when accident occur ACTIONS: 0 – none 1 – save data 2 – save data + send via GPRS 3 – save data + send via SMS 4 – save data + send via GPRS + SMS 5 – save data + Call Alarm 6 – save data + send via GPRS + Call Alarm 7 – save data + send via SMS + Call Alarm 8 – save data + send via GPRS + SMS + Call Alarm 9 – enable movement sensor A – disable accelerometer 101…108 – enable output 1…8 201…208 – disable output 1…8 301…308 – toggle output 1…8 400…407 – select filter 0…7 500…503 – do action group **Example:** *.set ACCIDENT_CFG 1500,4*

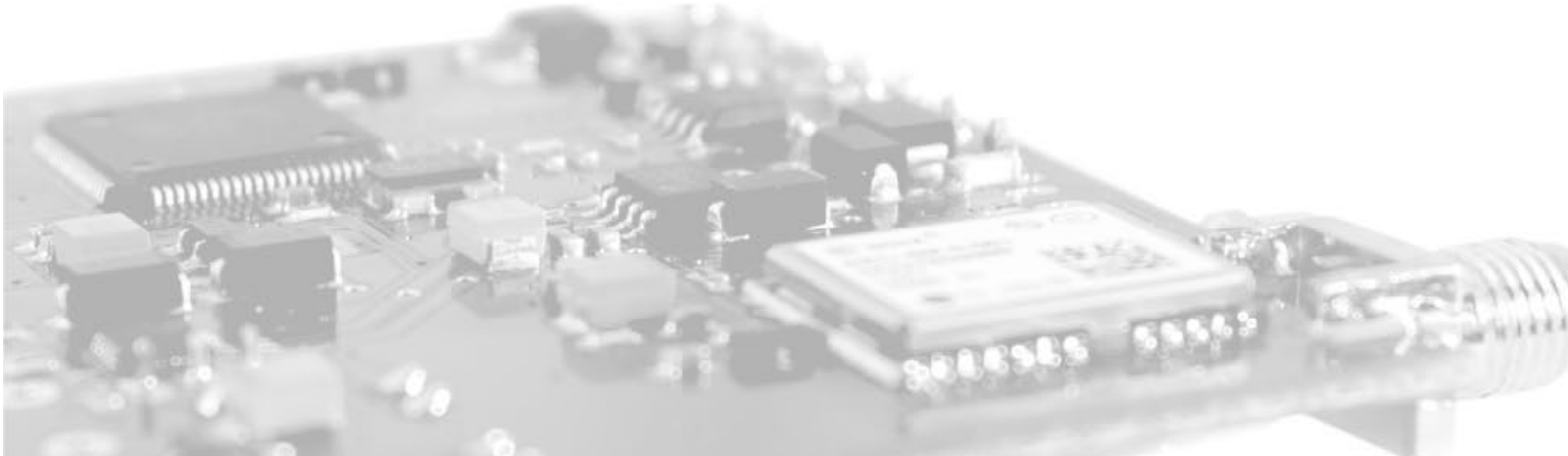
OPS_CFG	Enabling/disabling operator list. Operator groups have priorities (1 – high, unknown – low) and GLP will try connect to operator codes with the highest priority, it will check them with available operators. **Syntax:** *oc* *oc* – 2 options, default 0 0 – disable 1 - enable **Example:** *.set OPS_CFG 0*
CANFUEL	Parameter to hold calculated consumed fuel over CAN Bus. **Example:** *.set CANFUEL 1234*
OPS_GROUP0 ... OPS_GROUP4	Set behaviour if operator code matches element from operator list. **Syntax:** *a0,a1,a2,a3,a4,a5,a6,a7* *a0…a7* – actions to do ACTIONS: 0 – none 1 – save data 2 – save data + send via GPRS 3 – save data + send via SMS 4 – save data + send via GPRS + SMS 5 – save data + Call Alarm 6 – save data + send via GPRS + Call Alarm 7 – save data + send via SMS + Call Alarm 8 – save data + send via GPRS + SMS + Call Alarm 9 – enable movement sensor A – disable accelerometer 101…108 – enable output 1…8 201…208 – disable output 1…8 301…308 – toggle output 1…8 400…407 – select filter 0…7 500…503 – do action group **Example:** *.set OPS_GROUP0 1,101,401*
DKEY_CFG	Configuration of driver IDs service. **Syntax:** *time,exe,ibut,ipky,ifmi,itach,nrids* *time* – time to logout driver key (from iButtons and iPkeys) when ignition is OFF, 20 to 3600 second, FMI ID will logout immediately when Garmin is disconnected, Tacho ID will logout immediately when driver card is ejected Driver ID will logout immediately when GLP goes sleep *exe* – execute common actions for all identifiers, FMI and Tacho always will execute it, 2 options: 0 – disable 1 - enable *ibut* – enable iButtons (turn on OneWire), 2 options: 0 – disable 1 – enable *ipky* – enable iPkeys (requires RSMBP setting), 2 options: 0 – disable 1 - enable *ifmi* – change FMI IDs parameter, 2 options: 0 – standard FMI ID parameter - 240 1 – driver ID parameter - 242 *itach* – enable ID from tachograph cards (requires FMS connection), 2 options: 0 – disable 1 – enable *nrids* – number of IDs from iButtons or iPkeys to send, 2 options: 1 – one 2 – two **Example:** *.set DKEY_CFG 240,0,1,1,1,1,2*



DKEY_LOGON	Actions to do after driver key is logged. **Syntax:** *a0,a1,a2,a3,a4,a5,a6,a7* *a0…a7* – actions to do ACTIONS: 0 – none 1 – save data 2 – save data + send via GPRS 3 – save data + send via SMS 4 – save data + send via GPRS + SMS 5 – save data + Call Alarm 6 – save data + send via GPRS + Call Alarm 7 – save data + send via SMS + Call Alarm 8 – save data + send via GPRS + SMS + Call Alarm 9 – enable movement sensor A – disable accelerometer 101…108 – enable output 1…8 201…208 – disable output 1…8 301…308 – toggle output 1…8 400…407 – select filter 0…7 500…503 – do action group **Example:** *.set DKEY_ LOGON 1,101,401*
DKEY_LOGOFF	Actions to do after driver key is logged out. **Syntax** like DKEY_LOGON **Example:** *.set DKEY_ LOGOFF 1,101,401*
GEO_CFG	Enabling/disabling geofences. **Syntax:** *gc* *gc* – 2 options, default 0 0 – disable 1 - enable **Example:** *.set GEO_CFG 0*
CAN_CFG	Configuration of vehicle CAN bus parameters. **Syntax:** *mode,speed,car* *mode* - CAN working mode, 9 options: 0x1 – standard frames 0x2 – extended frames 0x3 – SAE J1939 (FMS) mode 0x81, 0x82, 0x83 – like above but with old (3-wire) SafeCan 0x41, 0x42, 0x43 – like above but in active mode (use only with additional element) *speed* - interface speed in kbit/s, dependent upon vehicle brand and model *car* - use to display properly car model in configurator, optional **Example:** *.set CAN_CFG 0x3,250*
CAN_ID0 … **CAN_ID7** **CAN_I10** … **CAN_I15**	Configuration of IDs (frames) to be received from vehicle CAN bus and processed by device. 16 frames supported. Bank1: CAN_ID0 … CAN_ID7. Bank2: CAN_I10 … CAN_I15 **Syntax:** *pn,id,d0,d1,d2,d3,mask,a,c,b,d,s1,s2,tr* *pn* - parameter numbers for corresponding CAN frame measurements *id* - ID of the CAN frame (in j1939, PDU format and PDU specific – 0xFEFC; optionally source address - 0xFEFC18) *d0…d3* - indexes of data contained in CAN frame, which will be used to calculate measurement, d0 indicates LSB, and d3 - MSB *mask* - mask used to filter combined value of D3,D2,D1,D0 *a,c,b,d* - adaptation factors where: MEASUREMENT = a/c * DATA + b/d, DATA – Value combined before [(d3,d2,d1,d0) & mask] *s1* – flag to control special functions on CAN parameters, 6 mixed options: 0 or none - none special function 10 or 11 or 12 or 13 or 14 or 15 - enabling zeroing max value (based on mask) 1 or 11 - averaging all data between two saved positions 2 or 12 - averaging exactly s2 numbers of frames 3 or 13 - event mode, on any change on CAN data it execute action s2 4 or 14 - total counter, count buffer overflow 5 or 15 - total counter, count all value *s2* – parameter for special functions *tr* – threshold for CAN data for alarms max 63 characters in total **Example:** *.set CAN_ID1 11,0x000006B5,0,1,2,3,0xFFFFFFFF,1,200,0,1,0,0,0* It is possible to send encrypted data (*id,d0,d1,d2,d3,mask,a,c,b,d)*, such as in the configurator. First and two last parameters have to be send normally. **Example:** *.set CAN_ID0 10,KJDwTbsM4lLQb9mYzRgZPdOz@vQOY4UOWI3w4FcC5g47,0,0,0*



FAULT_CFG	Enabling/disabling reading status of fault lamps and fault codes (DM1 in SAEJ1939-73) They will be send on parameter 244: PL,AWL,RSL,MIL,FAULT for example 0,0,0,0,00000000, where: PL – Protect Lamp, lamp used to relay trouble code information with a vehicle system that is most probably not electronic subsystem related. AWL - Amber Warning Lamp, lamp used to relay trouble code information that is reporting a problem with the vehicle system but the vehicle need not be immediately stopped. Most equipment manufacturers populate this lamp on the dashboard or control panel. RSL - Red Stop Lamp, lamp used to relay trouble code information that is of a severe enough condition that it warrants stopping the vehicle or equipment. Most equipment manufacturers populate this lamp on the dashboard or control panel. MIL - Malfunction Indicator Lamp, lamp used to relay only emissions-related trouble code information. This lamp is illuminated when there is an emission-related trouble code active. FAULT – diagnostic trouble code, this is combine of SPN (Suspect Parameter Number), FMI (Failure Mode Identifier), CM (Conversion Method) and OC (Occurrence Count) lamps can assume 4 states: 0 – off 1 – on 2 – slow flash 3 – fast flash **Syntax:** *fc* *fc* – 3 options, default 0 0 – disable 1 – monitoring (position saved on other events) 2 – save position when fault change 3 – save and send position when fault change **Example:** *.set FAULT_CFG 1*
TOTALCAN0 ... TOTALCAN7	Accumulated value from CAN. **Syntax:** *tc* *tc* - max 4294967295 **Example:** *.set TOTALCAN0 1234000*
SHOW_NAME	Enable or disable parameter 245 – device name. **Syntax:** *en* *en = 1 - enable parameter 245;* *en = 0 - disable parameter 245;* **Example:** *.set SHOW_NAME 1*
BLUETOOTH (optional)	Power supply modes for Bluetooth module. **Syntax:** *mode* *mode = 0 - disable Bluetooth module;* *mode = 1 - enable to first reset / sleep;* *mode = 2 - enable always;* **Example:** *.set BLUETOOTH 1*
RSDEVICE0 … RSDEVICE7	ID of device connected to RS485 interface **Syntax:** *ID* *ID = RS0000 to RSFFFF;* **Example:** *.set* `RSDEVICE0 RSFFFF`
CARSH_CFG	Configures how GLP reacts to car sharing related commands. **Syntax:** *parameter#,mode,doorl ock timeout,status* *parameter# - parameter used to report status (default 71)* *mode* - 2 options: 0 – GLP 2.0 outputs are used (default) 1 – DroneKey is used *door lock timeout* - time in seconds used to automatically trigger door lock commands after ignition off. status – value of reported parameter; future use **Example:** *.set CARSH_CFG 71,0,300,0*



4 Configuration of basic functions

4.4 Unlock the device

The device is default locked for communication via SMS (from non SERVICE_TEL number) and communication with configurator. Communication lock controlled by flag FLAG_TEL. You can disable it by set 0 to correspond parameter if you have TCP connections or if your number is SERVICE_TEL. If not, you should send .FLAG_TEL_RESET with password to GLP to reset FLAG_TEL .

Examples of commands:

.FLAG_TEL_RESET OPBEYWLS

.set FLAG_ISBAR 0

.set FLAG_TEL 0

4.5 Communication with server

To make the device work properly with the server it is necessary to set the following variables, for example:

.set SERVICE_TEL +XXXXXXXXXXX

.set SERVER_HOST www.example.com

.set SERVER_PORT 50092

.set GPRS_APN ”internet”,””,””

.set GPRS_DNS "8.8.8.8","8.8.4.4"

4.6 Update procedure

The device keeps configuration after update.

Update procedure:

1. Before update please contact us in order to receive new GLP password (IMEI will be needed).
2. Set USER_TEL; this number will be automatically copied to the SERVICE_TEL.
   .set USER_TEL +48794700078
3. Set server address, port and path to the update file.
   .set SERVICE_PORT 80
   .set SERVICE_HOST support.gpsguardian.pl
   .set SERVICE_UPDATE_FILE /update/V2update27002.glu
4. Send update command
   .update
   After the update, device will send SMS with confirmation; in case of failure, wait a moment and try again.
5. To unlock getting SMS from non SERVICE_TEL numbers set FLAG_TEL parameter to 0 via TCP or SMS (only from SERVICE_TEL)
   .set FLAG_TEL 0

   In case of blocking device without SERVICE_TEL set and without communication via TCP, the only way to unlock it is using SMS command with GLP password, for example:
   .FLAG_TEL_RESET OPBEYWLS

4.7 Update procedure over USB connection for GLP 2.0

Pre-update procedure

1. Please disconnect main power source from device.
2. Please disconnect backup power source (battery).
3. Please connect device to your computer using cable with mini USB connector.

Update procedure

When you connect GLP2 to your computer over USB cable it will appear as external removable drive.

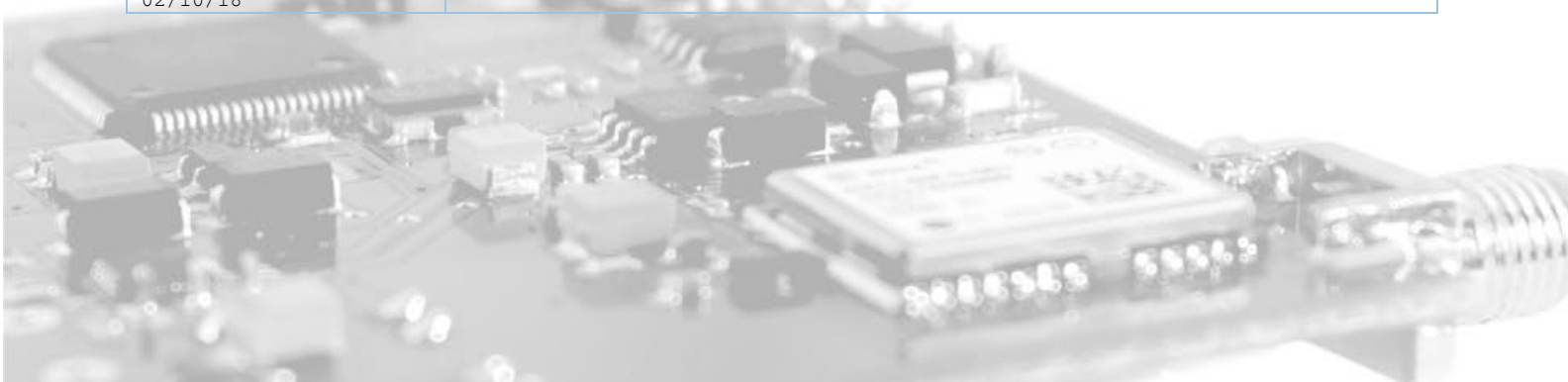
1. Remove any existing file(s).
2. Copy new firmware BIN file.
3. Disconnect device from USB cable.
4. Connect external power source (power up device) and wait for the update.

https://docs.geotik.com/display/GLPDOC/GLP2+firmware+update+using+USB+cable

5 Firmware changelog GLP 2.0

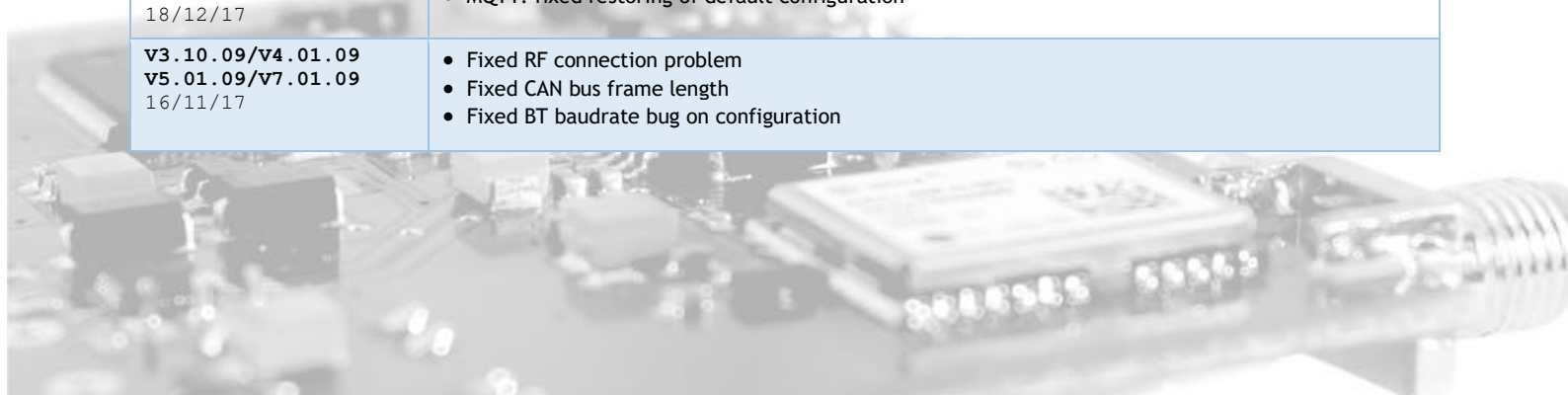
Version	Description
Please follow on-line	http://bit.ly/GLP - changelog
1.02.11 20/09/19	• fixed .gprscmd function
1.02.10 06/09/19	• fixed bugs related with Tester software
1.02.09 20/08/19	• fixed bug in @/.testcan function
1.02.08 15/08/19	• improvement in RF connection with Configurator • recognizing version of mounted accelerometer • optimization in firmware size
1.02.07 18/07/19	• fixed bug in get time from GSM • new command to update device: .updcmd • new command to set communication settings: .gprscmd • new function to reset device by RF connection • fixed bug in sending data function
1.02.06 11/07/19	• improvement in sending historic data from buffer • fixed minor bugs
1.02.05 19/06/19	• new core of system management • fixed bug in saving configuration with Tester software
1.02.03 07/06/19	• fixed minor bugs
1.02.02 30/05/19	• auto selection of preferred band
1.02.1 29/05/19	• device save first data package after power on
1.01.63 02/08/2019	• fixed bugs in GSM module
1.01.62 01/08/2019	• fixed bug in reading IMEI • fixed some bugs in GSM module algorithm
1.01.61 **09/05/19**	• improvement in collecting data from AGPS • fixed some bugs in GSM module algorithm
1.01.60 30/04/19	• fixed some bugs in GSM module algorithm • fixed bugs in LIMIT_TEL and FLAG_TEL • fixed bug in sending big amount of data: e.x. cansearch
1.01.59 24/04/19	• improved SMS receiving • fixed some bugs in GSM module algorithm

1.01.56 04/04/19	• improved .alllogs command • fixed some bugs in GSM module algorithm • fixed bug in sending big amount of data: e.x. cansearch
1.01.55 28/03/19	• MQTT: improved reconnection feature • saving additional logs in memory from GSM module
1.01.54 25/03/19	• saving additional logs in memory from GSM module • improved .lastlogs, more logs available • fixed error in sending logs by .allogs
1.01.53 15/02/19	• fixed bug in writing configuration by Tester
1.01.52 12/02/19	• setting CAN baud automatically after @testcan/.testcan
1.01.51 01/03/19	• new feature: ignition source from CAN
1.01.50 26/02/19	• fixed bug in answering on .set command • sending reason of data acquisition on parameter 67 • added new parameter in .startcansearch function. Sending only differences between CAN data packages
1.01.49 19/02/19	• MQTT: changes in GSM module algorithm • Changed default values to trigger alarms from EcoDriving events
1.01.47 06/02/19	• added additional Alarms • added feature to Action List: enable output on 2 seconds • fixed and implemented changes from beta version
1.01.46 23/01/19	• improvement in restarting GSM module if no active GPRS session • faster powering down GSM module
1.01.45 18/01/19	• beta version: fixed bug related to alarm triggered by CAN data
1.01.44 16/01/19	• beta version: changes in DF7xx RFID reader implementation
1.01.43 14/01/19	• improvement of connection with Drone
1.01.42 11/01/19	• MQTT: allow to send data only to one server • MQTT: next reconnection after 1h when connection error to native server • MQTT: changes in type of variables (IMEI, IMSI, MSISDN) • increase single message size to server from 360 to 480 bytes
1.01.41 09/01/19	• clearing record buffer after 4 unsuccessful attempts • MQTT: improvement in counting passengers connected to Drone • new values in ECO_CFG parameter to trigger events after specified duration
1.01.40 03/01/19	• improvement in GSM module algorithm (DNS settings) • fixed building packages of records • new value in GPS_CFG parameter to restart GPS module at a certain time
1.01.39 13/10/18	• beta version: added alarm after speeding over 120kmph • beta version: improvement of connection with Drone
1.01.38 12/12/18	• beta version: new feature to enable output on 3 seconds • beta version: added alarm after exceeding speed over 20kmph • beta version: new ECO_CFG value to trigger alarm on harsh acceleration
1.01.37 11/12/18	• fixed bug in time collected from GSM module • reduction in project size (new GSM module on older GLP version supported) • MQTT: fixed issue with no server address/port • GSM module algorithm improvements
1.01.36 10/12/18	• improvement in GSM module algorithm (keeping active connection socket, change priority of sending data, fix MCC/MNC reading, SMS answering) • faster data preparing/sending after configuration change
1.01.34 30/10/18	• new GSM module algorithm
1.01.33 30/10/18	• beta version of ASN1 message scheme
1.01.31 08/10/18	• MQTT: tiny improvement of seat sensor counting logic
1.01.30 02/10/18	• MQTT: fixed problem with mixing Events (braking/accelerating and accidents)

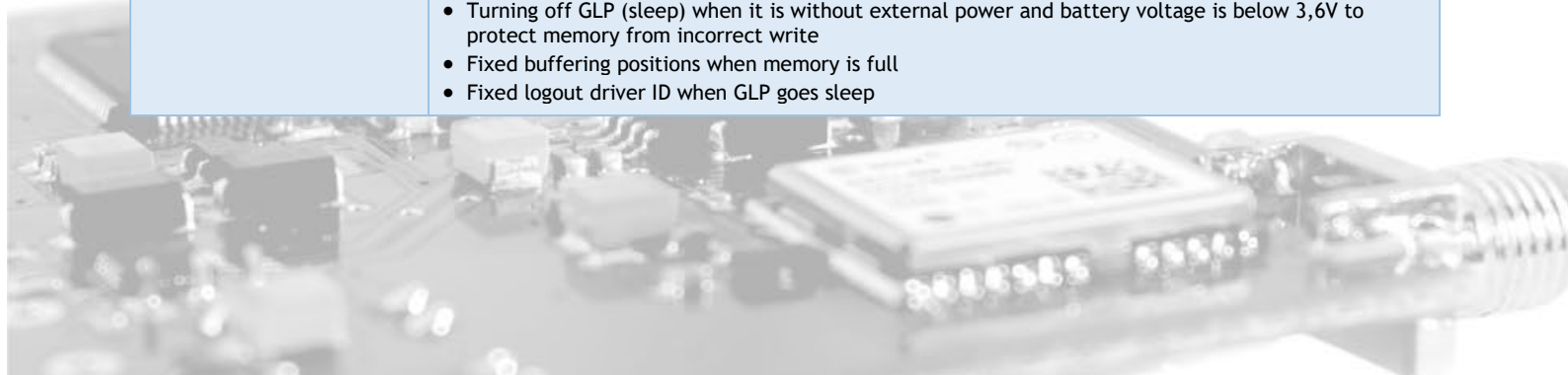


6 Firmware changelog GLP 1.5

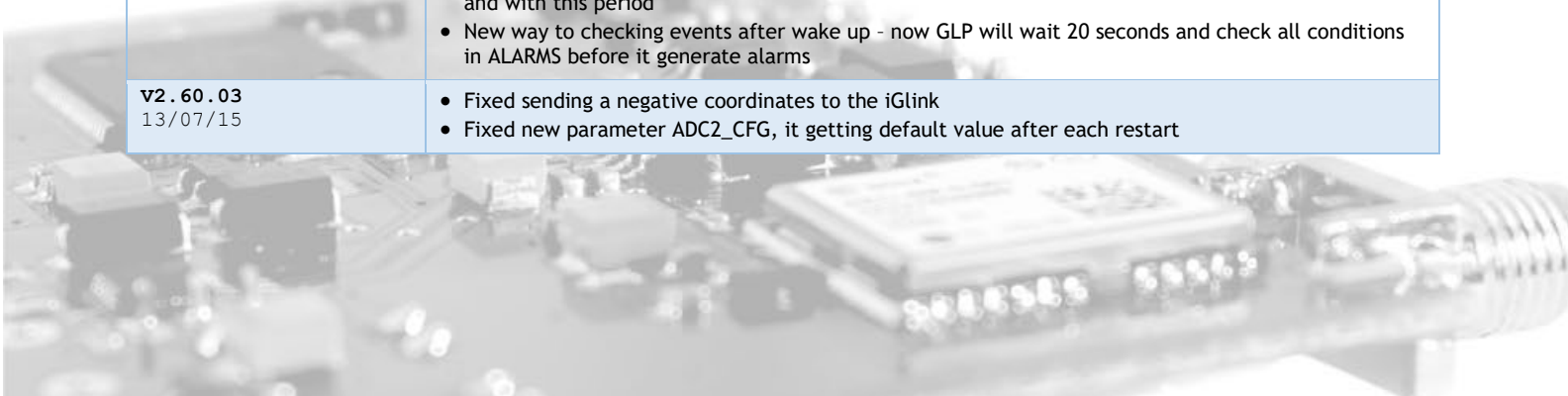
Version	Description
V4.01.30 02/10/18	• MQTT: fixed problem with mixing Events (braking/accelerating and accidents)
V4.01.29 25/09/18	• MQTT: fixed problem with EcoDriving events • MQTT: back to old seat sensor counting logic
V4.01.28 19/09/18	• MQTT: improvement of “mil” value
V4.01.27 15/09/18	• MQTT: fixed “mil” value in BasicMetaInfo tag • MQTT: new seat sensor counting logic • MQTT: odometer value in meters
V2.72.26/V3.10.26 **V4.01.26/V5.01.26** **V7.01.26** 07/08/18	• Added ignition source from movement sensor and battery voltage • Fixed problem with saving configuration on low battery voltage • Added .takepic function to action list in filter settings • Added possibility to send picture by connection without ACK confirmation. • Fixed bug with downloading operator list file from server
V3.10.25/V4.01.25 27/07/18	• Sending last saved course value in dataframe
V2.72.24/V3.10.24 **V4.01.24/V5.01.24** **V7.01.24** 13/06/18	• Fixed problem in wakeup by accelerometer • Fixed connection problem in 3G version • Added function .idsupdate to update IMEI/IMSI if wrong
V2.72.22/V3.10.22 **V4.01.22/V5.01.22** **V7.01.22** 14/05/18	• Added .fcanfast X function, where X is time in seconds to trigger cansearch • Added function .uploadconf
V2.72.21/V3.10.21 **V4.01.21/V5.01.21** **V7.01.21** 29/03/18	• Added function .fcanreset X
V4.01.20 28/03/18	• MQTT: fixed counter of passengerCount
V4.01.17 06/03/18	• MQTT: sending DriverID • MQTT: fixed dataframe of drivername/id
V4.01.16 01/03/18	• MQTT: changed passengerCount value on 65 parameter
V3.10.14/V4.01.14 **V5.01.14/V7.01.14** 08/02/18	• CAN enabled in default configuration • 3G: disabled SIM Card checking on changing operator
V4.01.13 22/12/17	• MQTT: added function ".485conf"
V4.01.12 20/12/17	• MQTT: sending odometer value if higher than 0
V4.01.11 18/12/17	• MQTT: fixed restoring of default configuration
V3.10.09/V4.01.09 **V5.01.09/V7.01.09** 16/11/17	• Fixed RF connection problem • Fixed CAN bus frame length • Fixed BT baudrate bug on configuration



V4.01.08 29/10/17	• MQTT: Added to passengerCount singals from DroneIO/OW
V4.01.07 22/10/17	• MQTT: changed engine status value • MQTT: disabled sending odometer value if 0
V2.72.06/V3.10.06 **V4.01.06/V5.01.06** **V7.01.06** 11/10/17	• Changed variable user and service tel length to 20 bytes • SSL disabled when updating
V2.72.02/V3.10.02 **V4.01.02/V5.01.02** **V7.01.02** 27/08/17	• Added support for new inputs
V2.72.01/V3.10.01 **V4.01.01/V5.01.01** **V7.01.01** 15/08/17	• Added read IMSI from gsm module, parameter "imsi" to read • Fixed FW check for updating to different versions • Fixed working of accelerometer LIS3DH
V2.71.09/V3.09.09 **V4.00.09/V5.00.09** **V7.00.09** 08/05/17	• New hardware versions introduced: o 4.x.x - GLP 2G V3 – 2G network and new accelerometer o 5.x.x – GLP 3G V3 – 3G network and new accelerometer o 6.x.x - reserved o 7.x.x – GLP 3G V3 + Bluetooth – 3G network, new accelerometer and support for Bluetooth communication • Added Bluetooth communication for technical support • Ability to reset (SMS command) and change (technical support commands) the Bluetooth PIN • Added support for new accelerometer (LIS3DH) • Added ability to send device name (DEV_NAME variable) in data frames (Parameter 245) • Added SIM card detection in slot • Increased speed of saving configuration data to FLASH
V2.70.02/V3.08.02 08/11/16	• GLP disable outputs before go to sleep in shippingmode • DIS_CFG – new parameter to change meaning of third parameter in FILTER0…7. Now GLP can save position when distance is reached instead of speed change
V2.70.01/V3.08.01 13/09/16	• Added support of acknowledges (requires GLP plugin/driver version 3.0.0.3073) • Fixed translating hostnames to IP addresses in UDP mode • UPDATE_SMS – new parameter to disable sending SMS after update
V2.69.06/V3.07.06 11/08/16	• GLP goes sleep faster in intelligent sleep mode • Fixed problem with multiple command executing from SMS (problem was caused by promotional SMS)
V2.69.05/V3.07.05 27/07/16	• Fixed problem in 3G version that GLP cannot make call
V2.69.04/V3.07.04 20/07/16	• Fixed problem that cause problems with GPRS connection especially with low gsm signal
V2.69.02/V3.07.02 07/07/16	• Code optimization
V2.69.01/V3.07.01 28/06/16	• Added events to alarms from drone (analog inputs and no available drone) • GSMMODE – new parameter to send information about 2G/3G mode (only in 3G version of GLP)
V2.68.06/V3.06.06 22/06/16	• Added confirmation during picture sending (requires GLP plugin/driver version 3.0.0.3072)
V2.68.04/V3.06.04 16/06/16	• Removed call answering because of possibility to lock device after call • Fixed reading position after .systemreset
V2.68.02/V3.06.02 17/05/16	• Added support for accumulate reading from CAN, for example for reading total fuel level from fuel rate, two modes • TOTALCAN0 to TOTALCAN7 – new parameter to store accumulated CAN data • Turning off GLP (sleep) when it is without external power and battery voltage is below 3,6V to protect memory from incorrect write • Fixed buffering positions when memory is full • Fixed logout driver ID when GLP goes sleep

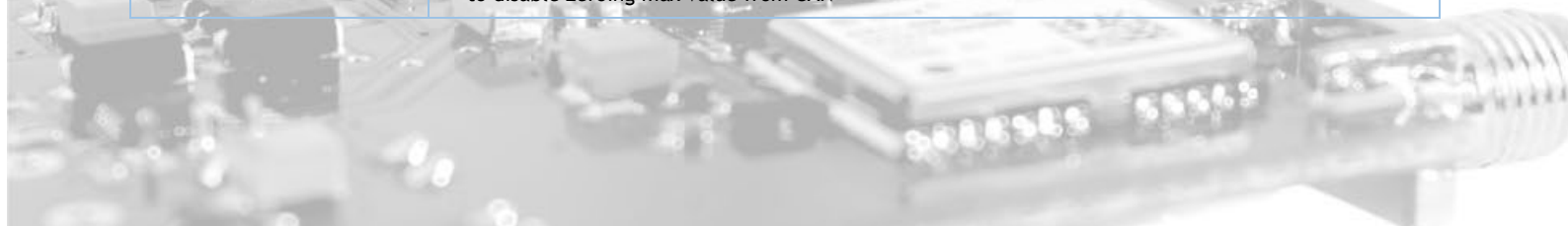


V2.67.08/V3.05.08 29/02/16	• Improved testcan and cansearch command – now it can read up to 170 frames • .testflash/@testflash – new command to test whole memory
V2.67.05/V3.05.05 17/02/16	• Fixed taking picture from RS485 Camera with different ID than 0
V2.67.03/V3.05.03 10/02/16	• Added support for camera (requires GLP plugin/driver version 3.0.0.3070) • EXTEN_CFG – new parameter to control working of extension devices (camera RS232 and RS485) • .takepic – new command to take a picture from camera • Fixed bug which caused measure VCC voltage instead ADC in sleep mode
V2.66.03/V3.04.03 02/02/16	• Support for the new model of GSM module (SIM800F) • Fixed bug that caused in specific situation GLP stucking after GSM operator changed
V2.66.01/V3.04.01 20/01/16	• Fixed bug which caused that GLP could stuck on start when memory errors occurs • Fixed update from old FW • Added support for two Driver IDs • DKEY_CFG – new sub-parameter to set number of IDs from IButtons or iPkeys to send • .shippingmode – new function that forces sleeping independently of configuration, GLP will wake up when voltage rise up to 10V
V2.65.01/V3.03.01 14/12/15	• Added support for A-GPS • Added GPS Jamming detection • GPS_CFG – new sub-parameter to enable A-GPS and to set parameter for GPS Jamming detection
V2.64.06/V3.02.06 09/12/15	• Fixed GSM buffer
V2.64.03/V3.02.03 26/10/15	• Changed time to aggregate fault codes from 30 seconds to 5 minutes • Changed actions in FAULT_CFG (added “monitoring”) • Time to send when ignition on can be more than 255 seconds (65535 seconds)
V2.63.07/V3.01.10 20/10/15	• Fixed bug which caused that GLP could stuck on start (chance to occurs is 1 to 10000)
V2.63.06/V3.01.09 08/10/15	• Added possibility to set manual FMS parameter with source address
V2.63.04/V3.01.07 25/09/15	• Added confirmation with terminal status (required to using chat/dispatch with plugin 3.0.0.3069)
V2.63.03/V3.01.06 15/09/15	• New 3G version! • Fixed PIN configuration (problem with setting PIN beginning with 0)
V2.62.08 11/09/15	• Fixed reading fault lamps
V2.62.06 28/08/15	• Fixed method of switching operator codes
V2.62.05 25/08/15	• Fixed method of getting fault codes
V2.62.04 18/08/15	• Fixed bug which caused that GLP could lost configuration
V2.62.03 17/08/15	• Fixed Monitor functions • Disabled Preferred operator code (it caused problems with connections to network) • Fixed fault codes (all the time it was “00000000”)
V2.61.05 28/07/15	• Added reading status of fault lamps and fault codes from FMS bus • FAULT_CFG – new parameter to control faults reading (requires GLP plugin/driver version 3.0.0.3066)
V2.60.05 15/07/15	• Fixed bug which caused that GLP not wake up from sleep using ignition when ALARM0 has “AND” logical condition • New way to detect movement event • TITL_CFG – new sub-parameters with number of movements in period to generate movement event and with this period • New way to checking events after wake up – now GLP will wait 20 seconds and check all conditions in ALARMS before it generate alarms
V2.60.03 13/07/15	• Fixed sending a negative coordinates to the iGlink • Fixed new parameter ADC2_CFG, it getting default value after each restart



V2.60.02 07/07/15	• Driver ID will logout immediately when GLP goes sleep • ADC_CFG – new sub-parameter with time for averaging analog input • ADC2_CFG – new parameter to support two additional analog inputs (parameter, threshold and averaging), needs additional elements on PCB • CAN_ID0..7 – new sub-parameter with threshold for CAN data for alarms • Added CAN data event for alarms
V2.59.15 10/06/15	• Added unix time stamp in response from cansearch • .startcansearch – new command to start continuous cansearch • .stopcansearch – new command to stop continuous cansearch • Disabled outputs when update from FW older than 2.51.03
V2.59.14 08/06/15	• Added High Resolution Speed (0.01km/h) (requires GLP plugin/driver version 3.0.0.3063) • GPS_CFG – new sub-parameter to enable high resolution speed • Added sending CSQ parameter (requires GLP plugin/driver version 3.0.0.3063)
V2.59.10 22/05/15	• Added EcoDriving based on GPS • Added EcoDriving based on accelerometer • ECO_CFG – new parameter to control EcoDriving mode • Added Accident mode based on accelerometer • ACCIDENT_CFG – new parameter to control Accident mode • .acccaltab/@acccaltab – new command to calibrate accelerometer on table (step 1) • .acccalcar/@acccalcar – new command to calibrate accelerometer in car (step 2) • TILT_CFG – new sub-parameter to delay movement sensor calibration after ignition is off • Changed LED blinking • Fixed saving position in short time • Fixed sending buffered data • gsm_model – new parameter to read GSM model
V2.58.11 08/04/15	• Fixed reading driver ID from tachograph • Added reading second driver ID from tachograph (send on 243 parameter) • .testcan automatically enable CAN if not enabled • Added Update Error 0 – it will work when updated from this FW
V2.58.06 26/03/15	• Added getting time from GSM when GPS is unavailable • Added a third option to pulse measurement, pulse duty cycle measurement • PLANE_MODE – new parameter to enable plane mode (1800Mhz only) • PULSE_THRESH – new parameter with threshold for pulse measurement for alarms • Added support for generate event from pulse measurement • Added support for reading Global Cell ID (MCC+MNC+LAC+CELL ID) • GCI_CFG – new parameter to control parameters for Global Cell ID • @getops – new command, equivalent to .getops but using TCP • .changeops – new command to forcing a change of operator • Fixed averaging values from CAN at a specified number of frames • Fixed reading read-only variables using TCP • Fixed working of OneWire interface • Fixed algorithm to checking geofences • Fixed configuration of old GSM module SIM300
V2.57.02 05/02/15	• Added support for pulse measurement (frequency, summation) on pin IN-3 • PULSE_CFG – new parameter to configuration of mode and parameter pulse measurement • PULSOMETER – new parameter with pulsometer value • Added support for AutoGeofencing • AGF_CFG – new parameter to configuration of conditions and radius of AutoGeofencing • Fixed logout time in Driver ID • Disable zeroing position when invalid position, send last known with invalid flag (requires GLP plugin/driver version 3.0.0.3057 for proper working; on previous versions, server will treat invalid position like valid)
V2.56.01 16/01/15	• Added support for digital thermometer DS18B20, DS18S20, DS1822 • THERM_CFG – new parameter to configuration of parameter number for temperature value
V2.55.02 05/01/15	• .getops – new command to get current and available operator codes • cops parameter removed • Fixed setting GSM modem when preferred operator are not entered • Operator groups have priorities (1 – high, unknown – low) and GLP will try connect to operator codes with the highest priority, it will check them with available operators.

V2.55.01 15/12/14	• Fixed sending output state (requires new GLP plugin/driver on server) • Fixed file writing to memory (e.g. update) when Driver ID is enabled, it could cause Driver ID file erasing • DKEY_CFG – new sub-parameter to independent control iButton, iPkey, and FMI ID • DKEY_CFG – new sub-parameter to control readings driver ID from Tachograph • ALARM_CFG removed, now alarm flags are built into the protocol (requires new GLP plugin/driver on server)
V2.54.12 28/11/14	• Fixed alarm from analog inputs (supply and AIN1)
V2.54.05 25/11/14	• Fixed sending data from events often than every second • .pincfg - new command to setting PIN on SIM card
V2.54.03 30/10/14	• Three times smaller current consumption in sleep mode • SLEEP_CFG – new parameter to control sleep rules • ALARM0...ALARM3 – new parameters to control event rules • FILTER0...FILTER7 – removed some of sub-parameters and added new sub-parameter to control alarm actions • FLAG_ISBAR and EVENT_CFG removed • ALARM_CFG – new parameter to control alarm flags • ACT_GROUP0...ACT_GROUP3 – new parameters to group actions together • Fixed problem with freeze RSMBP devices • Improve work of iPkey, handling few iPkeys, removed its sub-parameters • Improve work of odometer • Fixed default time – now starting time is not from future • Fixed sub-parameter HDOP in FILTER • Removed restriction on prefix “+” and “00” in phone numbers • ID event in FILTER launches only when ID is on list or “execute common actions for all identifiers” option is enabled • TILT_CFG – new parameter to control accelerometer sensitivity (accelerometer is not standard equipment)
V2.53.02 26/06/14	• Fixed sending data after error block • Retransmit timeout clears gprs blocking errors but not attempts to send data again
V2.53.01 16/06/14	• Operator list – ability to control sending data in different gsm operators • OPS_CFG – new parameter to enable operator list, 0 – disabled, 1 – enabled, default 0 • OPS_GROUP0 … OPS_GROUP4 – new parameters with actions to do after operator code is found. • Actions fixed and standardized for all elements • Calling in actions takes longer (60s) • Ability to send Driver ID via SMS • DID_LOGIN and DID_LOGOUT fixed, configuration scheme like parameters with actions • FILTER_CFG – new parameter with current active filter; actions or user can change it • FILTER0 – new sub-parameter to control mode of sending data • FILTER0 – new sub-parameter to control actions from driver id and FMI • GSM_CFG can now be disabled by setting it to 0 • Parameters in FILTER0 can now be disabled by setting it to 0 • SERVICE_DID_FILE – new parameter with path to the driver ids file • SERVICE_OPS_FILE – new parameter with path to the operator list file • .getdidfile – new command to download the Driver IDs file from the service web server • .getopsfile – new command to download the Operator Codes file from the service web server • Fixed averaging values from CAN when CAN parameters are encrypted
V2.52.05 29/04/14	• Fixed averaging values from CAN at a specified number of frames (default 100, max 65000)
V2.52.04 28/04/14	• .testgprs sends information even if there is no GPRS connection • Save position when speed changes more than specified value in FILTER0 • GEO_CFG – new parameter to disable geofence, 0 – disabled, 1 – enabled, default 0 • .test (@test) – new command to test device basic functions • .testcan (@testcan) – new command to test CAN connection • New rules about FLAG_TEL, default 2, described in “Telephone lock level” chapter • LIMIT_TEL – new parameter which allows to configure device from “other” number • Number must be entered with prefix “+” or “00” • CAN_CFG – new sub-parameter to enable averaging values from CAN, averaging between two saved position or averaging at a specified number of frames (default 60, max 250), there are also options to disable zeroing max value from CAN

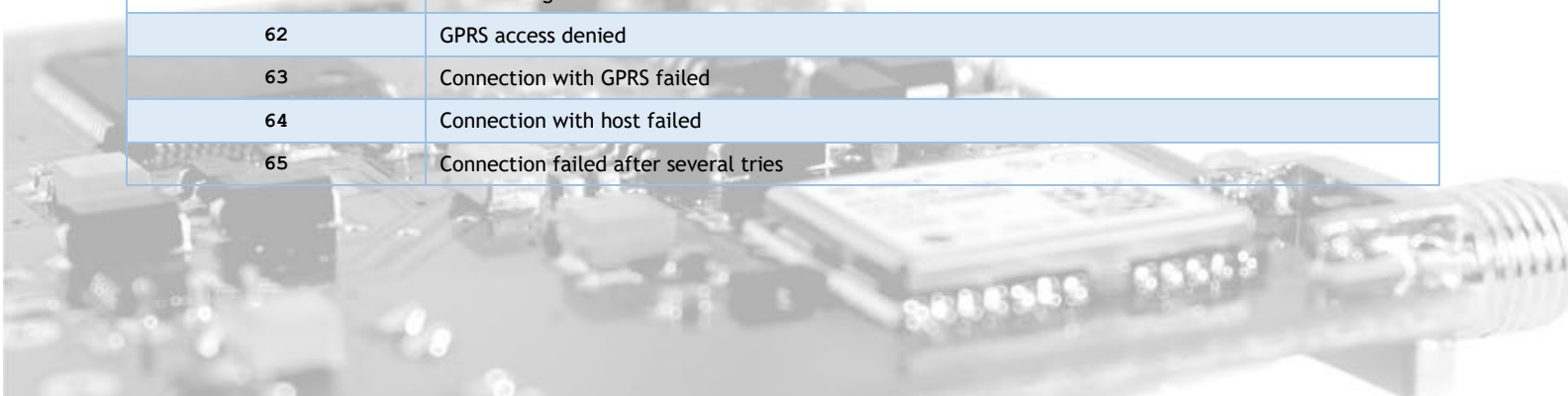


V2.52.03 10/04/14	• New compiler, code 20% smaller
V2.52.02 24/03/14	• Fixed problem with disabling CutOff wake up
V2.52.01 14/02/14	• Event rules added • FILTER0 - new sub-parameter to control action of event rules • ADC_CFG - new sub-parameter with thresholds to event rules • EVENT_CFG - new parameter to control when event rules occur • EVENT_TEL - new parameter to control who has to be informed in a case of event occurring • New rules for sleep function, device go sleep in a defined range • ":" deletion from answer after .get and .setc • „variable not faund" deletion after .set with wrong parameter • The device send sms to a number from which was the request • SMS from device after .FLAG_TEL_RESET
V2.51.05 17/01/14	• Improve work of SIM card with PIN • GLP password changing
V2.51.04 13/01/14	• The default lock device (communication with configurator and via SMS from non SERVICE_TEL numbers)
V2.51.03 09/01/14	• "=" deletion from answer after @get • .getrap with IMEI • SERVICE_TEL adopt USER_TEL function • Two additional USER_TEL and SERVICE_TEL • Answering the phone after three signals from USER_TEL numbers (prepared for a possible upgrades of the speaker and microphone) • Getting commands only from SERVICE_TEL numbers • LENGTH_TEL - to check the exact number of digits from the end (differences in prefix) • FLAG_TEL - to check the phone number, 0-doesn't check, 1-check SERVICE_TEL, 2-check USER_TEL, 3-check both • FLAG_ISBAR - lock flag of communication with configurator, 0 - unlocked, 1 - locked • .FLAG_TEL_RESET XXXXXXXX - command to reset FLAG_TEL. XXXXXXXX is GLP password • Solving problem with freeze CAN after cansearch
V2.51.02 07/10/13	• Remotely adding, editing and deleting iButtons • iPkey treaded as an iButton
V2.51.01 29/09/13	• TimeoutMove in HybridFilter fix • Sending the iButton ID to server

7. Error codes

7.4 From GSM modem

Number	Description
41	To low battery voltage
42	Turn on GSM modem error
45	AT command error
46	Operator change error
51	SMS service error
59	Temporary GPRS lock to check operator codes
60	GPRS lock from active FILTER
61	GPRS configuration error
62	GPRS access denied
63	Connection with GPRS failed
64	Connection with host failed
65	Connection failed after several tries



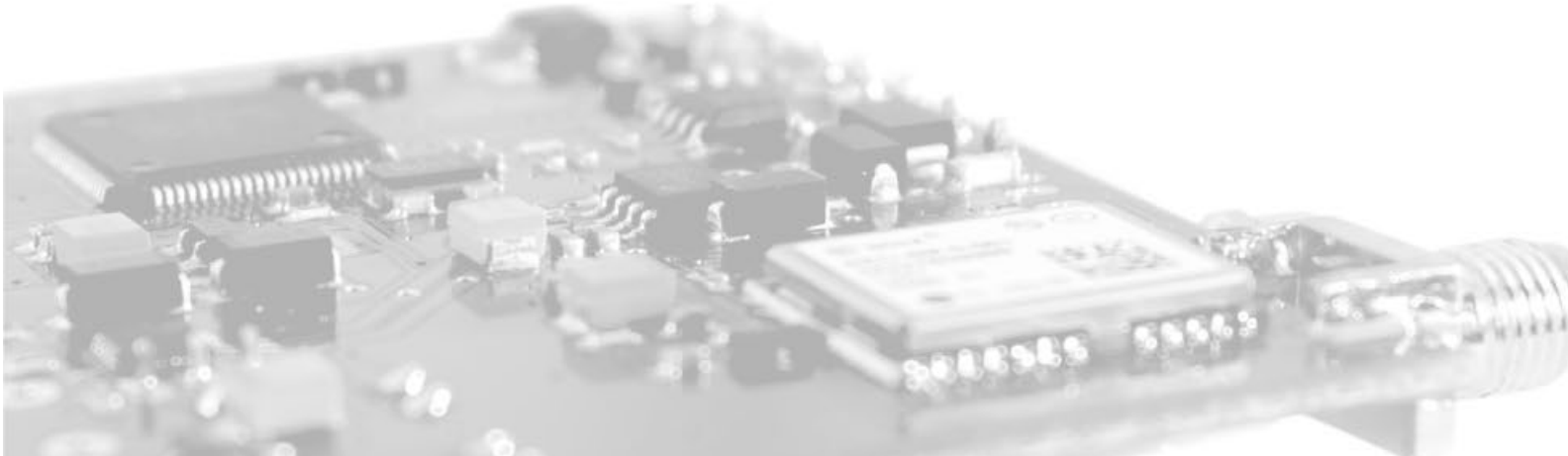
66	Data transmission error (host connection available)
67	Wrong server HTTP response
68	TCP connection closing failed

7.5 From HTTP server

Number	Description
70	Too frequent data update (min 2sec delay)
71	Server didn't pick up the data
72	Header error
73	Wrong password
74	Unrecognized transmission type
75	Data quantity/declaration mismatch
76	Marking samples with date and time in the future is forbidden
77	Samples CRC mismatch - data corruption
78	Database connection error
79	Database insert error

7.6 Update Errors

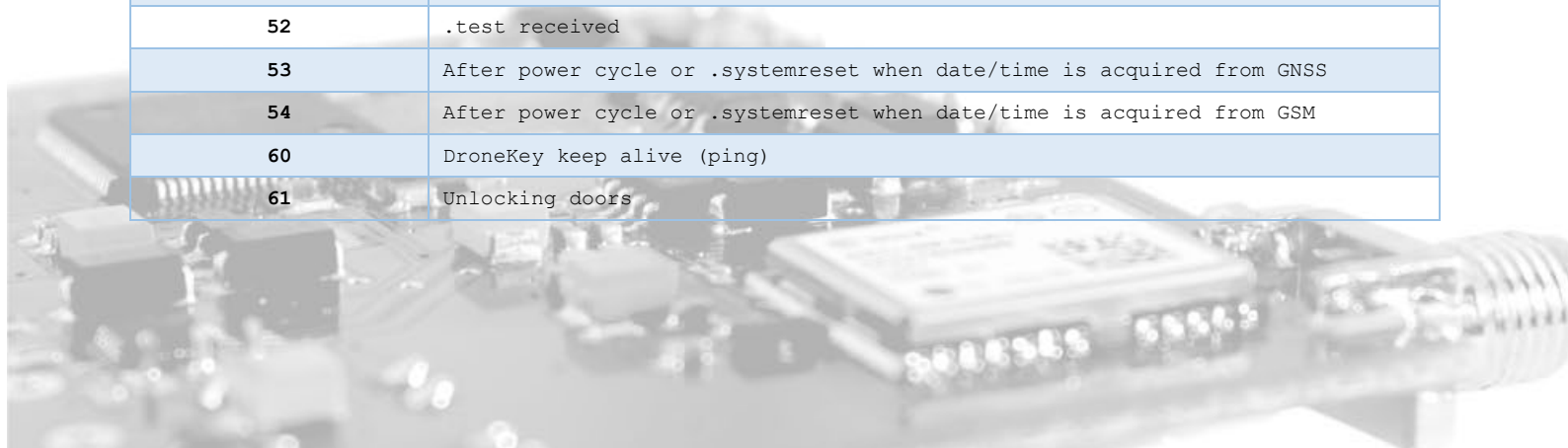
Number	Description
UPDATE error 0	GSM/GPRS error
UPDATE error 1	Memory error
UPDATE error 2	File error
UPDATE error 3	Write error



8. Data Acquisition Triggers

Data acquisition triggers are reported on parameter 67. Available from firmware 2.71.50 / 3.10.50 / 4.1.50.

Number	Trigger
11	Acceleration (source: accelerometer) over threshold (eco driving event)
12	Braking (source: accelerometer)over threshold (eco driving event)
13	Cornering (source: accelerometer) over threshold (eco driving event)
14	Acceleration (source: GNSS) over threshold (eco driving event)
15	Braking (source: GNSS)over threshold (eco driving event)
16	Cornering (source: accelerometer) over threshold (eco driving event)
17	Current filter: timeout
18	Current filter: degrees change
19	Current filter: distance change (speed change with DIS_CFG=1 used as distance change)
20	Current filter: speed change
21	New Driver ID found on the list
22	Driver Status changed; source: Garmin FMI
23	New Driver ID read
24	Driver logout event based on DKEY_CFG (time out)
30	GSM Operator changed (MNC)
31	Accident alarm
32	Ignition ON
33	Ignition OFF
34	Entered Geo-zone
35	Left Geo-zone
36	Entering sleep mode
40	Alarm0 triggered
41	Alarm1 triggered
42	Alarm2 triggered
43	Alarm3 triggered
44	Alarm4 triggered
45	Alarm5 triggered
46	Alarm6 triggered
47	Alarm7 triggered
50	.SAVEPOS received via SMS or TCP or UDP
51	.SAVEPOS send via Debug Terminal (S)
52	.test received
53	After power cycle or .systemreset when date/time is acquired from GNSS
54	After power cycle or .systemreset when date/time is acquired from GSM
60	DroneKey keep alive (ping)
61	Unlocking doors



62	Locking doors
64	Immobilizer OFF (inactive)
68	Immobilizer ON (active)
70	GSM module reset (starting from firmware 1.02.48)
71	GNSS module reset (starting from firmware 1.02.48)
72	Default configuration restored
73	Bootloader update
74	Simulated ignition ON (starting from firmware 1.02.48)
75	Successful GNSS module update (BG96 in GLP 2.0) (starting from firmware 1.02.48)
76	Periodic configuration save to memory
80	Output (any on GLP) deactivated (OFF)
81	Output (any on GLP) activated (ON)
90	CRC error when reading data from queue (buffer).
91	Error when reading from FLASH memory – garbage.
92	Incorrect initial number when reading data from queue (buffer).
93	Error when reading from FLASH memory – deleted data.
241	GSM module reset (changed to 70 from firmware 1.02.48)
242	Reset of module because of FIX issue (changed to 71 from firmware 1.02.48)
245	Simulated ignition ON (changed to 74 from firmware 1.02.48)
246	Successful GSM module update (GLP 2.0) (changed to 75 from firmware 1.02.48)

9. Jamming / interference monitor reported states

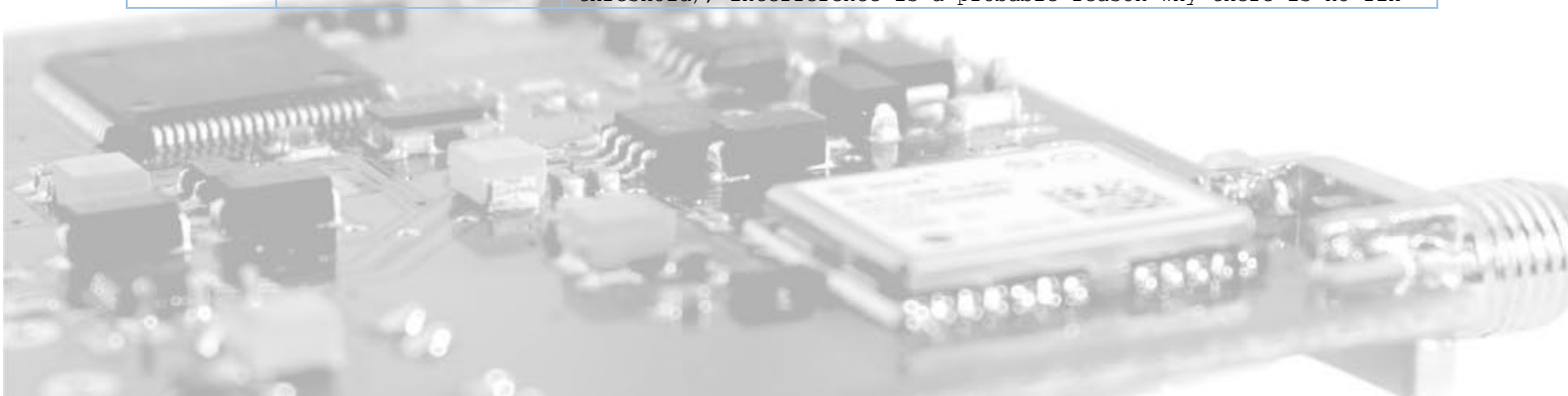
To make sure your hardware supports Jamming please send **@get gsm_model**. If reply is SIM800 then your device supports Jamming.

Values reported from firmware 2.65.01/V3.03.01 and above.

Value	Reported state	Description
0	OK	no interference detected
1	Critical	not reliable position fix with interference visible (above the threshold); interference is a probable reason why there is no fix

Values reported from firmware 2.73.27/4.02.12 and above.

Value	Reported state	Description
0	Unknown	jammer monitor not enabled, uninitialized or antenna disconnected
1	OK	no interference detected
2	Warning	position ok but interference is visible (above the threshold)
3	Critical	not reliable position fix with interference visible (above the threshold); interference is a probable reason why there is no fix



10. Indicators from CANbus

Bit	Reported indicator	Description
1	Doors	1-open; 0-close/data not available
2	Hood/boot	1-open; 0-close/data not available
3	Central lock	1-locked 0-unlocked/data not available
4	Factory alarm	1-armed 0-unarmed/data not available
5	Seat belt	1-unfasten; 0-fasten/data not available
6	Handbrake	1-on; 0-off/data not available
7	Brake	1-on; 0-off/data not available
8	Clutch	1-on; 0-off/data not available
9	Hazard light	1-on; 0-off/data not available
10	Running light/low beam	1-on; 0-off/data not available
11	High beam	1-on; 0-off/data not available
12	Rear/front fog light	1-on; 0-off/data not available
13	Fuel reserve	1-on; 0-off/data not available
14	Cruise control	1-on; 0-off/data not available
15	A/C indicator	1-on; 0-off/data not available
16	Reverse gear	1-on; 0-off/data not available

11. DroneKey integration

When DroneKey is successfully detected it starts to send keep alive packets to GLP device every few seconds. GLP generates additional data package with reason of acquisition #60. If GLP looses communication with DroneKey it will power cycle it using GND line supplied to DroneKey via OUTPUT 2. As a backup OUTPUT 1 in GLP 2.0 is used too.

When .immoon / .immooff command is used you will receive both confirmation the command was received (reason #64 and #68).

Configuration is perform using CARSH_CFG setting. You can find more details regarding care sharing functionality online https://docs.geotik.com/display/GLPDOC/Car+Sharing