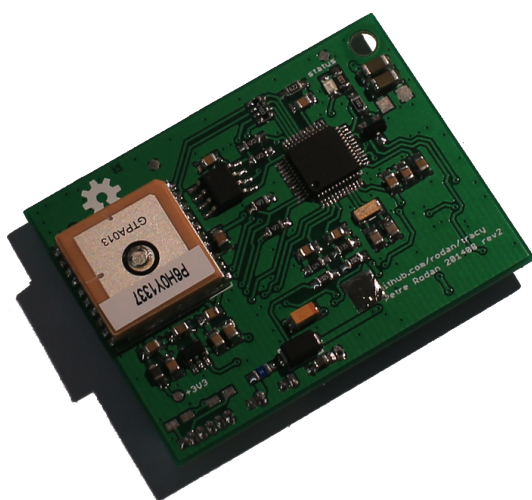

Gps tracking module with gprs connectivity

Petre Rodan <http://www.subdimension.ro>

created September 2014, revised December 6, 2018



Description: This module can be used to track goods wherever gsm signal is present. It is optimized for very low power consumption and has configurable timings which would make it fit to any use case scenario.

Main features: both hardware and software are open-source, an f-ram chip for input data buffering, resilient software architecture, all gps/gsm operations controlled by timer interrupts which allows control of both sub-systems in the same time, great flexibility given by configuration variables

Size: 50mm x 38mm x 9mm

Voltage input: 3.7V battery and optional 5V DC for charging

Contents

1	Principles of operation	3
2	What's included	3
3	Quick reference guide	3
3.1	Getting started	3
3.1.1	Requirements	3
3.1.2	SIM cards and initial setup	3
3.2	gprs setup	4
3.3	Web visualization of data	5
3.4	SMS commands	5
4	Developer guide	7
4.1	Software	7
4.1.1	Tweaking the firmware	8

4.1.2	HTTP POST packets	9
4.1.3	Timings and intervals	11
4.1.4	List of timing variables	12
4.2	Hardware	14
4.2.1	Absolute maximum ratings	14
4.2.2	Electrical characteristics	14
4.2.3	Components	14
4.2.4	Connectors	16

1 Principles of operation

Geographic coordinates are obtained by decoding NMEA sentences received from the GPS chipset and then transmitted via the mobile phone network to a web server. The server stores the received information into a database and provides it on demand to the end-user.

Cell tower identifiers are also transmitted in the same packet and this allows approximate localization when the gps signal is obstructed. When the gsm signal is missing data is being stored on the device and it's transmission will resume once connection is re-established.

2 What's included

1 x tracking module
1 x GPRS antenna with ufl adapter (manufacturer id: WLS119E1B)
2 x 2 pin Molex picoblade connectors and short cable for battery and DC input (manufacturer id: MOLEX 0510210200 and MOLEX 0500798000)
1 x 6 pin Molex picoblade connector and crimp terminals for creating a programming cable (manufacturer id: MOLEX 0510210600 and MOLEX 0500798000)
1 x shrink tube

Soldering of the wires is needed. Batteries, sim cards and enclosures are not included.

3 Quick reference guide

3.1 Getting started

3.1.1 Requirements

Warning Special care must be taken with the gprs antenna. It has to be connected to the connector marked 'gprs' at all times or otherwise the module could be damaged.

This module works best with two power sources. One 3.7V Li-Polymer/Li-Ion cell (at least 2000mAh recommended) and a 5V input DC that is only used for recharging the cell. Only the second one is optional due to the fact that for very short bursts the gprs modem needs more current than the DC input can handle. This circuitry is USB power friendly. If the module gets installed onto a vehicle, a high efficiency DC-DC converter can be provided to lower the voltage from up to 24V to the 5V needed. Double check the polarity on the connectors before attempting to apply any power to the device. Pin 1 is marked on the silkscreen with a small triangle and that represents the ground (negative) terminal for both the battery and the DC inputs.

The user will also need a mobile phone in order to send commands to the module via SMS and a web browser to visualize the generated data.

3.1.2 SIM cards and initial setup

Below is a step-by-step guide for installing a local SIM card into the tracking module. The setup of a 'Vodafone prepaid card' is shown as an example.

- buy a SIM card. any SIM card will do: prepaid, subscription (but do keep in mind that no voice transmission will ever be used on this SIM tho), machine to machine (also known as M2M). mini SIM form factor cards are needed.

- obtain the associated phone number and enter it into your phone's address-book - don't forget to also add the country prefix so you can contact the tracker while abroad.
- place the newly purchased SIM card into a mobile phone
- if it has no initial credit, fill it with at least 5EUR
- make sure no PIN is set on the SIM itself
- disable any 'best value plan' the provider might have selected - this SIM will mostly be used to connect to the internet, and it will send a minimal number of SMS messages during the setup process. the only extra options that are needed are the cheapest internet plan available ('100MB internet for 1 Month - 2EUR' for vodafone). also activate data roaming.
- disable 'VOX voicemail' by calling ##002#
- place the SIMs back into their proper locations: the newly purchased one in the tracker and your old one into the phone.
- while inserting the SIM into the tracker write down the 15 IMEI digits that you see written on the SIM900 IC that is adjacent to the SIM holder.
- send a SMS from the phone to the tracker that contains:

`code 5551`

'5551' will need to be replaced by the last 4 digits you wrote down during the previous step

- power on the tracker and wait. it only connects to the gsm network every 15 minutes by default and it has a 1 minute interval in which it's actively listening to incoming SMS messages. once the code is received and acknowledged you will get a 'code ok' reply on the phone. from this moment on you can send commands to the tracker.

3.2 gprs setup

In order for the module to connect to the internet to send out the tracking data it needs a working gprs connection to the mobile service provider. This connection is established only if a set of credentials (apn, username and password) are used to authenticate the device. You can ask your mobile phone company to provide these. Sometimes one company has customized apn and/or usernames for each sim card type, so don't rely on web searches.

Once you get your correct credentials, send them as 3 different SMS commands to the device:

`apn STRING`

`user STRING`

`pass STRING`

Warning All strings can be max 20 characters long. If you need more on any of them please contact me.

You can check if all data has been received ok with the following command:

```
gprs?
```

3.3 Web visualization of data

You can check all the info the tracker sends out by visiting http://www.subdimension.ro/scripts/l1?i=IMEI_NUMBER . the IMEI_NUMBER is a 15 digit number and it's written on the sim900 module.

This project offers the possibility to advanced users to redirect the tracking data to another web server. Two commands control the location where HTTP POST packets are sent:

```
srv your.server.com
```

```
port port_number
```

Warning All strings can be max 20 characters long. If you need more on any of them please contact me.

The post request will be sent to http://your.server.com:port_number/u1. u1 has to be a CGI script (or an alias to one) that decodes the binary packets sent. See Section 4.1.2 for details.

3.4 SMS commands

command	<pre>code STRING</pre>
action	pair a phone to the tracker. STRING has to be the last 4 digits of the SIM900's IMEI

command	<pre>gprs?</pre>
action	show APN, USER, PASSWORD that are currently used in order to authenticate for a gprs session

command	<pre>setup?</pre>
action	show a hex value containing current switches

command	<pre>fix?</pre>
action	try to send back a GPS fix as a SMS reply

command	<pre>ping</pre>
action	a HTTP POST is sent with all available data once this command is received

command	<code>default</code>
action	restores ALL variables to the factory defaults and re-boots the device. the user will have to use the code command again to pair it's phone

command	<code>apn STRING</code>
default action	live.vodafone.com set the gprs apn value - max 20 char long in case STRING is missing, an empty apn is used should be set to 'net' for an orange SIM

command	<code>user STRING</code>
default action	live set the gprs user - max 20 char long in case STRING is missing, an empty username is used

command	<code>pass STRING</code>
default action	vodafone set the gprs pass - max 20 char long in case STRING is missing, an empty password is used

command	<code>srv STRING</code>
default action	t.subdimension.ro what server to contact in order to send the tracking data

command	<code>port STRING</code>
default action	80 TCP port used during connection to the server

command	<code>vref STRING</code>
default action	198 STRING is a number between 190 and 210 that acts as calibration coefficient to the on board ADC. only used if the voltage values reported by the module differ from the measured ones

command	<code>err?</code>
action	shows a hex value in case there were any errors logged in the device. this value gets reset when all power is lost, when an 'err?' command is replied to or after a HTTP POST packet is sent out/stored to F-RAM. only used for debug

command	<code>spt?</code>
action	shows a list of gps related timing values

command	<code>smt?</code>
action	shows a list of gprs related timing values

commands	<code>spl</code> STRING <code>spw</code> STRING <code>spi</code> STRING <code>spg</code> STRING <code>sml</code> STRING <code>smst</code> STRING <code>smtt</code> STRING
action	timing variables. see section 4.1.3 for details.

4 Developer guide

4.1 Software

All the documentation, sources and hardware schematics are provided as Open Source in order to encourage further improvements by the community. The TI spy-bi-wire protocol can be used to re-flash the enclosed microcontroller.

Firmware is written in C and it's created to be fully non-blocking and interrupt driven. The server-side scripts are in perl and bash.

Any web server can be used to receive the binary data sent from the tracker. The `unpack()` function present in perl, python, php will reconstruct the information and make it ready to be inserted into a database. Special care has to be taken so that the web server's reply header is as short as possible. If it's too long the module will not parse it properly and the data flow will

be disrupted. In case nginx is used using the following settings are highly recommended:

```
more_clear_headers 'Server';  
more_clear_headers 'Date';  
more_clear_headers 'Content-Type';  
more_clear_headers 'Transfer-Encoding';  
more_clear_headers 'Connection';
```

Provided scripts have the following run time dependencies (some are optional):

- perl at least ver 5.8.2
- sqlite at least ver 3.0.0
- bash
- nginx
- fcgi

4.1.1 Tweaking the firmware

The original source code is kept in a git repository at github.com. One can grab a copy by running

```
git clone http://github.com/rodan/tracy
```

or by downloading a snapshot archive

```
wget https://github.com/rodan/tracy/archive/master.zip
```

compiling is as easy as

```
cd firmware  
make clean  
make
```

The following packages are used to compile and manage the firmware:

- msp430-gcc (ver 4.6.3_p20120406 has been used)
- msp430-binutils (2.22_p20120911)
- msp430-libc (20120716)
- msp430mcu (20120716)
- mspdebug (0.22)

Burning the firmware can be achieved with any MSP430 programmer that supports SPY-Bi-Wire.

Warning Specialized programmers can detect the 2.8V rail the microcontroller uses and will not send signals with a higher voltage. Anything higher than 3V is just asking for trouble.

mspdebug can be used to program the firmware:

```
mspdebug rf2500 "prog proj.elf"
```


4.1.2 HTTP POST packets

Tracking information is sent to the server by TCP packets via the HTTP POST method. A web server is thus needed to receive the data and feed it to the parsing script (see `server/scripts/u1` in the repository).

In order to have a good bandwidth usage the data is sent in a binary format and the F-RAM chip that is being used allows the data to be buffered and sent out at set intervals. Most scripting languages have an `unpack()` function that can correctly decode these packets. A sample perl script is provided in the repository (see `server/scripts/11`).

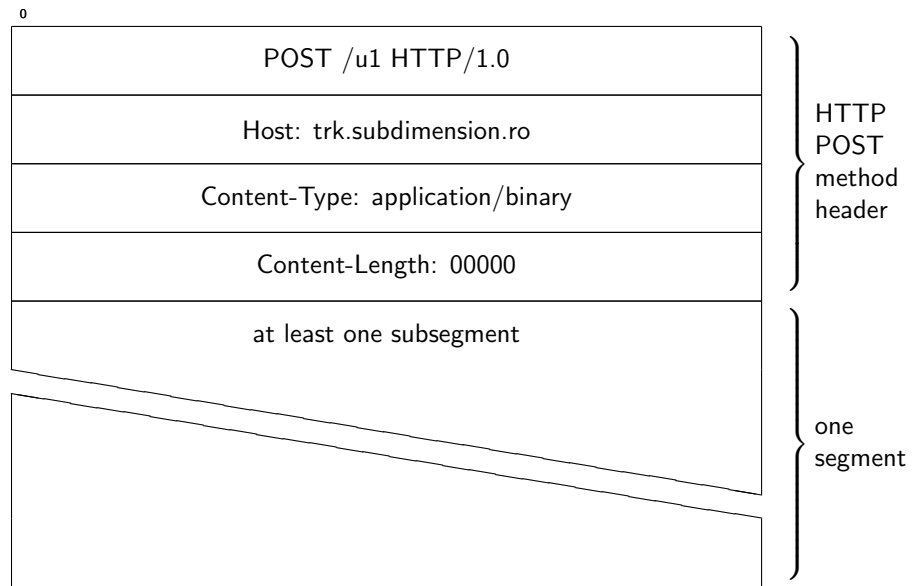


Figure 1: packet format

Each positional measurement will be placed in a subsegment. At least one of these subsegments will constitute one segment that is sent out together with a HTTP header to create an HTTP POST request (see Figure 1).

Each subsegment (shown in Figure 2) has a header containing the timestamp at the time of the fix, modem IMEI, system voltages, previous errors detected, basic device configuration, an auto increment id and a byte that describes the type of the positional data. This can either be a GPS fix (Figure 3) and/or one up to four tower identification strings (Figure 4).

The tower identification strings are post-processed on the web server and a very approximate geographic position is obtained.

These subsegments are simply concatenated together to constitute the final segment being sent. The size of one such segment varies but SIM900's internal buffering limits it up to 1000bytes. During one GPRS session one up to MAX_SEG are sent out at a time. If GPRS connections cannot be established for a long interval and MAX_SEG is reached then the oldest segment is freed in order to make place for new data.

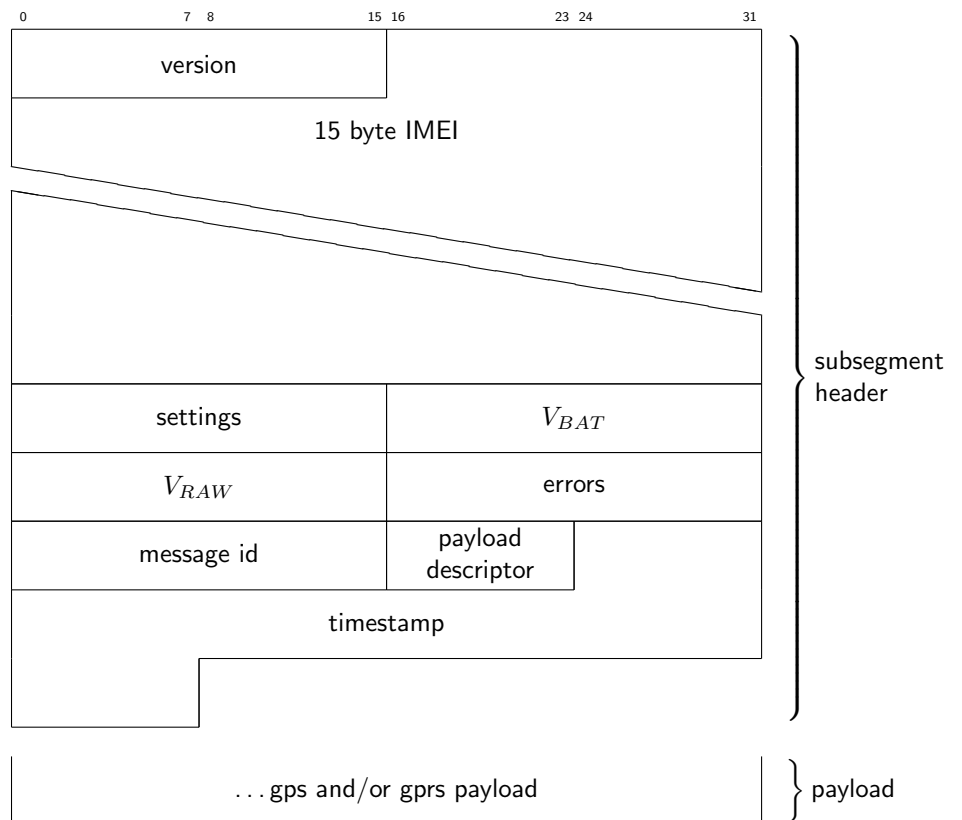


Figure 2: subsegment format

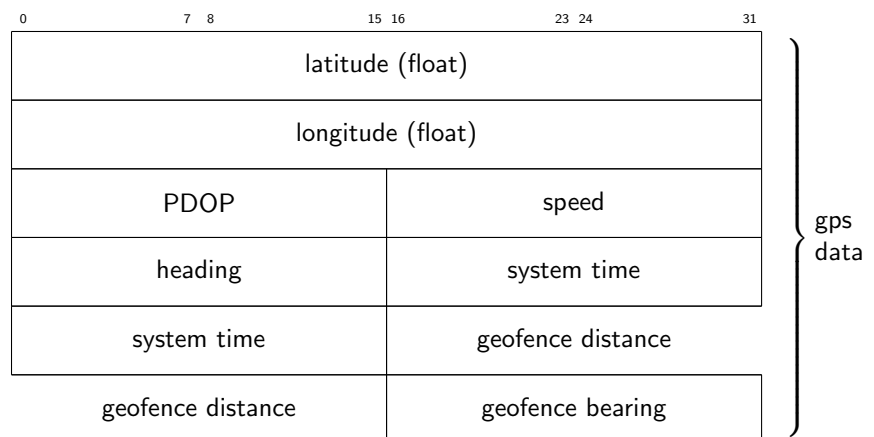


Figure 3: gps payload

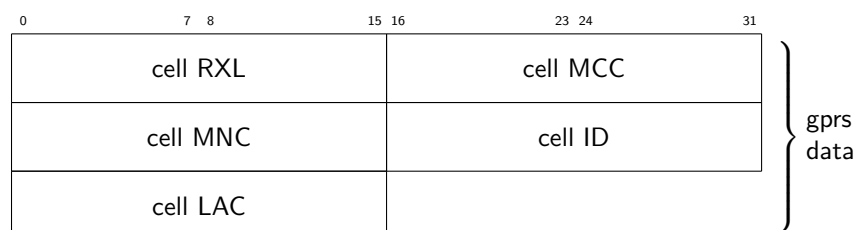


Figure 4: gprs payload

4.1.3 Timings and intervals

This tracking module allows fine-grained timing customization. It can either preserve battery life for more than a month when sparse measurements are being made or it can record positional data every 10 seconds if needed. Either behavior can be set by the user via SMS commands, without the need to change the firmware.

A 'reset to factory settings' function is also provided in case the user needs to return to known-good timing values.

Default values for the gps intervals are depicted in Table 1. The yellow time intervals are configurable: the entire loop takes 180 seconds - from which in the first 135 the gps module is powered off, a 45 sec warmup brings the device to life and it should be enough to ensure a good fix at the end of that interval. During the invalidation period the module will choose the fix with the best PDOP value and that will get stored into F-RAM.

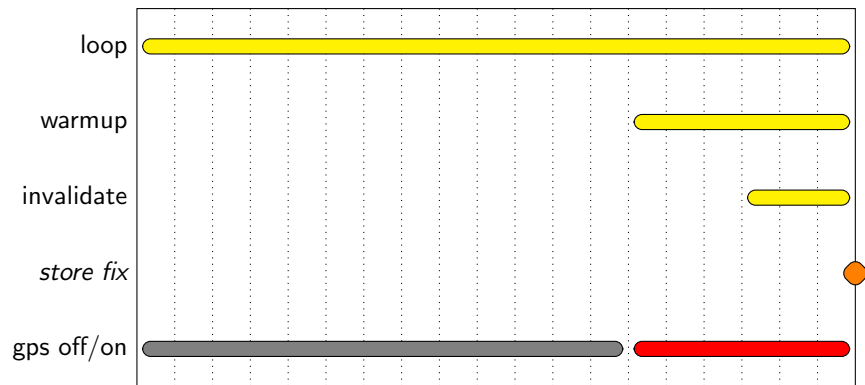


Table 1: *default time intervals for the gps subsystem*

If much more frequent gps data needs to be acquired, ie $\text{loop interval} \leq \text{warmup interval} + 30\text{s}$ then the gps will never be powered off. See Table 2 for an example where all gps intervals are set to 10 seconds.

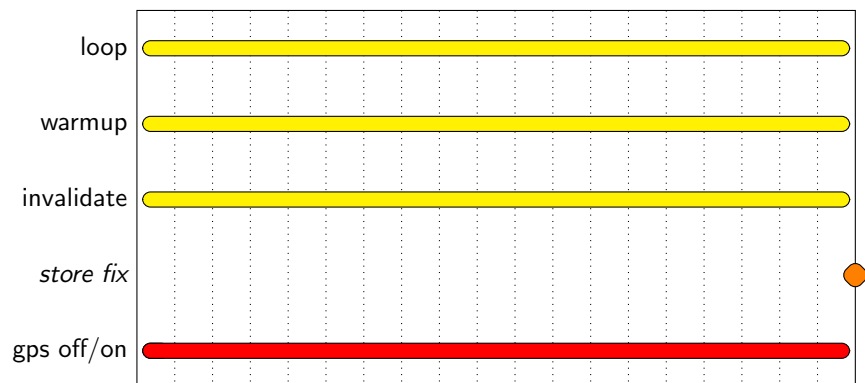


Table 2: *10s loop for the gps subsystem*

As a rule of thumb try to have at least 45 seconds as the warmup interval in case the gps gets powered off. And given the fact that the gps is set to send data every two seconds to the microcontroller make sure the invalidation interval is at least 6-8 seconds.

Unlike the gps, the gsm modem needs to be kept off most of the time since it is by far the most power hungry subsystem on this module. Also a little known fact is that internet data plans have small hidden costs that can add up in time. Sometimes there is a minimum data unit (between 1-10kbytes) that gets counted for every established gprs connection, also while in roaming each gprs initialization has a cost associated with it. So depending on your gsm provider and use case, you want to get the best trade-off using the configurable variables described in this section.

The modem gets woken up by one of these two triggers:

- a periodic check for any new SMS command to be acted upon. this wake-up has no costs associated with it if no SMS reply is to be sent back to the user
trigger controlled by `gprs loop interval`
- upload of data stored in F-RAM via gprs. SMS commands are also accepted at this point. this trigger can happen either because enough data has piled up in the memory, one of the `gprs tx intervals` has been reached, the ping command was received or the module just started to move after being stationary.
trigger controlled by `gprs static tx interval` when the module is not moving or `gprs moving tx interval` otherwise.

Whatever is the reason for the modem startup, the neighboring tower cell IDs are stored as a secondary (but less precise) positioning method.

The module behavior when using the default gprs interval settings is shown in a simplified form in Table 3. Every 900 seconds the `gprs loop interval` wakes up the modem for a 2 minute period in which possible SMSs are parsed and tower cell IDs are captured.

A movement that was detected triggers an upload of data and makes the next upload happen after 600sec (`gprs moving tx interval`). The next transfer will happen after 3600sec (`gprs static tx interval`) since the module is not moving.

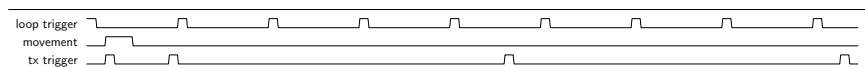


Table 3: *default module behavior*

4.1.4 List of timing variables

gps loop interval	value			unit
	min	default	max	
value	10	180	65535	seconds
sms command to set	spl <i>value</i>			
sms command to view	spt?			
description	sets the interval between two gps measurement sessions			

gps warmup interval	value			unit
	min	default	max	
value	0	45	65535	seconds
sms command to set	spw <i>value</i>			
sms command to view	spt?			
description	time interval between gps powerup and the actual measurement			

gps invalidate interval	value			unit
	min	default	max	
value	0	20	65535	seconds
sms command to set	spi <i>value</i>			
sms command to view	spt?			
description	time interval at the end of the measurement session during which the gps fix with the best PDOP is searched for			

spi must be a smaller value then sp1.

gps geofence trigger	value			unit
	min	default	max	
value	0	300	65535	meters
sms command to set	spg <i>value</i>			
sms command to view	spt?			
description	minimal distance between two consecutive fixes at which the device is considered non-stationary			

Heavy cloud coverage or large adjacent buildings can obscure or create multipath errors in the gps signal detection. It is thus advisable to not set the **spg** variable too low in order to minimize false-positives.

gprs loop interval	value			unit
	min	default	max	
value	180	900	65535	seconds
sms command to set	sml <i>value</i>			
sms command to view	smt?			
description	time interval between two gsm connection attempts (these are used to get tower id data and to receive and parse sms commands)			

each application needs a different trade-off between sms responsiveness and low power consumption.

gprs static tx interval	value			unit
	min	default	max	
value	180	3600	65535	seconds
sms command to set	smst <i>value</i>			
sms command to view	smt?			
description	time interval between two HTTP POSTs when the device is stationary (see spg 4.1.4)			

gprs moving tx interval	value			unit
	min	default	max	
value	180	600	65535	seconds
sms command to set	smmt <i>value</i>			
sms command to view	smt?			
description	time interval between two HTTP POSTs when the device is moving (see spg 4.1.4)			

all the timings in this section are used for scheduling the respective actions. the final timestamps received in the web reports will not follow exactly the values entered. this happens because the device tries to get the best fix possible in the **spi** interval based on PDOP measurements and also because most actions performed are dependent on both outside factors (how quickly a good fix is received from the gps, mobile service availability, how many sms actions need to be executed) as well as local interactions (the gprs modem can only perform one task at any given time).

4.2 Hardware

4.2.1 Absolute maximum ratings

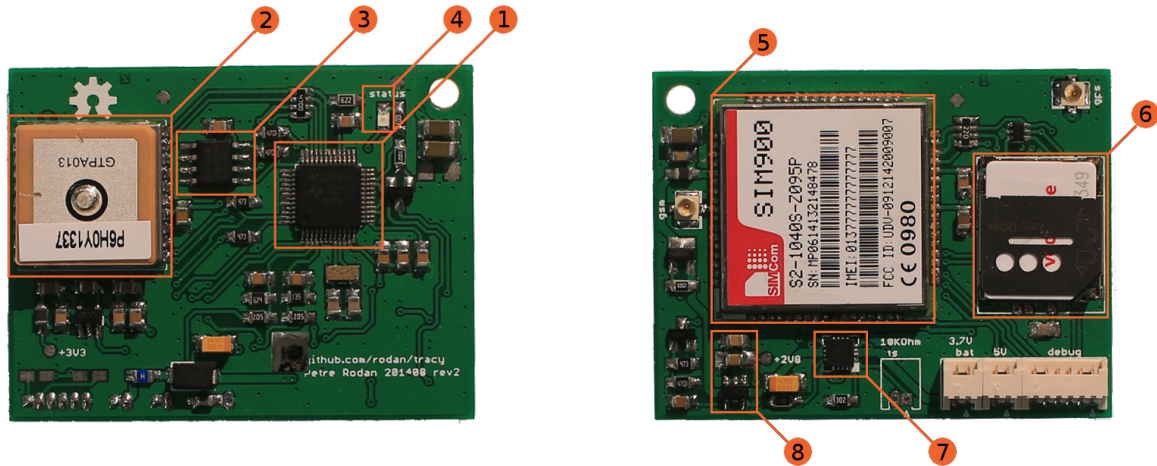
			value		unit
			min	max	
V_I	input voltage	IN (with respect to VSS)	-0.3	28	V
		VBAT (with respect to VSS)	-0.3	5	V
V_{IO}	IO voltage thru the debug connector		-0.3	3.1	V
T_{stg}	storage temperature		-65	150	°C
T_{op}	operating temperature		0	125	°C

Stresses above those listed in absolute maximum ratings may cause permanent damage to the device. This is a stress rating only, functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

4.2.2 Electrical characteristics

		value			unit
		min	typ	max	
V_I	input voltage range	4.35	5	6.4	V
I_I	input current	X	X	1.5	A
V_{BAT}	battery voltage range	3.5		4.1	V
I_{BAT}	battery current on sim900 poweron			3200	mA
I_{BAT}	battery current while sim900 tx			1600	mA
I_{BAT}	battery current while gps on		25	45	mA
I_{BAT}	battery current while sleeping	70		75	μA
I_{CHG}	battery charging current ¹		0.3		A
V_{IO}	IO voltage thru the debug connector	0		2.8	V

4.2.3 Components



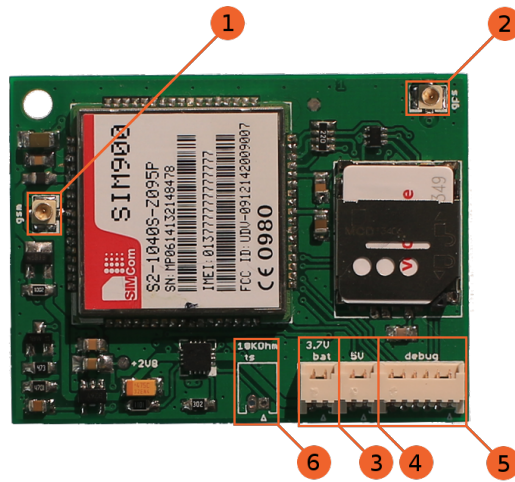
- 1 MSP430F5510 mixed signal microcontroller

- 16bit RISC architecture with 32KB flash and 4KB of SRAM


¹The charging current is configurable by changing the value of R11. See the module schematic and the bq24072 datasheet for details


- low supply voltage - 2.8V used in this implementation
 - ultra-low power consumption
 - 10bit ADC - both the battery voltage and input voltage levels are kept in check
 - 4 16bit timers with interrupt generation - used to synchronize all processes
 - real-time clock - keeps calendar data and the clock when gps signal is missing
 - port mapping controller - allows digital functions to be switched between different ports
 - watchdog, brown out reset - ensures the system runs in stable parameters
- 2 MT3339 MediaTek GPS chipset
- tiny gps module that can track 22 satellites at any given moment
 - even if the module is mostly shut down to conserve power, the backup voltage is always provided thus ensuring a fast time-to-fix when started up
 - automatic antenna switching detects external antennas
- 3 FM24CL64B-G F-RAM IC
- 4 status LED
- additional 8kbytes of F-RAM for data buffering and storage
 - very high efficiency LED that uses 28 μA while lit
 - very faint blips every two seconds while gps is on and there is no usable gps signal
 - less faint blips every two seconds while gps is on and gps signal is fine
 - full brightness for 2 seconds while modem powers up
- 5 SIM900 quad-band GSM/GPRS engine
- GSM 850MHz, EGSM 900MHz, DCS 1800MHz, PCS 1900MHz frequencies are supported
 - a dedicated user-provided micro SIM is needed to connect to the mobile phone network of choice. M2M SIMs also work since voice is never used in this application.
- 6 SIM card MICRO form factor SIM card holder
- 7 BQ24072 power path management IC
- allows charging of the Li-Ion/LiPo cell from a USB friendly 5V source
 - maximum charging current is limited to 300mA
- 8 2.8V rail the only power rail that remains on during microcontroller standby intervals. it can be used to power low voltage sensors/accelerometers/etc via the programming connector.

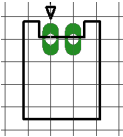
4.2.4 Connectors

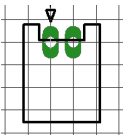


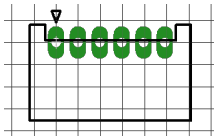
for all pin connectors, pin 1 is marked by a small triangle in the silkscreen

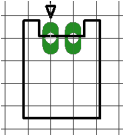
① gsm antenna	
connector	UFL
	
silkscreen	gsm
use	it is mandatory to connect the enclosed antenna via this connector
pin assignments	
shroud	GND
center	signal

② gps antenna	
connector	UFL
	
silkscreen	gps
use	the gps module provides a small patch antenna but one could improve signal reception by using an external one connected here
pin assignments	
shroud	GND
center	signal

3 LiPo/Li-Ion 1 cell battery	
connector footprint	Molex 053048-0210 
silkscreen	3.7V bat
use	1 cell (3.7V) rechargeable Li-Ion or Li polymer battery
pin assignments	
1	GND
2	+3.7V

4 5V DC input	
connector footprint	Molex 053048-0210 
silkscreen	5V
use	5V DC either from a USB connection or a DCDC converter. used only to charge the included battery with a current up to 300mA
pin assignments	
1	GND
2	+5V

5 programming/debug connector	
connector footprint	Molex 053048-0610 
silkscreen	debug
use	pins 3-6 are used for TI's spy-bi-wire protocol in order to program or to debug the microcontroller. pins 1 and 2 can have a custom assignation
pin assignments	
1	P4.1 - context dependent microcontroller port
2	P4.0 - context dependent microcontroller port
3	2.8V - VCC OUT
4	TEST
5	RST
6	GND

6 battery thermistor	
connector footprint	Molex 053048-0210 
silkscreen	10K0hm ts
use	some battery packs provide this component as a charge ending protection. by default this connector is unpopulated and if it's use is required the 10K resistor located on the other side of the PCB needs to be removed
pin assignments	
1	10K NTC thermistor
2	10K NTC thermistor (also GND)

