THINGSLOG

Command line guide

Low power mobile data logger

ThingsLog LPMDL-1101 GSM / GPRS Data logger

11 Magnaurska shkola str., fl. 3, office 315 (Hightech Business Park, BIC IZOT),

1784, Sofia, Bulgaria

Phone (+359) 875 32 80 70• Email: info@thingslog.com

Access to CLI console port

To access the command line interface you have to connect with an USBtoTTL cable to the logger console (UART) port.

LPMDL-1101-1-reed

To access the console port you have to take out the logger from the enclosure turn it upside down and connect to the 3 pins on the other side of the PCB.

The inputs are GND, TX and RX as it is shown on the picture.

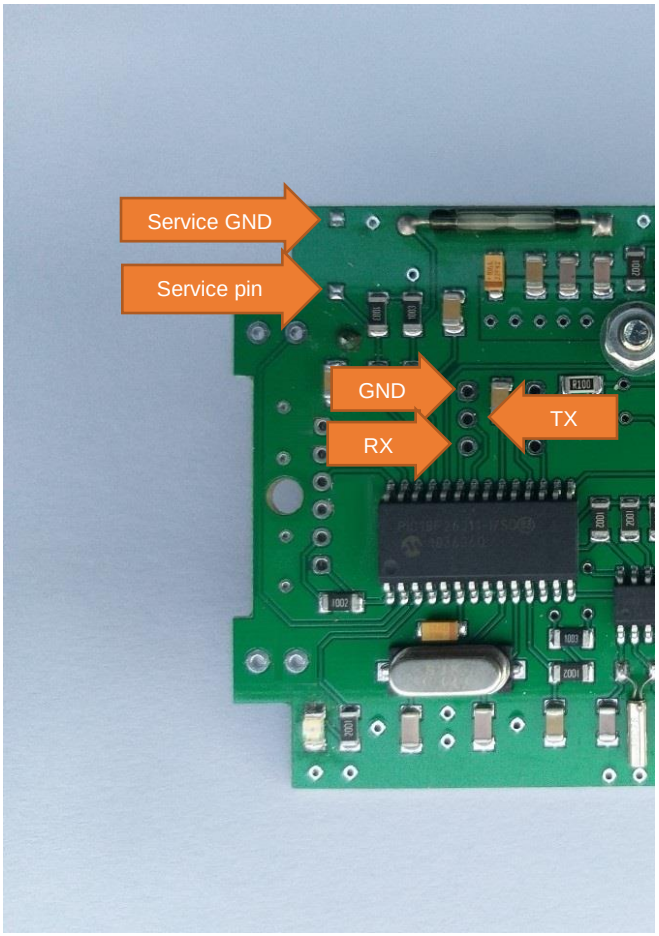

Figure 1 UART & service mode pins in 2 channel data loggers (v.3.0)

LPMDL-1101-2-reed/2-s0,1-reed-1-s0

Figure 2 UART & service mode pins in LPMDL-1101-2-reed

To put the device into service mode you need to short circuit the two pins close to the reed switch, see on the picture the “Service pin” and “Service GND”.

LPMDL-1101-with analog ports

To access the console port on the loggers you have to connect to the following 3 pins on the PCB.

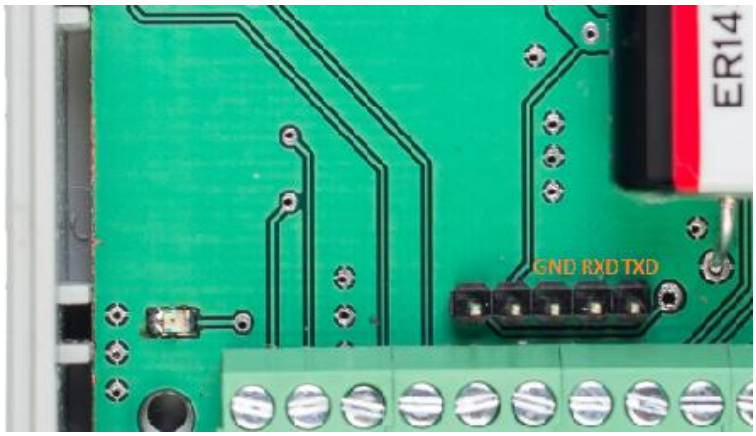

Figure 3 UART & service pins in LPMDL-1101-2-analog

Note: RX on the cable must be connected to TX on the board and RX on the cable must be connected with TX on the board.

CLI commands

There are different commands that can be used to configure the GPRS device when it is in service mode. In order to do this configuration it is necessary to put device in service mode using the service button.

Enter in base service mode

1. Connect to the debug UART port using baud rate 57600
2. Press the service button
3. Verify that the following message appear in UART console: "service # ". This indicates that the device is in the base service mode

Example

```
service #
```

Enter in gprs_service mode

1. Execute `extension` command. This will enter in device (LoRA, GPRS, NBIoT) specific sub-menu
2. Verify "gprs_service # " prompt appears.

Example

```
service # extension
gprs_service #
```

Exit the gprs_service mode and enter in base service mode

1. Verify you are in gprs_service mode - the prompt is "gprs_service # "
2. Execute `exit` command to exit "gprs_service" mode and to enter the base service mode
3. Verify "service # " prompt for the base service mode appear

Example

```
gprs_service # exit
service #
```

Set collecting server address and port

GSM data loggers are using by default TCP communication on port 4444.

1. Verify you are in gprs_service mode - the prompt is "gprs_service # "
2. Execute command `set_addr_port`
3. Verify "address=" prompt appear

4. Enter the new server address and press enter
5. Verify "port=" prompt appear
6. Enter the new server port and press enter
7. Execute `resolveIp` to set DNS name resolution 8 Verify that resolveIp= appears. Set it 1 to resolve address by DNS or 0 to disable resolving (address should be an IP address).
8. Verify "done" appear

Example:

```
service # extension
gprs_service # set_addr_port
address=10.10.10.10
port=4444
resolve_ip=1
done
gprs_service #
```

Get collecting server address and port

1. Verify you are in gprs_service mode - the prompt is "gprs_service # "
2. Execute command `get_addr_port`
3. Verify "address=<some_ip>" appear
4. Verify "port=<some_port>" appear

Example:

```
service # extension
gprs_service # get_addr_port
address=10.10.10.10
port=4444
done
gprs_service #
```

Set APN

1. Execute command `set_apn`
2. Verify "apn=" prompt appears
3. Enter the new APN and press enter
4. Verify "done" appear

Example:

```
service # extension
gprs_service # set_apn
apn=some.operator.apn
done
gprs_service #
```

- Get APN

1. Execute command `get_apn`
2. Verify "apn=some.operator.apn" appear

Example:

```
service # extension
gprs_service # get_apn
apn=some.operator.apn
done
gprs_service #
```

Set 2G network attachment timeout

Network attachment delay is the time for which the logger will wait to be recognized and paged by the mobile network.

1. Navigate to extension mode
2. Execute command `set_net_att_delay`
3. Verify "net_att_delay=some.delay " appears.
4. Enter the delay value in milliseconds (default is 35000 ms) e.g 35 seconds.
5. Don’t forget to store the configuration

Example:

```
service # extension
gprs_service # set_net_att_delay
net_att_delay=some.delay
done
gprs_service #
```

Get 2G network attachment timeout

1. Navigate to extension mode
2. Execute command `get_net_att_delay`
3. Verify "net_att_delay=some.delay in ms" appears

Example:

```
service # extension
nbmodem_service # get_net_att_delay
net_att_delay=35000
done
```

Set ADC read time

ADC reading time is a parameter that is valid only if you have a logger with analog ports. It should be equal or bigger than the analog sensor setting time. Setting time is the time required by the sensor placed in its particular environment to produce a correct measurement and output it on its

analog port. The adc reading time is in ms. Sensors suitable for low power measurements have short adc read times (10-20ms).

1. Execute command `set_adc_read_time`
2. Verify "adc_read_time=" appears

```
ST=2018-09-03 11:50:56
WKUP 2018-09-03 11:50:56
Service mode triggered
Service
Type help for help
service # adc_read_time
adc_read_time: 400
service # set_adc_read_time
adc_read_time=2000
Done
service # adc_read_time
adc_read_time: 2000
service #
```

Store configuration

1. Verify you are in base service mode - the prompt is "service # "
2. Execute "store" command to save the configurations
3. Verify "service # " prompt appear

Example:

```
service # store
service #
```

Load configuration

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `load` command to load the configurations
3. Verify "service # " prompt appear

Example:

```
service # load
service #
```

Get device number

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `number` command to get the device number
3. Verify "number=00000001" hex device number appear

Example:

```
service # number
number=00000001
service #
```

Get firmware version

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `version` command to get the firmware version
3. Verify "version=<some_version>" appear

Example:

```
service # version
version=0x0604bbc1
service #
```

Get date

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `date` command to get the current date
3. Verify "date=<some_date>" appear

Example:

```
service # date
date=2017-04-01 12:40:00
service #
```

Set date

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `set_d` command to set the device number
3. Verify "date=" prompt appear
4. Enter the new date in format YYYY-MM-dd HH:mm:ss
5. Verify "done" appear

Example:

```
service # set_d
Date=2017-04-01 12:40:00
Done
service #
```

Get current initial config

1. Verify you are in base service mode - the prompt is "service # "
2. Get current config by executing `configure`.

Example:

```
*** Config ***
2018-09-03 12:03:04
-----------
digits1=8
digits2=8
count_init1=1
count_init2=1
record_period=0
every=1
counts_threshold=300
sensors=01: CNT1
-----------
```

Description of configuration parameters

Name	Default Value	Possible values	Description
digits1	8	• Min value: 1 • Max value: 9	Total number of countable digits on the display of the meter for the first pulse input
digits2	8	• Min value: 1 • Max value: 9	Total number of countable digits on the display of the meter for the second pulse input
count_init1	1	Max value depends on the digits1 value	The value of the meter display of the first pulse input
count_init2	1	Max value depends on the digits1 value	The value of the meter display of the second pulse input
record_period	0	• 0 – MINUTES • 1 – HOURS	The value of the record period 0 mean minutes
every	1	• Min value: 1 • Max value: 127	The value in record period (1 - every 1 minute
counts_threshold	300	• Min value: 1 • Max value: 256	How many counters to keep in memory
sensors	01	Mask of 4 bits • 0x000001 – Pulse sensor 1 • 0x000010 – Pulse sensor 2 • 0x000100 – Current sensor 1 • 0x001000 – Current sensor 2 • 0x010000 – On/Off sensor 1 • 0x100000 – On/Off sensor 2	Which input is active, 01 means the first pulse input

Set initial config

1. Verify you are in base service mode - the prompt is "service # "
2. Execute `set_conf` and enter the configuration menu

```
service # set_conf
config # help
commands:
 help
 set_count_init1
 set_count_init2
 set_digits1
 set_digits2
 set_rec_period
 set_every
 set_counts_threshold
 set_sensors
 save
 exit
```

3. Set inputs reading frequency. Execute `set_every`

```
config # set_every
set_every= 3
```

4. Enable sensor input port

```
config # set_sensors
sensors= 1
```

5. Set counts threshold (the number of counters to be gathered prior transmission attempt). Execute `set_counts_threshold` If the transmission is successful all counters will be deleted from memory. If transmission fails counters will be kept in memory and will be re-transmitted on next attempt. If memory got full the oldest counters will be deleted first.

```
config # set_counts_threshold
counts_threshold= 100
```

6. Set counter value. Has to be set all meaningful digits up to the magnet pointer. Execute `set_count_init1` for the first input and `set_count_init2` for the second.

```
config # set_count_init1
count_init1= 123
config # set_count_init2
count_init2= 234
```

7. Save and apply the configuration.

Execute `save`

```
config # save
Save config
Applying config
Config Counters
rec_conf, size=102, rec_size=2, buff_size=206
Alarm enabled
service #
```