EFENTO

Sensor test dongle

2021-06-30

Table of Contents

Chapter - 1 General information 3

Chapter - 2 Using the dongle 3

ET0-200 - Using the dongle in console 3
ET0-202 - Using the dongle with Python....3
Chapter - 3 Available commands 4

ET0-118 - Scan....4
ET0-119 - Get software version....4
ET0-120 - Enter Bootloader 5
ET0-121 - Data Transfer....5
ET0-122 - Change measurement period....5
ET0-123 - Get measurement period....5
ET0-124 - Set sensor’s time....6
ET0-125 - Get sensor’s time 6
ET0-126 - Restore defaults 6
ET0-127 - Get types of sensor’s channels 6
ET0-128 - Get sensor models 6
ET0-129 - Get factory descriptor 6
ET0-130 - Get self test results....7
ET0-131 - Get battery status 7
ET0-132 - Reset battery status 7
ET0-133 - Set sensor name 7
ET0-134 - Get sensor name....7
ET0-135 - Set sensor calibration date....7
ET0-136 - Set encryption key....8
ET0-137 - Get encryption counter 8
ET0-138 - Set Bluetooth TX power....8
ET0-139 - Get Bluetooth TX power 8
ET0-140 - Set Bluetooth turn off time....8
ET0-141 - Get cellular status....8
ET0-142 - Get cellular module statistics....8
ET0-143 - Enter installation mode....9
ET0-144 - Exit installation mode 9
ET0-145 - Set APN....9
ET0-146 - Get APN 9
ET0-147 - Set PLMN 10
ET0-148 - Get PLMN....10
ET0-149 - Trigger the communication with server....10
ET0-150 - Set server configuration 10
ET0-151 - Get server configuration 11
ET0-152 - Set cloud token 11
ET0-153 - Get cloud token 12
ET0-154 - Set transmission interval 12
ET0-155 - Get transmission interval....12
ET0-156 - Set transfer limit 12
ET0-157 - Get transfer limit....12
ET0-158 - Set supervision period 13
ET0-159 - Get supervision period 13
ET0-162 - Set rule....13

ET0-163 - Get rule....14
ET0-180 - Collect memory statistics....15
ET0-181 - Get memory statistics....15
ET0-182 - Get runtime errors....15
ET0-184 - Set CoAP endpoints....15
ET0-185 - Get CoAP endpoints....16
ET0-195 - Ping notification event....16

Chapter - 1 General information

USB dongle allows users to communicate with Efento sensors over Bluetooth. All sensors Bluetooth features are available using the dongle.

Chapter - 2 Using the dongle

- All the commands start with “sensor” (e.g. “sensor scan 28:2C:02:4F:FF:90”)
- If you want to repeat the recent commands, press the arrow up key on the keyboard until you find the command you want to repeat
- If you want to use another command with the same sensor, can use “-” dongle will put in its place the recently used sensor’s MAC address and PIN (e.g. “sensor scan - -”)
- Sensor’s MAC address needs to have “:” between each segment
- If you press TAB key while writing a command you will list all the commands, starting with the keyed in letters (e.g. typing “get*” and pressing TAB will list all the commands starting with “get*”)
- If you want to list all the available commands along with the list of their parameters, type “sensor”
- Dongle’s response is a YAML message, which always contains two fields: serial number and result

ET0-200 - Using the dongle in console

Dongle can be used with any operating system. Once connected to the computer connect to it over COM port.

- Windows: we recommend using PuTTY and open a serial connection with speed set to 115200
- MacOS: use “screen” command to connect to the dongle on the selected serial port

ET0-202 - Using the dongle with Python

- Make sure you have Python 3 and pip installed
- install **pyyaml** using command:

```
pip install PyYAML(Windows)
pip install pyyaml(Linux)
```

- install **pyserial** using command:

```
pip install pyserial (both Windows and Linux)
```

Chapter - 3 Available commands

All the variables in “<>” indicate the name of the variable that needs to be used in the command. For instance if the command description contains “<MAC_ADDRESS>“ it means that this part of the command has to be replaced to the sensor's mac address.

Note that some of the commands are only available for cellular sensors.

Commands listed below are available for sensors with software version 6.0 or higher.

ET0-118 - Scan

Command used to perform scanning for BLE device around the dongle. Return data from the BLE advertisement mode of a sensor with selected serial number

```
sensor scan <MAC_ADDRESS>
```

Optional parameters:

-t<TIMEOUT_VALUE> - defines the time of the scanning (e.g. 10 seconds)

-d - converts the measurement values to decimal (by default they are in HEX)

Examples

```
sensor scan 28:2C:02:4F:FF:90
sensor scan 28:2C:02:4F:FF:90 –t10
sensor scan 28:2C:02:4F:FF:90 –t10 –d
```

ET0-119 - Get software version

Returns sensor’s software version

sensor get_sw_version <MAC_ADDRESS>

ET0-120 - Enter Bootloader

Enters devices bootloader in order to perform software update

sensor enter_bootloader <MAC_ADDRESS> <BOOTLOADER_PIN>

ET0-121 - Data Transfer

Returns measurements from sensor's memory for the selected time period. Time period is defined by two epoch times.

sensor data_transfer <START_TIMESTAMP> <END_TIMESTAMP>

Example

sensor data_transfer 28:2C:02:4F:FF:90 1608070309 1608073937

ET0-122 - Change measurement period

Sets new measurement's period base and factor. Measurement period = base*factor (e.g base = 10s, factor = 6, measurement period will be set to 60s. The value of "base" is in seconds, the value of "factor" is a number. Changing the measurement period does not clear sensor's memory as long as the optional "-r"parameter is not used

sensor change_period <MAC_ADDRESS> <PIN> <BASE> <FACTOR>

Optional parameters:

-r - removes all the measurements from sensor's memory

Examples

sensor change_period 28:2C:02:4F:FF:90 1111 10 6
sensor change_period 28:2C:02:4F:FF:90 1111 10 6 -r

ET0-123 - Get measurement period

Returns the current measurement's period "base" and "factor" values. Measurement period = base*factor

sensor get_period <MAC_ADDRESS>

ET0-124 - Set sensor’s time

Sets the time of the sensor's internal clock. Timestamp in epoch time

sensor set_time <MAC_ADDRESS> <PIN> <TIME_STAMP>

Example

sensor set_time 28:2C:02:4F:FF:90 1111 1608070309

ET0-125 - Get sensor’s time

Returns the current time of the sensor's internal clock.

sensor get_time <MAC_ADDRESS> <PIN>

ET0-126 - Restore defaults

Restores the default (factory) settings of the sensor.

sensor restore_defaults <MAC_ADDRESS> <PIN>

ET0-127 - Get types of sensor’s channels

Returns the types of sensor’s channels (values measured by the sensor)

sensor get_channel_types <MAC_ADDRESS>

ET0-128 - Get sensor models

Returns the models of the sensors assigned to each slot

sensor get_channel_models <MAC_ADDRESS>

ET0-129 - Get factory descriptor

Returns the information about the hardware version and variant

sensor get_factory_descriptor <MAC_ADDRESS> <PIN>

ET0-130 - Get self test results

Returns the results of the self test performed on the device's boot up

sensor get_selftest_data <MAC_ADDRESS> <PIN>

ET0-131 - Get battery status

Returns the battery status including the voltage and the battery reset date

sensor get_battery_status <MAC_ADDRESS> <PIN>

ET0-132 - Reset battery status

Resets the battery status and sets the battery reset day. Battery rest timestamp in epoch time

sensor reset_battery_status <MAC_ADDRESS> <PIN> <RESET_TIMESTAMP>

ET0-133 - Set sensor name

Sets sensor's name. Name is used in Bluetooth advertisement frames

sensor set_name <MAC_ADDRESS> <PIN> <NAME>

ET0-134 - Get sensor name

Returns sensor's name

sensor_get_name <MAC_ADDRESS> <PIN>

ET0-135 - Set sensor calibration date

Sets the date, when the sensor was calibrated (date1) and the date, when it should be calibrated again (date2). Dates are epoch times

sensor write_calibraiton_date <MAC_ADDRESS> <MANUFACTURER_KEY> <DATE1> <DATE2>

Example

sensor write_calibraiton_date 28:2C:02:4F:FF:90 111111 1608070309 1609070309

ET0-136 - Set encryption key

Sets encryption key used to encrypt the BLE communication

sensor set_encryption_key <MAC_ADDRESS> <ENCRYPTION_KEY>

ET0-137 - Get encryption counter

Returns the value of the encryption counter

sensor get_encryption_counter <MAC_ADDRESS>

ET0-138 - Set Bluetooth TX power

Sets the value of the Bluetooth TX power. Value in dBm. Maximum TX power value 4

sensor set_bluetooth_tx_power <MAC_ADDRESS> <PIN> <TX_POWER>

ET0-139 - Get Bluetooth TX power

Returns the value of Bluetooth TX power

sensor get_bluetooth_tx_power <MAC_ADDRESS> <PIN>

ET0-140 - Set Bluetooth turn off time

Sets time period, after which BLE interface will switch off. BLE will automatically turn off after this time period each time the device is turned on. Turn off time in seconds - e.g. if set to 600, BLE will turn off 600 seconds after the device boots up

sensor set_bluetooth_turn_off_time <MAC_ADDRESS> <PIN> <TURN_OFF_TIME>

ET0-141 - Get cellular status

Returns the status of the cellular communication (signal strength, registration status, etc.)

sensor get_cellular_network_status <MAC_ADDRES> <PIN>

ET0-142 - Get cellular module statistics

Returns the statistics of the cellular module (tx power, ecl, snr, pci, rsrq, etc.)

sensor get_cellular_module_stats <MAC_ADDRES> <PIN>

ET0-143 - Enter installation mode

Enters the installation mode

sensor enter_installation_mode <MAC_ADDRESS> <PIN>

ET0-144 - Exit installation mode

Exits the installation mode (if not used, sensor automatically exits the installation mode 5 minutes after it entered it)

sensor exit_installation_mode <MAC_ADDRESS> <PIN>

ET0-145 - Set APN

Sets the APN value. To set APN value to “auto” use the optional “-a” parameter

sensor set_apn <MAC_ADDRESS> <PIN> <APN_VALUE>

Optional parameters

-a - used to set APN value to “auto”

Examples

set APN value to “test.apn”

sensor set_apn 28:2C:02:4F:FF:90 1111 test.apn

set APN value to “auto”

sensor set_apn 28:2C:02:4F:FF:90 1111 -a 0

ET0-146 - Get APN

Returns the value of the APN

sensor get_apn <MAC_ADDRESS> <PIN>

ET0-147 - Set PLMN

Sets the value of PLMN. To set the value to “auto” use the optional “-a” parameter

sensor set_plmn <MAC_ADDRESS> <PIN> <PLMN>

Optional parametrers

-a - used to set PLMN value to "auto"

Examples

set PLMN value to 12345

sensor set_plmn 28:2C:02:4F:FF:90 1111 12345

set PLMN value to "auto"

sensor set_plmn 28:2C:02:4F:FF:90 1111 -a

ET0-148 - Get PLMN

Returns the value of PLMN

sensor get_plmn <MAC_ADDRESS> <PIN>

ET0-149 - Trigger the communication with server

Triggers the communication with server over cellular network

sensor trigger_communication_with_server <MAC_ADDRESS> <PIN>

ET0-150 - Set server configuration

Sets data and update servers configuration

Data server:

sensor set_server_configuraiton <MAC_ADDRESS> <PIN> 0 <SERVER_IP_ADDRESS> <PORT>

Update server:

sensor set_server_configuraiton <MAC_ADDRESS> <PIN> 1 <SERVER_IP_ADDRESS> <UDP_PORT> <COAP_PORT>

Examples

Set the data server

sensor set_server_configuraiton 28:2C:02:4F:FF:90 1111 0 127.0.0.1 5683

Set the update server

sensor set_server_configuraiton 28:2C:02:4F:FF:90 1111 1 127.0.0.1 6000 5683

ET0-151 - Get server configuration

Returns the configuration of data and update servers

Data server:

sensor get_server_configuraiton <MAC_ADDRESS> <PIN> 0

Update server:

sensor get_server_configuraiton <MAC_ADDRESS> <PIN> 1

ET0-152 - Set cloud token

Sets the cloud token value (can be used to disable cloud token, set cloud token to user specified value or set cellular module's IMEI as cloud token)

Cloud token value

sensor set_cloud_token <MAC_ADDRESS> <PIN> 1 <TOKEN_VALUE>

Set IMEI as cloud token

sensor set_cloud_token <MAC_ADDRESS> <PIN> 2 0

Do not set cloud token

sensor set_cloud_token <MAC_ADDRESS> <PIN> 255 0

ET0-153 - Get cloud token

Returns the value and type of the cloud token

sensor get_cloud_token <MAC_ADDRESS> <PIN>

ET0-154 - Set transmission interval

Sets sensor’s transmission and ACK interval. To set ACK interval to always set its value to "0xFFFFFFFF"

sensor set_transmission_config <MAC_ADDRESS> <PIN> <TRANSMISSION_INTERVAL> <ACK_INTERVAL>

ET0-155 - Get transmission interval

Returns sensor’s transmission and ACK intervals

sensor get_transmission_config <MAC_ADDRESS> <PIN>

ET0-156 - Set transfer limit

Sets the transfer limit and transfer limit timer

sensor set_transfer_limit <MAC_ADDRESS> <PIN> <TRANSFER_LIMIT> <TRANSFER_LIMIT_TIMER>

ET0-157 - Get transfer limit

Returns the transfer limit and transfer limit timer

sensor get_transfer_limit <MAC_ADDRESS> <PIN>

ET0-158 - Set supervision period

Sets supervision period. Supervision period value should be in seconds. To disable set to "0xFFFFFFFF"

sensor set_supervision_period <MAC_ADDRESS> <PIN> <SUPERVISION_PERIOD>

ET0-159 - Get supervision period

Returns supervision period

sensor get_supervision_period <MAC_ADDRESS> <PIN>

ET0-162 - Set rule

Creates a new rule configuration / updates a rule with selected rule ID.

```
sensor set_rule <MAC_ADDRESS> <PIN> <RULE_ID> <CHANNEL_MASK> <CONDITION>
<ACTION> [<PARAM1> <PARAM2> <PARAM3> <PARAM4> <PARAM5>]
```

- RULE_ID - Id of the rule. Up to 12 rules can be defined. Range [0:11]
- CHANNEL_MASK - Channels to which the rule is assigned. One rule can be assigned to multiple channels as long as those are of the same type. Bit mask on bits [0:5]. E.g. To assign the rule for channel 1: "000001", to assign rule to channels 2 and 4: "001010"
- CONDITION - Condition to be checked by the device. If the condition is true, an action is triggered. Possible values:
    - 0 - Unspecified
    - 1 - Disabled
    - 2 - High Threshold
    - 3 - Low threshold
    - 4 - Differential threshold
    - 5 - Change of state (Binary sensors only)
- ACTION - Action to be triggered. Currently the only possible action is to trigger the transmission. Other actions will be available in next SW releases. Possible values
    - 0 - Unspecified
    - 1 - Trigger the transmission
- PARAM1..5 - parameters of CONDITION. Available params depend on the type of condition

CONDITION: High Threshold

Available for continuous sensors only. If the measurement (or average from a few measurements) is over the threshold, an action is triggered.

- PARAM1 - Threshold value. Must match channel type
- PARAM2 - Hysteresis value. Must much channel type. Set to "0" to disable
- PARAM3 - Average calculation mode:

    - 1 - moving average (a1=(n1+n2+n3)/3, a2=(n2+n3+n4)/3, etc.)
    - 2 - window mode (a1=(n1+n2+n3)/3, a2=(n4+n5+n6)/3, etc.)
- PARAM4 - Number of measurements for average value calculating. E.g PARAM4 equals 3, average value from three samples will be calculated and compared to the threshold value.
- PARAM5 - Type of measurement (as described in MeasurementType).

CONDITION: Low Threshold

Available for continuous sensors only. If the measurement (or average from a few measurements) is below the threshold, an action is triggered.

- PARAM1 - Threshold value. Must match channel type
- PARAM2 - Hysteresis value. Must much channel type. Set to "0" to disable
- PARAM3 - Average calculation mode:
    - 1 - moving average (a1=(n1+n2+n3)/3, a2=(n2+n3+n4)/3, etc.)
    - 2 - window mode (a1=(n1+n2+n3)/3, a2=(n4+n5+n6)/3, etc.)
- PARAM4 - Number of measurements for average value calculating. E.g PARAM4 equals 3, average value from three samples will be calculated and compared to the threshold value.
- PARAM5 - Type of measurement (as described in MeasurementType).

CONDITION: Differential Threshold

Continuous sensors only. If the absolute value of the difference between the last value sent to the server and the measurement value (or average from a few measurements) is greater or equal to the value of the threshold set, an action is triggered.

- PARAM1 - Threshold value. Must match channel type
- PARAM2 - Average calculation mode:
    - 1 - moving average (a1=(n1+n2+n3)/3, a2=(n2+n3+n4)/3, etc.)
    - 2 - window mode (a1=(n1+n2+n3)/3, a2=(n4+n5+n6)/3, etc.)
- PARAM3 - Number of measurements for average value calculating. E.g PARAM3 equals 3, average value from three samples will be calculated and compared to the threshold value.

Examples

Set a rule, which will trigger the transmission, if the temperature measured on channel 1 is over 10 C. Rule will be assigned to Rule_ID = 0

sensor set_rule 28:2C:02:4F:FF:90 1111 0 000001 2 1 100 0 1 1 1
Set a rule, which will trigger the transmission, if the humidity measured on channel 2 or 3 is below 40%. Rule will be assigned to Rule_ID = 1

sensor set_rule 28:2C:02:4F:FF:90 1111 0 000110 3 1 40 0 1 1 2

ET0-163 - Get rule

Returns a rule configuration for the selected rule ID.

```
sensor get_rule <MAC_ADDRESS> <PIN> <RULE_ID>
```

ET0-180 - Collect memory statistics

Triggers a restart of collecting memory statistics.

```
sensor collect_memory_statistics <MAC_ADDRESS> <PIN>
```

ET0-181 - Get memory statistics

Returns a memory statistics.

```
sensor get_memory_statistics <MAC_ADDRESS> <PIN>
```

ET0-182 - Get runtime errors

Returns a table of runtime errors, and can clear it.

```
sensor get_runtime_errors <MAC_ADDRESS> <PIN>
```

Optional parameters:

-r - removes all errors from the table

Examples

```
sensor get_runtime_errors 28:2C:02:4F:FF:90 1111
sensor get_runtime_errors 28:2C:02:4F:FF:90 1111 -r
```

ET0-184 - Set CoAP endpoints

Sets chosen CoAP endpoint (ASCII string).

Data endpoint:

sensor set_coap_endpoints <MAC_ADDRESS> <PIN> 0 <ENDPOINT>

Configuration endpoint:

sensor set_coap_endpoints <MAC_ADDRESS> <PIN> 1 <ENDPOINT>

Device info endpoint:

sensor set_coap_endpoints <MAC_ADDRESS> <PIN> 2 <ENDPOINT>

Time endpoint:

sensor set_coap_endpoints <MAC_ADDRESS> <PIN> 3 <ENDPOINT>

ET0-185 - Get CoAP endpoints

Returns chosen CoAP endpoint.

Data endpoint:

sensor get_coap_endpoints <MAC_ADDRESS> <PIN> 0

Configuration endpoint:

sensor get_coap_endpoints <MAC_ADDRESS> <PIN> 1

Device info endpoint:

sensor get_coap_endpoints <MAC_ADDRESS> <PIN> 2

Time endpoint:

sensor get_coap_endpoints <MAC_ADDRESS> <PIN> 3

ET0-195 - Ping notification event

Command used to create a ping notification event. The result of the request will be transmitted on Notify characteristic.

sensor ping_notification_event <MAC_ADDRESS> <PING_NUMBER>

Example

sensor ping_notification_event 28:2C:02:4F:FF:90 1