

---

**COOPER**

**HARDWARIO s.r.o.**

**Oct 13, 2020**



# CONTENTS:

- 1 Overview** **1**
- 1.1 Integrated Sensors . . . . . 1
- 1.2 Basic Features . . . . . 1
- 1.3 Applications . . . . . 2
  
- 2 CO2 Sensor Calibration** **3**
  
- 3 AT Commands** **5**
- 3.1 Interfacing AT Commands . . . . . 5
- 3.2 Sensor AT Commands . . . . . 6
- 3.3 Dongle AT Commands . . . . . 11
- 3.4 Sensor URC Messages . . . . . 12
- 3.5 Dongle URC Messages . . . . . 12
  
- 4 Indices and tables** **15**



**OVERVIEW**

**COOPER** is a flexible sensor platform for the Internet-of-Things (IoT) applications. It has been designed to meet the highest quality standards for environmental monitoring, low-power operation from batteries and wireless communication. Customized sensor and firmware configuration is available on request including silicone strap color customization, logo printing (using colored UV printing technology) and high-speed CNC milling in the top cover. The device is powered from 3x AA Alkaline 1.5V cells and it can provide up to 3 years of service time from the battery installation. Service time is given by the sensor measurement frequency, chosen communication technology, and communication interval.

## 1.1 Integrated Sensors

- Acceleration
- Acoustic noise
- Altitude (sea level)
- Atmospheric pressure
- Ambient temperature
- Battery voltage
- CO<sub>2</sub> concentration
- Light intensity
- Motion detection (PIR)
- Relative air humidity
- VOC concentration

## 1.2 Basic Features

- Battery operation (3x AA 1.5V)
- 868 MHz ISM low-power radio (encrypted communication - AES-128)
- NB-IoT communication (optional)
- LoRa communication (optional)
- Sigfox communication (optional)
- Bluetooth Low Energy (optional)

- Push button with LED backlight (optional)
- Loudspeaker with speech output (optional)
- USB interface (with power support)
- Firmware updates over USB
- Dimensions (mm) 145(l) x 88(w) x 33(h)

### 1.3 Applications

- Environmental monitoring
- HVAC systems
- Facility management
- Smart home
- Schools
- Education
- Development kit for IoT

## CO2 SENSOR CALIBRATION

You can calibrate the sensor in the outdoors so it will set up its own lowest CO2 level to 400 ppm.

Procedure to start the calibration:

1. Press and hold the button for 6 seconds (button is located next to the USB connector).
2. Release the button.
3. The LED will start to blink every 5 seconds.
4. Place the Cooper outdoor to the fresh air.
5. The complete calibration takes 72 minutes.
6. Calibration is done when the LED stops blinking.





## AT COMMANDS

Both sensor device and gateway device can be controlled using AT commands over the USB interface. The USB interface provides virtual serial port (USB CDC) interface which is always available when the USB cable is plugged in.

These are the basic AT command configuration and usage rules:

- The device is further in AT commands context referred to as **MT** (Mobile Terminal).
- The host is further in AT commands context referred to as **TE** (Terminal Equipment).
- Serial port parameters are 115200 Bd, 8 data bits, no parity, 1 stop bit.
- From TE it is possible to use any new line delimiter combination combination of <CR> (0x0d) or <LF> (0x0a).
- The new line is always delimited from MT using <CR><LF>.
- MT supports the possibility to clear input buffer using <ESC> key (0x1b).
- MT supports the correction of the last character using <BS> key (0x08).
- Every AT command starts with the AT string prefix.
- Some commands are part of existing standards and inspired by existing equipment.
- Vendor-specific AT commands always start with AT\$.
- Every command is terminated with OK in case of success, or with ERROR in case of failure.
- All commands are case-sensitive and have to follow patterns exactly described below.
- URC messages (Unsolicited Result Codes) is a kind of message, which can appear from MT at any time (these are asynchronous events).
- Messages that appear on the serial port and start with the # character, represent logging messages, which can (like URC messages) appear at any time.

### 3.1 Interfacing AT Commands

There are many serial port terminals that can be used to interface AT commands. For example, on macOS and Linux, you can use **picocom**. On macOS this can be easily installed using Homebrew:

```
brew install picocom
```

Then you can start the program as:

```
picocom -b 115200 --omap crclrf --echo /dev/cu.usbserial-00001014
```

## 3.2 Sensor AT Commands

### 3.2.1 AT - Communication test

This command only serves the purpose to test communication with MT.

**Format**

AT

**Example**

Command:	AT
Response:	OK

### 3.2.2 AT&F - Restore configuration to factory defaults

This command puts the device configuration to factory default settings. It does not store the settings to non-volatile memory (see AT&W below).

**Format**

AT&F

**Example**

Command:	AT&F
Response:	OK

### 3.2.3 AT&W - Store configuration to non-volatile memory

This command saves the current configuration settings to a non-volatile (EEPROM) memory.

**Format**

AT&W

**Example**

Command:	AT&W
Response:	OK

### 3.2.4 ATI - Request product information

This command reads the compact product information of MT.

**Format**

ATI

**Example**

Command:	ATI
Response:	COOPER R1.1 0.1.0 OK

### 3.2.5 AT+CGMI - Request manufacturer identification

This command reads the manufacturer identification of MT.

#### Format

AT+CGMI

#### Example

```
Command: AT+CGMI
Response: HARDWARIO s.r.o.
         OK
```

### 3.2.6 AT+CGMM - Request model identification

This command reads the model identification of MT.

#### Format

AT+CGMM

#### Example

```
Command: AT+CGMM
Response: COOPER R1.1
         OK
```

### 3.2.7 AT+CGMR - Request revision identification

This command reads the serial number of MT.

#### Format

AT+CGMR

#### Example

```
Command: AT+CGMR
Response: 0.1.0
         OK
```

### 3.2.8 AT+CGSN - Read serial number identification

This command reads the serial number (id) of MT.

#### Format

AT+CGSN

#### Example

```
Command: AT+CGSN
Response: 0123456789012345
         OK
```

### 3.2.9 AT+CLAC - List available AT commands

This command lists all available AT commands.

#### **Format**

AT+CLAC

#### **Example**

Command:	AT+CLAC
Response:	AT
	AT&F
	...
	OK

### 3.2.10 AT\$CHANNEL - Set/read radio channel

This command allows to set or read radio channel. The supported range of channels is 0..19. The command affects configuration settings, which have to be permanently stored using the AT&W command.

#### **Format**

AT\$CHANNEL?

AT\$CHANNEL=<channel>

#### **Example**

Command:	AT\$CHANNEL?
Response:	\$CHANNEL: 0
	OK
Command:	AT\$CHANNEL=1
Response:	OK

### 3.2.11 AT\$KEY - Set encryption key (AES-128)

This command allows setting key for encrypted radio communication. It expects the key in the hexadecimal format (32 characters). The command affects configuration settings, which have to be permanently stored using the AT&W command.

#### **Format**

AT\$KEY=<key>

#### **Example**

Command:	AT\$KEY=f2e891014be3e94151c66249203e2246
Response:	OK

### 3.2.12 AT\$STATUS - Retrieve device status

This command retrieves the current device status.

#### **Format**

AT\$STATUS

**Example**

```

Command:  AT$STATUS
Response: $STATUS: "Acceleration",0.11,0.00,0.96
          $STATUS: "Altitude",321.8
          $STATUS: "CO2 Concentration"
          $STATUS: "Humidity",48.4
          $STATUS: "Illuminance",28
          $STATUS: "Orientation",1
          $STATUS: "Press Count",0
          $STATUS: "Pressure",97521
          $STATUS: "Sound Level",0
          $STATUS: "Temperature",24.64
          $STATUS: "VOC Concentration"
          $STATUS: "Voltage",0.01
          OK

```

**3.2.13 AT\$SEND - Send data**

This command send data immediately.

**Format**

```
AT$SEND
```

**Example**

```

Command:  AT$SEND
Response:  OK

```

**3.2.14 AT\$PULSE - Pulse LED**

This command pulses LED on MT for 3 seconds.

**Format**

```
AT$PULSE
```

**Example**

```

Command:  AT$PULSE
Response:  OK

```

**3.2.15 AT\$BEEP - Beep speaker**

This command run beep speaker on MT for 3 seconds.

**Format**

```
AT$BEEP
```

**Example**

```

Command:  AT$BEEP
Response:  OK

```

### 3.2.16 AT\$HELP - List help

This command lists AT command help.

#### **Format**

AT\$HELP

#### **Example**

Command:	AT\$HELP
Response:	AT&F Restore configuration to factory defaults AT&W Store configuration to non-volatile memory ... AT\$HELP Print this help OK

### 3.2.17 AT\$LOCK - Lock configuration

This command allows setting password and lock configuration to the read-only mode. It expects a parameter of up to 12 ASCII characters - all printable characters inside the quotation marks are allowed. The command affects configuration settings, which have to be permanently stored using the AT&W command.

#### **Format**

AT\$LOCK="**<password>**"

#### **Example**

Command:	AT\$LOCK="jzvuk5oTwBs6"
Response:	OK

### 3.2.18 AT\$UNLOCK - Unlock configuration

This command allows removing password and unlock configuration. It expects a parameter of up to 12 characters - all printable characters inside the quotation marks are allowed. The command affects configuration settings, which have to be permanently stored using the AT&W command.

#### **Format**

AT\$UNLOCK="**<password>**"

#### **Example**

Command:	AT\$UNLOCK="jzvuk5oTwBs6"
Response:	OK

### 3.2.19 AT\$CONFIG - Configuration

This command allows configuration some parameters. The command affects configuration settings, which have to be permanently stored using the AT&W command.

#### **Format**

AT\$CONFIG

AT\$CONFIG="**<name>**", **<value>**

**Parameters**

- Report Interval unit: seconds, default: 300, min: 30, max: 65535

**Example**

```
Command: AT$CONFIG
Response: $CONFIG: "Report Interval",300
         OK

Command: AT$CONFIG="Report Interval",600
Response: OK
```

## 3.3 Dongle AT Commands

The following AT commands are supported on the gateway device.

### 3.3.1 AT\$LIST - List nodes

This command prints whitelist of all the nodes.

**Format**

```
AT$LIST
```

**Example**

```
Command: AT$LIST
Response: $CONFIG: "Report Interval",300
         OK

Command: AT$LIST
Response: 0123456789012345, "Room 1 "
         5432109876543210, ""
         OK
```

### 3.3.2 AT\$ATTACH - Attach new node

This command adds a new node to the whitelist. The command affects configuration settings, which have to be permanently stored using the AT&W command.

**Format**

```
AT$ATTACH=<id>,<key>,"<alias>"
```

```
AT$ATTACH=<id>,<key>
```

**Example**

```
Command: AT$ATTACH=0123456789012345,f2e891014be3e94151c66249203e2246,"Room 1"
Response: OK
```

### 3.3.3 AT\$DETACH - Detach existing node

This command removes the existing node from the whitelist. The command affects configuration settings, which have to be permanently stored using the AT&W command.

**Format**

AT\$DETACH=<id>

**Example**

Command:	AT\$DETACH=0123456789012345
Response:	OK

### 3.3.4 AT\$PURGE - Remove all paired nodes

This command deletes all paired nodes. The command affects configuration settings, which have to be permanently stored using the AT&W command.

**Format**

AT\$PURGE

**Example**

Command:	AT\$PURGE
Response:	OK

## 3.4 Sensor URC Messages

### 3.4.1 \$BOOT - Device restart

This URC informs TE about the MT restart.

**Format**

\$BOOT

### 3.4.2 \$PRESS - Push button press

This URC informs about the push button press event. It provides a number of press events.

**Format**

\$PRESS: <count>

## 3.5 Dongle URC Messages

### 3.5.1 \$BOOT - Device restart

This URC informs TE about the MT restart.

**Format**



---

\$BOOT

### 3.5.2 \$RECV - Message from node

This URC provides extracted message information from the node.

#### Format

\$RECV: <p01>, <p02>, ...<p15>

#### Parameters

- <p01> rssi
- <p02> id
- <p03> sequence
- <p04> altitude
- <p05> co2\_conc
- <p06> humidity
- <p07> illuminance
- <p08> motion\_count
- <p09> orientation
- <p10> press\_count
- <p11> pressure
- <p12> sound\_level
- <p13> temperature
- <p14> voc\_conc
- <p15> voltage

---

**Note:** If some parameter is missing, it means it is invalid (sensor is not yet ready, or sensor is broken).

---



## INDICES AND TABLES

- genindex
- modindex
- search