

# MySensors Serial Protocol (1.4)

**Older Versions:** [➤ 1.3 \(/build/serial\\_api\\_13\)](#)

The serial protocol used between the Gateway and the Controller is a simple semicolon separated list of values. The last part of each "command" is the payload. All commands ends with a newline. The serial commands has the following format:

**node-id ; child-sensor-id ; message-type ; ack ;  
sub-type ; payload \n**

## Message Structure Elements

Message Part	Comment
<b>node-id</b>	The unique id of the node that sends or should receive the message (address)
<b>child-sensor-id</b>	Each node can have several sensors attached. This is the child-sensor-id that uniquely identifies one attached sensor
<b>message-type</b>	Type of message sent - See table below
<b>ack</b>	The ack parameter has the following meaning: Outgoing: 0 = unacknowledged message, 1 = request ack from destination node Incoming: 0 = normal message, 1 = this is an ack message
<b>sub-type</b>	Depending on messageType this field has different meaning. See tables below
<b>payload</b>	The payload holds the message coming in from sensors or instruction going out to actuators.

### ⚠ The maximum payload size is 25 bytes!

The NRF24L01+ has a maximum of 32 bytes. The MySensors library (version 1.4) uses 7 bytes for the message header.

## message-type

Type	Value	Comment
presentation	0	Sent by a node when they present attached sensors. This is usually done in <b>setup()</b> at startup.
set	1	This message is sent from or to a sensor when a sensor value should be updated
req	2	Requests a variable value (usually from an actuator destined for controller).
internal	3	This is a special internal message. See table below for the details
stream	4	Used for OTA firmware updates

## presentation

When a presentation message is sent, **sub-type** has to be set to one the following:

Type	Value	Comment
S_DOOR	0	Door and window sensors
S_MOTION	1	Motion sensors
S_SMOKE	2	Smoke sensor
S_LIGHT	3	Light Actuator (on/off)
S_DIMMER	4	Dimmable device of some kind
S_COVER	5	Window covers or shades
S_TEMP	6	Temperature sensor
S_HUM	7	Humidity sensor
S_BARO	8	Barometer sensor (Pressure)
S_WIND	9	Wind sensor
S_RAIN	10	Rain sensor
S_UV	11	UV sensor
S_WEIGHT	12	Weight sensor for scales etc.
S_POWER	13	Power measuring device, like power meters
S_HEATER	14	Heater device
S_DISTANCE	15	Distance sensor
S_LIGHT_LEVEL	16	Light sensor
S_ARDUINO_NODE	17	Arduino node device
S_ARDUINO_RELAY	18	Arduino repeating node device
S_LOCK	19	Lock device
S_IR	20	Ir sender/receiver device
S_WATER	21	Water meter
S_AIR_QUALITY	22	Air quality sensor e.g. MQ-2
S_CUSTOM	23	Use this for custom sensors where no other fits.
S_DUST	24	Dust level sensor
S_SCENE_CONTROLLER	25	Scene controller device

## set , req

When a set or request message is being sent, the **sub-type** has to be one of the following:

Type	Value	Comment
V_TEMP	0	Temperature
V_HUM	1	Humidity
V_LIGHT	2	Light status. 0=off 1=on
V_DIMMER	3	Dimmer value. 0-100%
V_PRESSURE	4	Atmospheric Pressure
V_FORECAST	5	Whether forecast. One of "stable", "sunny", "cloudy", "unstable", "thunderstorm" or "unknown"
V_RAIN	6	Amount of rain
V_RAINRATE	7	Rate of rain
V_WIND	8	Windspeed
V_GUST	9	Gust
V_DIRECTION	10	Wind direction
V_UV	11	UV light level
V_WEIGHT	12	Weight (for scales etc)
V_DISTANCE	13	Distance
V_IMPEDANCE	14	Impedance value
V_ARMED	15	Armed status of a security sensor. 1=Armed, 0=Bypassed
V_TRIPPED	16	Tripped status of a security sensor. 1=Tripped, 0=Untripped
V_WATT	17	Watt value for power meters
V_KWH	18	Accumulated number of KWH for a power meter
V_SCENE_ON	19	Turn on a scene
V_SCENE_OFF	20	Turn of a scene
V_HEATER	21	Mode of header. One of "Off", "HeatOn", "CoolOn", or "AutoChangeOver"
V_HEATER_SW	22	Heater switch power. 1=On, 0=Off
V_LIGHT_LEVEL	23	Light level. 0-100%
V_VAR1	24	Custom value
V_VAR2	25	Custom value
V_VAR3	26	Custom value
V_VAR4	27	Custom value
V_VAR5	28	Custom value
V_UP	29	Window covering. Up.
V_DOWN	30	Window covering. Down.
V_STOP	31	Window covering. Stop.
V_IR_SEND	32	Send out an IR-command
V_IR_RECEIVE	33	This message contains a received IR-command
V_FLOW	34	Flow of water (in meter)
V_VOLUME	35	Water volume
V_LOCK_STATUS	36	Set or get lock status. 1=Locked, 0=Unlocked
V_DUST_LEVEL	37	Dust level
V_VOLTAGE	38	Voltage level
V_CURRENT	39	Current level

## internal

The internal messages are used for different tasks in the communication between sensors, the gateway to controller and between sensors and the gateway.

When an internal messages is sent, the **sub-type** has to be one of the following:

Type	Value	Comment
I_BATTERY_LEVEL	0	Use this to report the battery level (in percent 0-100).
I_TIME	1	Sensors can request the current time from the Controller using this message. The time will be reported as the seconds since 1970
I_VERSION	2	Sensors report their library version at startup using this message type
I_ID_REQUEST	3	Use this to request a unique node id from the controller.
I_ID_RESPONSE	4	Id response back to sensor. Payload contains sensor id.
I_INCLUSION_MODE	5	Start/stop inclusion mode of the Controller (1=start, 0=stop).
I_CONFIG	6	Config request from node. Reply with (M)etric or (I)mperal back to sensor.
I_FIND_PARENT	7	When a sensor starts up, it broadcast a search request to all neighbor nodes. They reply with a I_FIND_PARENT_RESPONSE.
I_FIND_PARENT_RESPONSE	8	Reply message type to I_FIND_PARENT request.
I_LOG_MESSAGE	9	Sent by the gateway to the Controller to trace-log a message
I_CHILDREN	10	A message that can be used to transfer child sensors (from EEPROM routing table) of a repeating node.
I_SKETCH_NAME	11	Optional sketch name that can be used to identify sensor in the Controller GUI
I_SKETCH_VERSION	12	Optional sketch version that can be reported to keep track of the version of sensor in the Controller GUI.
I_REBOOT	13	Used by OTA firmware updates. Request for node to reboot.
I_GATEWAY_READY	14	Send by gateway to controller when startup is complete.

## Serial Communication Examples

Here are a few example of the serial protocol (from the gateways perspective).

**Received message from radio network from one of the sensors:** Incoming presentation message from node 12 with child sensor 12. The presentation is for a binary light **S\_LIGHT**. The payload tells that library version 1.4 is used on the node. Gateway passes this over to the controller.

```
12;6;0;0;3;1.4\n
```

**Received message from radio network from one of the sensors:** Incoming temperature **V\_TEMP** message from node 12 with child sensor 6. The gateway passed this over to the controller.

```
12;6;1;0;0;36.5\n
```

**Received command from the controller that should be passed to radio network:**

Outgoing message to node 13. Set **V\_LIGHT** variable to 1 (=turn on) for child sensor 7. No ack is requested from destination node.

```
13;7;1;0;2;1\n
```