

Senstick Temperature Food 2.3 LoRaWAN Protocol FW v1.8

1. LoraWAN DATA Payload (Uplink)

Parameter:	stat	t	rh	ap	mov	bat	soil	SUM
Size:	1B	2B	2B	1B	1B	1B	2B	10B

Parameter	Name	Range	Size	Type	Description
Status	stat	0 - 255	1B	uint8	Status Codes: 0x00 - OK, Bit0 - Movement Detected, Bit1 - Accelerometer Failure, Bit2 - T/RH Sensor Failure, Bit3 - AP Sensor Failure, Bit4 - VOC Sensor Failure, Bit5 - NFC Failure, Bit6 - Reserved, Bit7 - Reserved
Temperature	t	-128.00 - 127.00	2B	int16	Temperature (t / 100)
Relative Humidity	rh	0.00 - 100.00%	2B	uint16	Relative Humidity (rh1 / 100)
Air Pressure	ap	845 - 1100 mbar	1B	uint8	Air Pressure (ap1 + 845)
Movement	mov	0 - 2.00G	1B	uint8	Acceleration on the max axis (mov / 100)
Battery Level	bat	1 - 3.55V	1B	uint8	Battery Level (bat / 100 + 1) (min. 2.2V)
TemperatureProbe	tp	-50.00 - 350.00	2B	uint16	Temperature Probe (t / 100)

NOTE: LoRaWAN Port 2 is used.

2. LoraWAN CONFIG Payload (Uplink)

Par	sendp	joinr	joinp1	joinp2	adren	dr	accen	tmpen	apen	vocen	accth	ackco	fww	S
Size	2B	2B	2B	2B	1b	3b	1b	1b	1b	1b	1B	1B	1B	11B

Parameter	Name	R/W	Size	Type	Default Value	Description
Send Period	sendp	R/W	2B	uint16	900 sec	Send period in seconds.
Join Retries	joinr	R/W	2B	uint16	25	A number of Join retries after boot using Join Retry Period 1.
Join Retry Period 1	joinp1	R/W	2B	uint16	60 sec	First Join Retry Period after boot.
Join Retry Period 2	joinp2	R/W	2B	uint16	1800 sec	Second Join Retry Period after all Join Retries failed.
Enable ADR	adren	R/W	1b	bool	True	Enable Adaptive Data Rate. If False always use Data Rate settings.
Data Rate	dr	R/W	3b	uint3	0	Use only if Enable ADR is False. DR0 - DR7 (x = DRx).
Accelerometer Enabled	accen	R/W	1b	bool	True	Accelerometer enabled.
Temp/Humidity Enabled	tmpen	R/W	1b	bool	True	HDC2080 enabled.
Air Pressure Enabled	apen	R/W	1b	bool	True	DPS310 enabled.
VOC Enabled	vocen	R/W	1b	bool	False	BMP680 enabled.
Accelerometer Threshold (x, y, z)	accth	R/W	1B	uint8	10 % max value	If accelerometer x, y or z-axis value is greater than the threshold value, wakeup system.
Packet Confirm	ackco	R/W	1B	uint8	4	Request confirmed packed every N transmissions. 0 == OFF.
Firmware Version	fww	R	1B	uint8	1.8	Firmware Version (fww / 10).

NOTE: LoRaWAN Port 2 is used.

3. LoRaWAN CONFIG Payload (Downlink) - Send Period

Param:	sendp	SUM
Size:	2B	2B

Parameter	Name	R/W	Size	Type	Default Value	Description
Send Period	sendp	R/W	2B	uint16	900 sec	Send period in seconds.

DEFAULT DOWNLINK PACKET: 0384

4. LoRaWAN CONFIG Payload (Downlink) - Reboot

Param:	rejoin	SUM
Size:	1B	1B

Parameter	Name	R/W	Size	Type	Default Value	Description
Rejoin	reboot	W	1B	uint8	1	Start REBOOT procedure.

DEFAULT DOWNLINK PACKET: 01

5. LoRaWAN CONFIG Payload (Downlink)

Param	sendp	joinr	joinp1	joinp2	adren	dr	accen	tmpen	apen	vocen	accth	ackco	SUM
Size:	2B	2B	2B	2B	1b	3b	1b	1b	1b	1b	1B	1B	10B

Parameter	Name	R/W	Size	Type	Default Value	Description
Send Period	sendp	R/W	2B	uint16	900 sec	Send period in seconds.
Join Retries	joinr	R/W	2B	uint16	25	A number of Join retries after boot using Join Retry Period 1.
Join Retry Period 1	joinp1	R/W	2B	uint16	60 sec	First Join Retry Period after boot.
Join Retry Period 2	joinp2	R/W	2B	uint16	1800 sec	Second Join Retry Period after all Join Retries failed.
Enable ADR	adren	R/W	1b	bool	True	Enable Adaptive Data Rate. If False always use Data Rate settings.
Data Rate	dr	R/W	3b	uint3	0	Use only if Enable ADR is False. DR0 - DR7 (x = DRx).
Accelerometer Enabled	accen	R/W	1b	bool	True	Accelerometer enabled.
Temp/Humidity Enabled	tmpen	R/W	1b	bool	True	HDC2080 enabled.
Air Pressure Enabled	apen	R/W	1b	bool	True	DPS310 enabled.
VOC Enabled	vocen	R/W	1b	bool	False	BMP680 enabled.
Accelerometer Threshold (x, y, z)	accth	R/W	1B	uint8	10 % max value	If accelerometer x, y or z-axis value is greater than the threshold value, wakeup system.
Packet Confirm	ackco	R/W	1B	uint8	4	Request confirmed packed every N transmissions. 0 == OFF.

DEFAULT DOWNLINK PACKET: 03840019003C07088E3204

[PACKET GENERATOR](#)

6. TTN Payload Decoder

TTN PAYLOAD DECODER

```

/*

$$\frac{\text{TTN Payload Decoder}}{\text{Senstick STF20 - HWv2.3 FWv1.8}}$$

*/

function Decoder(bytes) {

    // If Config Packet
    if (bytes.length == 12) {

        var SendPeriod = (bytes[0] << 8) + bytes[1];
        var JoinRetries = (bytes[2] << 8) + bytes[3];
        var JoinRetryPeriod1 = (bytes[4] << 8) + bytes[5];
        var JoinRetryPeriod2 = (bytes[6] << 8) + bytes[7];
        var EnableADR = bytes[8] >> 7;
        var DataRate = (bytes[8] >> 4) - 8*EnableADR;
        var AccEnabled = (bytes[8] >> 3) - 16*EnableADR - 2*DataRate;
        var TempHumidityEnabled = (bytes[8] >> 2) - 32*EnableADR - 4*DataRate - 2*AccEnabled;
        var AirPressureEnabled = (bytes[8] >> 1) - 64*EnableADR - 8*DataRate - 4*AccEnabled -
        2*TempHumidityEnabled;
        var VOCEnabled = bytes[8] - 128*EnableADR - 16*DataRate - 8*AccEnabled -
        4*TempHumidityEnabled - 2*AirPressureEnabled;
        var AccThreshold = bytes[9];
        var PacketConformation = bytes[10];
        var FirmwareVersion = bytes[11]/10;

        return {
            SendPeriod: SendPeriod,
            JoinRetries: JoinRetries,
            JoinRetryPeriod1: JoinRetryPeriod1,
            JoinRetryPeriod2: JoinRetryPeriod2,
            EnableADR: EnableADR,
            DataRate: DataRate,
            AccEnabled: AccEnabled,
            TempHumidityEnabled: TempHumidityEnabled,
            AirPressureEnabled: AirPressureEnabled,
            VOCEnabled: VOCEnabled,
            AccThreshold: AccThreshold,
            PacketConformation: PacketConformation,
            FirmwareVersion: FirmwareVersion
        }
    }
}

```

```
};

}
// If Data Packet
else if (bytes.length == 10) {

    var S = bytes[0];
    var T = (bytes[1] << 8) + bytes[2];
    var H = (bytes[3] << 8) + bytes[4];
    var AP1 = bytes[5];
    var MOV = bytes[6];
    var BAT = bytes[7];
    var VADC3 = (bytes[8] << 8) + bytes[9];

    if (AP1 != 0) AP1 = AP1 + 845;

    // Calculate Temperature from NTC Probe
    var R1 = 10000;
    var c1 = 1.009249522*Math.pow(10,-3), c2 = 2.378405444*Math.pow(10,-4), c3 =
2.019202697*Math.pow(10,-7);

    var Vo = VADC3/1000;
    var VR = 3 - Vo;
    var RT = Vo / (VR / R1);
    var ln = Math.log(RT / R1);
    var T0 = 25 + 273.15;
    var TX = (1 / ((ln / 3977) + (1 / T0)));
    var Tc = TX - 273.15;

    return {
        Status: S,
        Temperature: sintToDec(T),
        Humidity: H / 100.0,
        AirPressure: AP1,
        Movement: MOV / 100,
        BatteryLevel: (BAT + 100) / 100,
        TemperatureProbe: Math.round(Tc * 100) / 100
    };

}

}

function sintToDec(T){
    if (T > 32767) {
        return ((T - 65536) / 100.0);
    }
    else {
        return (T / 100.0);
    }
}
}
```

7. TTN Downlink Guide

Overview Live data **Messaging** Location Payload formatters Claiming General settings

Uplink **Downlink**

Schedule downlink

Insert Mode

Replace downlink queue
 Push to downlink queue (append)

FPort*

←

Payload type

Bytes JSON ←

Payload

←

The desired payload bytes of the downlink message

Confirmed downlink ←

←

Application > End Device > Messaging > Downlink