



PLACEPOD® SMART PARKING SENSOR COMMUNICATIONS PROTOCOL

This document describes the structure of messages exchanged between the PlacePod sensor and the LoRa® Server.

TABLE OF CONTENTS

1	COPYRIGHT & WARRANTY INFORMATION	2
2	TX PACKET FORMAT	3
2.1	PLACEPOD DATA PACKET TYPES	4
2.1.1	CAR DETECTOR SENSOR (0X15)	4
2.1.1.1	CAR DETECTOR RESULT	4
2.1.2	PONG (0X33).....	4
2.1.3	VERSION (0X34)	5
2.1.3.1	VERSION COMPONENT ID	5
2.1.4	CONFIGURATION PARAMETER (0X35).....	6
2.1.4.1	CONFIGURATION PARAMETERS LIST	6
2.1.4.2	CONFIGURATION PARAMETER DESCRIPTIONS.....	6
2.1.5	RADIO STATS (0X36)	7
2.1.5.1	STATISTIC TYPES	8
2.1.5.2	STATISTIC VALUES	8
2.1.6	KEEP-ALIVE (0X37)	8
2.1.6.1	CAR STATUS VALUES.....	8
2.1.6.2	TEMPERATURE AND BATTERY STATUS VALUES	9
2.1.7	META EVENT (0XFE).....	9
3	RX PACKET FORMAT	10
3.1	PLACEPOD COMMAND PACKET TYPES	11
3.1.1	RECALIBRATE (0X01).....	11
3.1.2	SET KEEP ALIVE INTERVAL (0X07)	11
3.1.3	SET CONFIGURATION PARAMETER (0X0A).....	11
3.1.4	GET CONFIGURATION PARAMETER (0X0B)	11
3.1.5	PING (0X0D).....	12
3.1.6	RESET TO FACTORY DEFAULT (0X0F)	12
3.1.7	SET RADIO TX POWER (0X11)	12
3.1.8	SET RADIO TX SPREADING FACTOR (0X12).....	12
3.1.9	SET RADIO ADAPTIVE DATA RATE (0X14)	13
3.1.10	SET RADIO FREQUENCY SUB BAND (0X15).....	13
3.1.11	GET RADIO STATS (0X16).....	13
3.1.12	GET MULTIPLE CONFIGURATION PARAMETERS (0X17)	14
3.1.13	SET TX RETRIES (0X18).....	14
3.1.14	SET LINK-CHECK THRESHOLD (0X19)	14
3.1.15	ENABLE SHIPPING MODE (0X1C)	14
3.1.16	SOFT RESET (0X3F).....	14

1 COPYRIGHT & WARRANTY INFORMATION

Copyright ©2018 Protonex LLC dba PNI Sensor. All Rights Reserved.

Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under copyright laws.

PNI Sensor
2331 Circadian Way
Santa Rosa, CA 95407, USA
Tel: (707) 566-2260
Fax: (707) 566-2261

Warranty and Limitation of Liability – PlacePod Smart Parking Sensor (“Product”). PNI Sensor ("PNI") manufactures its Products from parts and components that are new or equivalent to new in performance. PNI warrants that each Product to be delivered hereunder, if properly used, will, for one year (365) days following the date of shipment be free from defects in material and workmanship and will operate in accordance with PNI’s published specifications and documentation for the Product in effect at time of order. Any Product that is non-functional and unresponsive to firmware and software updates will be considered failed hardware units and will be replaced at no charge. This warranty includes one year (365 days) of debugging maintenance for customers that use the PNI Parking Cloud Software to manage PNI’s Product.

This warranty does not cover wear and tear due to normal use, or damage to the Product as the result of improper installation, misuse, neglect of care, alteration, vandalism, theft, accident, or unauthorized repair.

The above warranty is in lieu of any other warranty, whether express, implied, or statutory, including but not limited to, any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. PNI neither assumes nor authorizes any person to assume for it any other liability.

If any Product furnished hereunder fails to conform to the above warranty, Customer’s sole and exclusive remedy and PNI’s sole and exclusive liability will be, at PNI’s option, to repair, replace, or credit Customer’s account with an amount equal to the price paid for any such Product which fails during the applicable warranty period provided that (i) Customer promptly notifies PNI in writing that such Product is defective and furnishes an explanation of the deficiency; (ii) such Product is returned to PNI’s service facility at Customer’s risk and expense; and (iii) PNI is satisfied that claimed deficiencies exist and were not caused by improper installation, misuse, neglect of care, alteration, vandalism, theft, accident, or unauthorized repair. If a Product is defective, transportation charges for the return of the Product to Customer within the United States and Canada will be paid by PNI. For all other locations, the warranty excludes all costs of shipping, customs clearance, and other related charges. PNI will have a reasonable time to make repairs or to replace the Product or to credit Customer’s account. PNI warrants any such repaired or replacement Product to be free from defects in material and workmanship on the same terms as the Product originally purchased.

Except for the breach of warranty remedies set forth herein PNI shall have no liability for any indirect or speculative damages (including, but not limited to, consequential, incidental, punitive and special damages) relating to the use of or inability to use this Product, whether arising out of contract, negligence, tort, or under any warranty theory, or for infringement of any other party’s intellectual property rights, irrespective of whether PNI had advance notice of the possibility of any such damages, including, but not limited to, loss of use, revenue or profit. In no event shall PNI’s total liability for all claims regarding a Product exceed the price paid for the Product. PNI neither assumes nor authorizes any person to assume for it any other liabilities.

2 TX PACKET FORMAT

A single, fixed-length 17 byte packet structure is used to wrap the various data structures to be sent to the radio for transmission. All structures are packed and little-endian unless otherwise noted.

Type	Bytes	Field	Description
uint8	1	packet_type	Parking Sensor Data Packet Type
uint8[]	12	payload	See payload section for each packet type
uint32	4	timestamp	32kHz ticks since boot

Table 1: PlacePod Data Packet Format

Example Code

```
struct parking_sensor_data_packet {
    uint8_t packet_type;
    union {
        struct car_detector car_detector;
        struct pong pong;
        struct version version;
        struct configuration_parameter configuration_parameter;
        struct radio_stats radio_stats;
        struct keep_alive keep_alive;
        struct meta_event meta_event;
    };
    uint8_t bytes[12];
    uint32_t timestamp;
};
```

Value	Description
0x15	Car Detector Sensor
0x33	Pong
0x34	Version
0x35	Configuration Parameter
0x36	Radio Stats
0x37	Keep-Alive
0xFE	Meta Event

Table 2: PlacePod Data Packet Types

Example Code

```
enum parking_sensor_packet_types {
    PARKING_TYPE_CAR_DETECTOR = 0x15,
    PARKING_TYPE_PONG = 0x33,
    PARKING_TYPE_VERSION = 0x34,
    PARKING_TYPE_CONFIGURATION_PARAMETER = 0x35,
    PARKING_TYPE_RADIO_STATS = 0x36,
    PARKING_TYPE_KEEP_ALIVE = 0x37,
    PARKING_TYPE_META_EVENT = 0xFE,
};
```

2.1 PLACEPOD DATA PACKET TYPES

Detailed information for each of the PlacePod data packet types.

2.1.1 CAR DETECTOR SENSOR (0X15)

Car Detector Sensor output contains the results of the car detection algorithm, and additional information.

Type	Bytes	Field	Description
uint8	1	result	Car Detector sensor results
uint8	1	unused	unused
float	4	temperature	Temperature reading in Celsius
float	4	battery_status	Battery reading in Volts
uint8	1	unused	unused
uint8	1	packet count	Current packet count

Example Code

```
struct car_detector {  
    uint8_t result;  
    float temperature;  
    float battery_status;  
    uint8_t pkt_count;  
};
```

2.1.1.1 CAR DETECTOR RESULT

Car detection results which contain the Vacant/Occupied state.

Value	Description
0x01	Vacant
0x03	Occupied

Example Code

```
enum car_detector_results {  
    CAR_DETECTOR_RESULT_VACANT = 0x01,  
    CAR_DETECTOR_RESULT_OCCUPIED = 0x03,  
};
```

2.1.2 PONG (0X33)

Ping response containing the RSSI and SNR from the ping request.

Type	Bytes	Field	Description
float	4	RSSI	RSSI from the ping request
float	4	SNR	SNR from the ping request
uint8	1	unused	unused
uint8	1	unused	unused
uint8	1	unused	unused
uint8	1	packet count	Current packet count

Example Code

```
struct pong {  
    float rssi;  
    float snr;  
};
```

2.1.3 VERSION (0X34)

Structure containing the version information for the software components of the PlacePod.

Type	Bytes	Field	Description
uint8	1	vid	Version Component ID
uint8	1	major	Major Version Number
uint8	1	minor	Minor Version Number
uint8	1	patch	Patch Version Number
uint32	4	build	Build Number
uint8	1	unused	unused
uint8	1	unused	unused
uint8	1	unused	unused
uint8	1	pkt counter	Current packet count

Example Code

```
struct version {  
    uint8_t vid;  
    uint8_t major;  
    uint8_t minor;  
    uint8_t patch;  
    uint32_t build;  
};
```

2.1.3.1 VERSION COMPONENT ID

ID of the component to which the attached version applies.

Value	Description
0x01	Host Application
0x02	Sensor Firmware
0x03	BLE Module Firmware
0x04	Radio (LoRa) Firmware

Example Code

```
enum version_id {  
    VID_HOST = 0x01,  
    VID_SENSOR = 0x02,  
    VID_BLE = 0x03,  
    VID_RADIO = 0x04,  
};
```

2.1.4 CONFIGURATION PARAMETER (0X35)

PlacePod configuration parameter information.

Type	Bytes	Field	Description
uint8	1	page_no	Page number of configuration parameter
uint8	1	param_no	Parameter number of configuration parameter
uint8[]	8	data	Data
uint8	1	unused	unused
uint8	1	packet count	packet count

Example Code

```
struct cfg_param {
    uint8_t page_no;
    uint8_t param_no;
    uint8_t data[8];
};
```

2.1.4.1 CONFIGURATION PARAMETERS LIST

List of available configuration parameters.

Page #	Param #	Values	Defaults	Description
16	3	Version Component ID	NA	Firmware Version Info
16	6	1-43200	300	Keep-Alive Interval (seconds)
16	7	0-20	20	LoRa Radio Tx Power (dBm)
16	8	0-8	3	LoRa Spreading Factor (see table)
16	9	0-8	1	LoRa Frequency Sub-Band (see table)
16	24	0-7	2	LoRa Tx Retries
16	25	0-255	15	LoRa Link Check Threshold

2.1.4.2 CONFIGURATION PARAMETER DESCRIPTIONS

Configuration parameter definitions:

Keep-Alive Interval

Interval in seconds for which the [Keep-Alive \(0x37\)](#) payload will report, default value = 300 (seconds).

LoRa Radio Tx Power

LoRa modules Tx Power setting in dBm, default value = 20 (dBm).

LoRa Spreading Factor

Default value = 3 (Spreading Factor 9).

Value	Spreading Factor	Bandwidth
0	12	125 kHz
1	11	125 kHz
2	10	125 kHz
3	9	125 kHz
4	8	125 kHz
5	8	500 kHz

6	7	125 kHz
7	7	250 kHz

Note: Due to PlacePod payload size PlacePod does not support anything lower than Spreading Factor 9.

LoRa Frequency Sub-Band

Sub-band on which the US915 PlacePod is operating. Default value = 1 (902.3 kHz -- 903.7 kHz - 125k).

Value	Sub Band
0	Hop all 64 channels
1	902.3 kHz -- 903.7 kHz - 125k
2	903.9 kHz -- 905.3 kHz - 125k
3	905.5 kHz -- 906.9 kHz - 125k
4	907.1 kHz -- 908.5 kHz - 125k
5	908.7 kHz -- 910.1 kHz - 125k
6	910.3 kHz -- 911.7 kHz - 125k
7	911.9 kHz -- 913.3 kHz - 125k
8	913.5 kHz -- 914.9 kHz - 125k

LoRa Tx Retries

The total number of times the device will retry the Rejoin-request, default value = 2.

0 : the Rejoin is sent only once (no retry)

1 : the Rejoin MUST be sent 2 times in total (1 + 1 retry)

...

7: the Rejoin MUST be sent 8 times (1 + 7 retries)

LoRa Link Check Threshold

Determines how many missed responses determine a network loss, default value = 15.

2.1.5 RADIO STATS (0X36)

Structure containing statistics related to the radio's network traffic. The statistics may be broken up into multiple packets so they are subtyped.

Type	Bytes	Field	Description
uint8	1	type	Statistic type
uint32[]	8	sensor	Values (2)
uint8	1	unused	unused
uint8	1	unused	unused
uint8	1	packet count	Current packet count

Example Code

```
struct radio_stats {
    uint8_t type;
    uint32_t values[2];
};
```


2.1.5.1 STATISTIC TYPES

Statistics types available.

Value	Description
0x01	Network Joins
0x02	Packets RX/TX
0x03	ACKS

Example Code

```
enum radio_stat_type {  
    RADIO_STAT_TYPE_JOINS = 0x01,  
    RADIO_STAT_TYPE_Packets = 0x02,  
    RADIO_STAT_TYPE_ACKS = 0x03,  
};
```

2.1.5.2 STATISTIC VALUES

Values for each of the Statistic Types.

Type	Value[0]	Value[1]
0x01	Join Attempts	Join Failures
0x02	Packets TX	Packets RX
0x03	ACKs Missed	CRC Errors

2.1.6 KEEP-ALIVE (0X37)

PlacePod will report a keep-alive payload at a fixed interval. Default interval is 300 seconds.

Type	Bytes	Field	Description
uint16	2	car status	Car Detector sensor results
uint32	4	tempBattStatus	Temperature and Battery Status
uint8	1	unused	
uint8	1	unused	
uint8	1	unused	
uint8	1	unused	
uint8	1	unused	
uint8	1	packet count	Current packet count

2.1.6.1 CAR STATUS VALUES

Car detection results which contain the Vacant/Occupied state.

Value	Description
0x00	Uninitialized
0x01	Vacant
0x03	Occupied

Example Code

```
enum car_detector_results {  
    CAR_DETECTOR_RESULT_UINITIALIZED = 0x00,  
    CAR_DETECTOR_RESULT_VACANT = 0x01,  
    CAR_DETECTOR_RESULT_OCCUPIED = 0x03,  
};
```

2.1.6.2 TEMPERATURE AND BATTERY STATUS VALUES

Information on the temperature and battery readings from the sensor.

Type	Bytes	Field	Description
uint16	4	temperature	Temperature in Celsius with scale factor 10
uint16	4	battery	Battery level in Volts with scale factor 1000

Example Code

```
struct temperature_battery_status {  
    uint16_t temperature;  
    uint16_t battery;  
};
```

2.1.7 META EVENT (0XFE)

Please contact PNI if this payload is observed.

3 RX PACKET FORMAT

Packets received from the radio consist of at least one byte for the packet type (command) and optionally a payload. All structures are packed in little-endian unless otherwise noted.

Value	Description
0x01	Recalibrate
0x07	Set Keep Alive Interval
0x0A	Set Configuration Parameter
0x0B	Get Configuration Parameter
0x0D	Ping
0x0F	Reset to Factory Default
0x11	Set Radio TX Power
0x12	Set Radio TX Spreading Factor
0x14	Set Radio Adaptive Data Rate
0x15	Set Radio Frequency Sub Band
0x16	Get Radio Stats
0x17	Get Multiple Configuration Parameters
0x18	Set TX Retries
0x19	Set Link-Check Threshold
0x1C	Enable Shipping Mode
0x3F	Soft Reset

Table 3: PlacePod Command Packet Types

Example Code

```
enum parking_command_types {
    PARKING_CMD_TYPE_RECALIBRATE = 0x01,
    PARKING_CMD_TYPE_SET_KEEP_ALIVE_INTERVAL = 0x07,
    PARKING_CMD_TYPE_SET_PARAMETER = 0x0A,
    PARKING_CMD_TYPE_GET_PARAMETER = 0x0B,
    PARKING_CMD_TYPE_PING = 0x0D,
    PARKING_CMD_TYPE_FACTORY_DEFAULT = 0x0F,
    PARKING_CMD_TYPE_SET_RADIO_TX_POWER = 0x11,
    PARKING_CMD_TYPE_SET_RADIO_TX_SPREAD_FACTOR = 0x12,
    PARKING_CMD_TYPE_SET_RADIO_ADR = 0x14,
    PARKING_CMD_TYPE_SET_RADIO_FSB = 0x15,
    PARKING_CMD_TYPE_GET_RADIO_STATS = 0x16,
    PARKING_CMD_TYPE_GET_MULTI_PARAMETERS = 0x17,
    PARKING_CMD_TYPE_SET_TX_RETRIES = 0x18,
    PARKING_CMD_TYPE_SET_LINK_CHECK_THRD = 0x19,
    PARKING_CMD_TYPE_SHIPPING_MODE = 0x1C,
    PARKING_CMD_TYPE_SOFT_RESET = 0x3F,
};
```

3.1 PLACEPOD COMMAND PACKET TYPES

Detailed information for each of the PlacePod command packet types.

3.1.1 RECALIBRATE (0X01)

Requests the PlacePod to perform its calibration procedure. This will set the baseline for a VACANT parking space. This **must** be done once the PlacePod has been installed in the parking space. The parking space and any adjacent spots should be empty.

No payload is required.

3.1.2 SET KEEP ALIVE INTERVAL (0X07)

Sets the [Keep-Alive \(0x37\)](#) message interval in seconds, between 10 and 43200 seconds. Default value = 300 (seconds).

Type	Bytes	Description
uint16	2	Keep Alive Interval (seconds)

Example Code

```
typedef uint16_t keepalive_interval_s;
```

Note: Setting the keep alive interval too low can drastically reduce battery life.

3.1.3 SET CONFIGURATION PARAMETER (0X0A)

Sets a specified configuration parameter. Integral parameters will need to be cast or converted to float.

Type	Bytes	Field	Description
uint8	1	page_no	Page number of parameter to set
uint8	1	param_no	Parameter number of parameter to set
float	4	Value	Value to be set

Example Code

```
struct set_cfg_param {  
    uint8_t page_no;  
    uint8_t param_no;  
    float value;  
};
```

See [Configuration Parameters List](#) for list of specific parameters.

3.1.4 GET CONFIGURATION PARAMETER (0X0B)

Requests a configuration parameter. Transmits the value as a [Configuration Parameter \(0x35\)](#) message.

Type	Bytes	Field	Description
uint8	1	page_no	Page number of parameter to set
uint8	1	param_no	Parameter number of parameter to set

Example Code

```
struct get_cfg_param {
    uint8_t page_no;
    uint8_t param_no;
};
```

See [Configuration Parameters List](#) for list of specific parameters.

3.1.5 PING (0X0D)

Transmits the requests RSSI and SNR as a [Pong \(0x33\)](#).

No payload is required.

3.1.6 RESET TO FACTORY DEFAULT (0X0F)

Resets all PlacePod settings back to defaults.

Parameter	Default	Units
Keep-Alive Interval	300	Seconds
LoRa Module Tx Power	20	dBm
LoRa Tx Spreading Factor	3	See lookup table in Configuration Parameter Descriptions
Adaptive Data Rate	0	0 = disabled, 1 = enabled
LoRa Frequency Sub Band	1	See lookup table in Configuration Parameter Descriptions
LoRa Tx Retries	2	See lookup table in Configuration Parameter Descriptions
LoRa Link Check Threshold	15	See lookup table in Configuration Parameter Descriptions

No payload is required.

3.1.7 SET RADIO TX POWER (0X11)

Sets the LoRa modules Tx power in dBm between 0 and 20. Default value = 20.

Type	Bytes	Description
uint8	1	Tx Power (dBm)

Example Code

```
typedef uint8_t radio_tx_power_dbm;
```

3.1.8 SET RADIO TX SPREADING FACTOR (0X12)

Sets the LoRa Spreading Factor, default value = 3 (Spreading Factor 9).

Type	Bytes	Description
uint8	1	Spreading Factor Value

Example Code

```
typedef uint8_t radio_tx_sf;
```

See [Configuration Parameter Descriptions](#) for values.

Example Code

```
enum parking_command_types {
    RADIO_TX_SF_SF12BW125 = 0x00,
    RADIO_TX_SF_SF11BW125 = 0x01,
    RADIO_TX_SF_SF10BW125 = 0x02,
    RADIO_TX_SF_SF09BW125 = 0x03,
    RADIO_TX_SF_SF08BW125 = 0x04,
    RADIO_TX_SF_SF08BW500 = 0x05,
    RADIO_TX_SF_SF07BW125 = 0x06,
    RADIO_TX_SF_SF07BW250 = 0x07,
};
```

3.1.9 SET RADIO ADAPTIVE DATA RATE (0X14)

Sets the LoRa Adaptive Data Rate Configuration, default value = 0 (disabled).

Type	Bytes	Description
uint8	1	ADR Config

Example Code

```
typedef uint8_t radio_adr;
```

Value	Description
0x00	ADR Disabled
0x01	ADR Enabled

Example Code

```
enum radio_adr_config {
    RADIO_ADR_CFG_DISABLED = 0x00,
    RADIO_ADR_CFG_ENABLED = 0x01,
};
```

3.1.10 SET RADIO FREQUENCY SUB BAND (0X15)

Sets the LoRa Frequency Sub Band Configuration, values vary between regions. US915 default value = 1.

Type	Bytes	Description
uint8	1	FSB Config

Example Code

```
typedef uint8_t radio_fsb;
```

See [Configuration Parameter Descriptions](#) for values.

3.1.11 GET RADIO STATS (0X16)

Requests the [Radio Stats \(0x36\)](#) from the device.

No payload is required.

3.1.12 GET MULTIPLE CONFIGURATION PARAMETERS (0X17)

Requests multiple configuration parameters up to 6 parameters from the PlacePod. Transmits the value of the configurations.

Type	Bytes	Description
uint8	1	Parameter 1
uint8	1	Parameter 2
uint8	1	Parameter 3
uint8	1	Parameter 4
uint8	1	Parameter 5
uint8	1	Parameter 6

Example Code

```
struct get_mult_cfg_param {  
    uint8_t param_no[5];  
};
```

See [Configuration Parameters List](#) for list of parameters, use param_# value.

3.1.13 SET TX RETRIES (0X18)

The total number of times the device will retry the Rejoin-request, default value = 2.

Type	Bytes	Description
uint8	1	Number of rejoin retries (0-255)

Example Code

```
typedef uint8_t radio_tx_retries;
```

3.1.14 SET LINK-CHECK THRESHOLD (0X19)

Determines how many missed responses determine a network loss, default value = 15.

Type	Bytes	Description
uint8	1	Number of message failures (0-255)

Example Code

```
typedef uint8_t radio_link_check_thrd;
```

3.1.15 ENABLE SHIPPING MODE (0X1C)

Requests the PlacePod to enter shipping mode.

No payload is required.

Note: Once the PlacePod has entered shipping mode you must follow the PlacePod activation procedure to take the PlacePod out of shipping mode.

3.1.16 SOFT RESET (0X3F)

Requests the device to perform a soft reset (reboot). Settings (keep-alive interval, FSB, TX Power, Spreading Factor, etc.) will persist through the reset.

No payload is required.