iSYS-5xxx Ethernet Targetlist Protocol

Data output

Devices: *iSYS-5020, iSYS-5021, iSYS-5110*

Revision: *4*

Date: *2018-11-23*

Table of content

0. History

Document revision	Date	Change log	Author	Reviewer
1	2017-01-17	initial release	PG	
2	2017-01-27	Added data packet number, changed default IP address	PG	
3	2018-05-09	Added description for iSYS-5020	JW	TS
4	2018-11-23	Added support for iSYS-5021	JK	

1. General Information

This document describes the Ethernet target list output protocol. The iSYS-5xxx with Ethernet target list interface sends the target list packets with a fixed IP-address. For the default IP-Address configuration of the iSYS-5xxx refer to Table 1 below. The IP-address and port number can be changed via serial interface. After power up the sensor starts transmitting continuously target lists.

Table 1: iSYS-5xxx default IP-Configuration

System	System IP-Address / Port	Destination IP-Address / Port
iSYS-5020 / iSYS-5021	192.168.252.10 / 2051	192.168.252.2 / 2050
iSYS-5110	192.168.60.20 / 2051	192.168.60.30 / 2050

2. Ethernet Protocol

- each package is wrapped in the following standard network protocols:
    1. Ethernet II
    2. IP v4
    3. UDP
- these protocols should add an overall offset of 42 bytes for each package
- all network protocol data (Eth, IP, UDP) are transmitted in big endian byte order (network byte order) format
- Packet Data (header/data) are in little endian order
- see graphic below as an overview

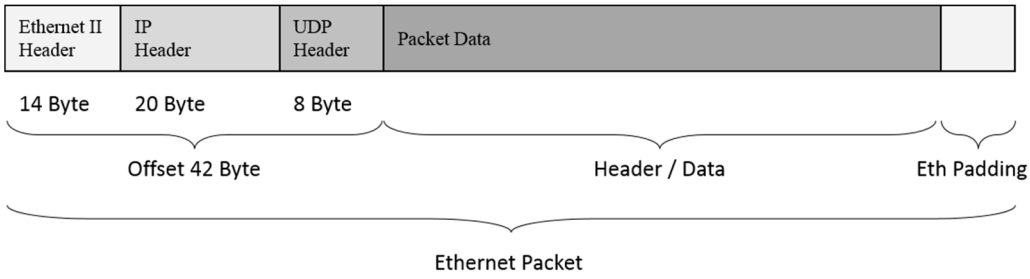

Figure 1: iSYS-5xxx ethernet output packet

2.1. Packet Data

The Packet Data includes the radar information. First a header is transmitted, which include general information (for example number of targets) about the target list (see 2.1.1). After the header the data packets (see 2.1.2) which include the targets will be send. The number of targets is be limited by 256. The number of data packets and the size can be calculated by the header.

- when the sensor is powered it will automatically start transmitting data
- the data will be sent in an infinite loop, every measurement cycle
- one loop (further referred to as “data set”) contains following packages:
    - 1 header packet (256 bytes)
    - Max. 7 data packets, which include the targets
- each data set is identified by a frameID which is incremented each measurement cycle
- the header will be sent, whether targets are available or not

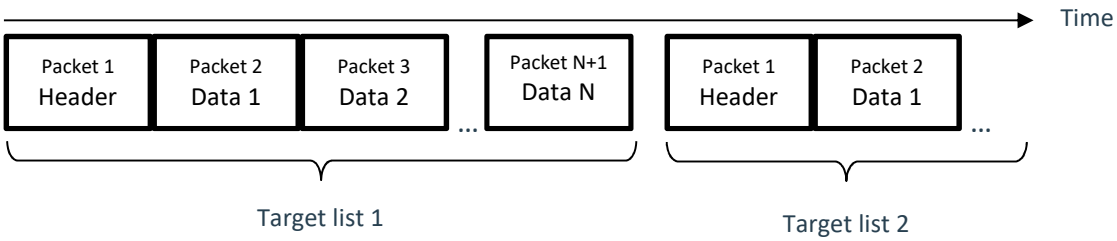

Figure 2: iSYS-5xxx targetlist output

2.1.1. Header

When transmitting a header packet a specified number of corresponding data packets will follow. The following sections specify the size and content of the header.

```
typedef struct targetListHeader{
	uint16_t frameID;		//unique frame ID warps to 0x0000 after 0xffff
	uint16_t FWmajor;		//Firmware version
	uint16_t FWfix;			//Firmware version
	uint16_t FWminor;		//Firmware version
	uint16_t nrOfDetections;	//number of detections (limited by 256)
	uint16_t nrOfTargets;		//number of targets (limited by 256)
	uint32_t crc;			//Checksum over all targets (see 2.1.3)
	uint16_t bytesPerTarget;	//number of bytes per target
	uint16_t nrOfDataPackets;	//number of data packet
	uint16_t reserved[118];		//reserved
}targetListHeader_t;
```

2.1.2. Data

The data packets includes max. 42 targets. Each targets has the information signal strength/RCS, range, velocity and azimuth angle. In the first two bytes of each data packet the frameID from header is also included. If there are more than 42 targets available, more data packets will be send one after the other (max. 7 data packets). The length of the packed data (without Ethernet II, IP and UDP header) is always 1012 bytes. If number of targets is not enough to fill the data packet completely, the remaining bytes are filled with zeros. Each data packet has an ongoing number, which starts with 0.

Following struct represents one data packet

```
typedef struct targetDataPacket{
	uint16_t frameID;				// unique frame ID from header
	uint16_t numberOfDataPacket;	// ongoing number for each data packet
	targetList_t targetList[42];
}targetDataPacket_t;
```

Following struct represents one target and needs 24bytes of memory

```
typedef struct targetList{
	float32_t f32_signalStrength;		/* [dB] */
	float32_t f32_range;				/* [m]*/
	float32_t f32_velocity;				/* [m/s] */
	float32_t f32_angleAzimuth;			/* [°] */
	float32_t f32_reserved1;
	float32_t f32_reserved2;
}targetList_t;
```

Note: *The target list output of iSYS-5020 and iSYS-5110 can changed from signal strength to radar cross section (RCS) with corresponding UART-command. The iSYS-5021 always returns the RCS.*

Each data set consists of up to 7 data packets, which represent at max. 256 targets.

The size of all data packets can be calculated by the header information.

Example: We received following header:

```
nrOfTargets     = 100;
bytesPerTarget  = 24;
nrOfDataPackets = 3;
```

1. We will receive 3 data packets with size of 1012bytes (42*24bytes + 2bytes from frameID + 2bytes from data packet number)
2. In third data packet consists of only 16 targets (16*24bytes = 384bytes) with the corresponding radar information. The rest of the frame will be padded with zeros.

2.1.3. Checksum

To make communication more robust, a checksum is transmitted in the header. The Checksum is calculated byte-wise over all targets in the data packets.

Example:

```
/* calc checksum */
targetList_t TargetList[256];       /* targetlist received from sensor */
uint32_t ui32_checksum = 0;
uint32_t i;
uint8_t *pCRC;

pCRC = (uint8_it *)TargetList;

for(i=0; i<(nrOfTargets*bytesPerTarget); i++){
       ui32_checksum += *pCRC;
       pCRC++;
}
```

The calculated checksum `ui32_checksum` must be equal to the `crc` of the header.

InnoSenT GmbH
Am Roedertor 30
97499 Donnersdorf
GERMANY

Tel.: +49 95289518-0
E-Mail: info@innosent.de
www.innosent.de